

# **Instituto Tecnológico Superior de Lerdo**



## **Instituto Mexicano Del Seguro Social – Unidad Médica Familiar #43**



**INSTITUTO MEXICANO DEL SEGURO SOCIAL**  
SEGURIDAD Y SOLIDARIDAD SOCIAL

“Sistema de inventarios para el departamento de informática.”

### **CARRERA**

Ingeniería Informática IINF-2010-220

### **RESIDENTE**

Jaime Iván Guerrero Artiño 12231063 9º semestre

#### **ASESOR EXTERNO**

Ing. Oscar Ricardo  
Guzmán Díaz

#### **ASESOR INTERNO**

Ing. Ricardo De Jesús  
Bustamante González

**Lerdo, Durango**  
**Noviembre 2016**

# ÍNDICE

1.-INTRODUCCIÓN .....	2
1.1.-JUSTIFICACION .....	2
1.2.-OBJETIVO.....	2
1.2.1.-OBJETIVO GENERAL.....	2
1.2.2.-OBJETIVOS ESPECÍFICOS .....	2
1.3. - PROBLEMAS A RESOLVER.....	3
2. PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADA .....	4
2.1.- ERS (IEEE 830) – ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE. ....	5
2.1.1.-INTRODUCCIÓN.....	5
2.1.1.2.-PROPÓSITO .....	5
2.1.1.3.-ALCANCE .....	5
2.1. 1.4.-PERSONAL INVOLUCRADO.....	5
2.1. 1.5.-DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	6
2.1. 1.6.-REFERENCIAS .....	6
2.1.1.7.-RESUMEN.....	6
2.1.2.-DESCRIPCIÓN GENERAL.....	6
2.1.2.1.-PERSPECTIVA DEL PRODUCTO.....	6
2.1.2.2.-FUNCIONALIDAD DEL PRODUCTO .....	6
2.1.2.3.-CARACTERÍSTICAS DE LOS USUARIOS.....	7
2.1.3.-REQUISITOS ESPECÍFICOS .....	7
2.1.3.1.-REQUERIMIENTOS FUNCIONALES .....	7
2.1.3.2.-REQUERIMIENTOS NO FUNCIONALES.....	10
2.1.4.-Requisitos comunes de las interfaces.....	12
2.1.4.1.-Interfaces de usuario.....	12
2.1.4.2.-Interfaces de hardware .....	13
2.1.4.3.-Interfaces de software .....	13
2.1.4.4.-Interfaces de comunicación.....	13
2.2.-Requisitos funcionales .....	13
2.2.5.-Requisito Funcional 05 - Modificar información .....	13

2.3.-Requisitos no funcionales .....	14
2.3.1.- Requisitos no funcionales - Requisitos de rendimiento .....	14
2.3.3.-Requisitos no funcionales - Seguridad.....	14
2.3.3.- Requisitos no funcionales - Fiabilidad .....	14
2.3.4- Requisitos no funcionales - Disponibilidad .....	14
2.3.5.- Requisitos no funcionales - Mantenibilidad .....	14
2.3.6.- Requisitos no funcionales - Portabilidad.....	14
2.4.-Mockups.....	15
2.4.1. Mockups de Administrador .....	15
2.4.2.-Mockups de Usuario .....	18
3.-RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MAQUETAS Y PROGRAMAS	20
3.1.-Resultados del proyecto.....	21
4.- CONCLUSIONES Y .....	63
RECOMENDACIONES.....	63
4.1.-CONCLUSIONES.....	64
4.2.-RECOMENDACIONES.....	64
5-.COMPETENCIAS .....	65
DESARROLLADAS Y/O .....	65
APLICADAS .....	65
5.1.-Competencias específicas.....	66
5.2.-Competencias Genéricas .....	66
5.2.1.-Instrumentales.....	66
5.2.2.-Sistémicas.....	66
5.2.3.-Interpersonales .....	66
6.-REFERENCIAS, BIBLIOGRÁFICAS Y VIRTUALES .....	67

## ***ÍNDICE DE ILUSTRACIONES***

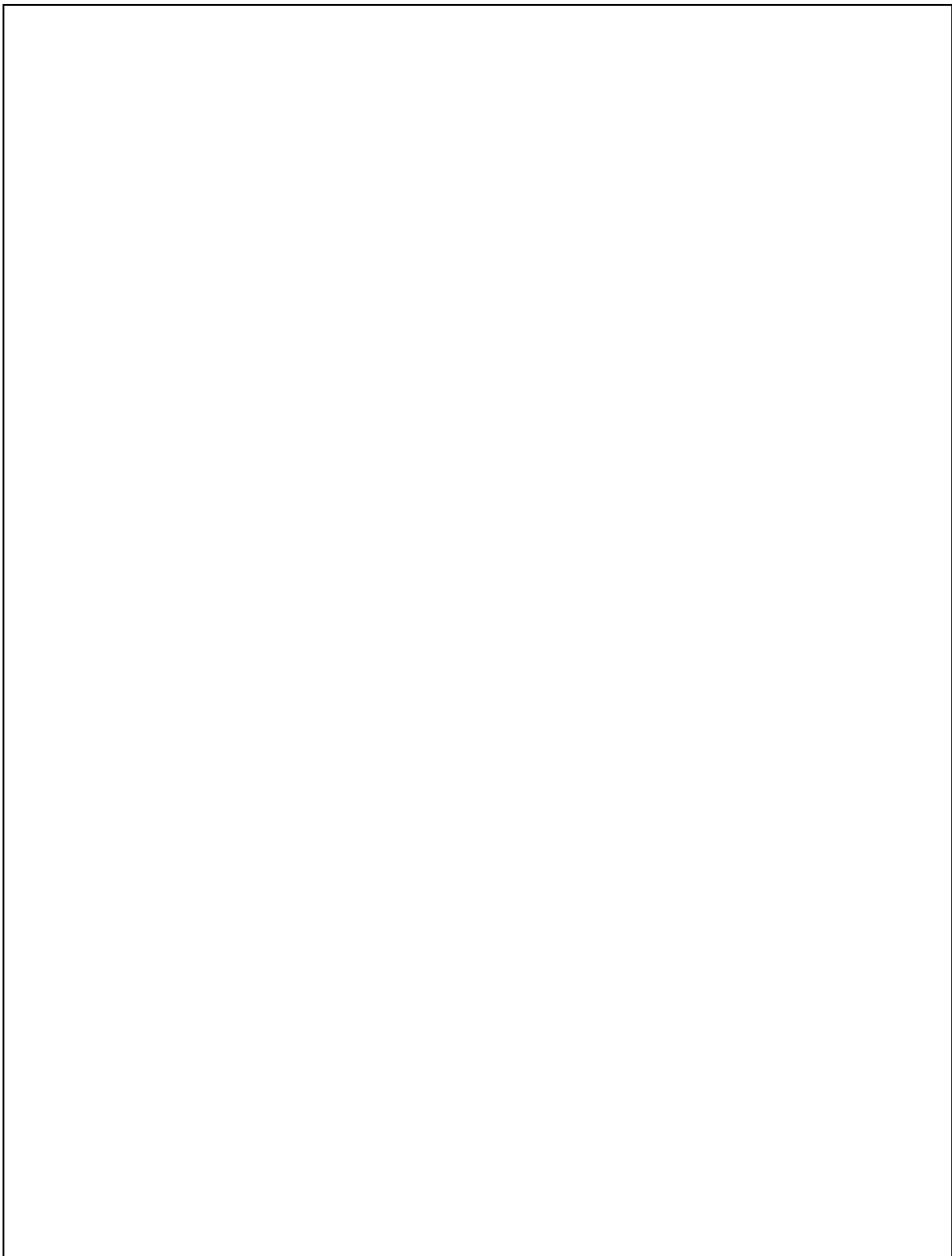
Ilustración 1 - Diagrama ULM .....	7
Ilustración 2 - Mockup interfaz principal .....	15
Ilustración 3 - Mockup interfaz de error .....	15
Ilustración 4 - Mockup interfaz de menú principal .....	16
Ilustración 5 - Mockup interfaz de ingresar .....	16
Ilustración 6 - Mockup interfaz de asignar .....	17
Ilustración 7 - Mockup interfaz de buscar .....	17
Ilustración 8 - Mockup interfaz de modificar .....	18
Ilustración 9 - Mockup interfaz de ingreso de usuario .....	18
Ilustración 10 - Mockup interfaz de buscar de usuario .....	19
Ilustración 11 - Mockup interfaz de modificar de usuario .....	19
Ilustración 12-código índice .....	21
Ilustración 13-código conexión .....	21
Ilustración 14-código bd 1 .....	22
Ilustración 15-código bd 2 .....	22
Ilustración 16-código bd 3 .....	22
Ilustración 17-código bd4 .....	23
Ilustración 18-código bd5 .....	23
Ilustración 19-código bd6 .....	23
Ilustración 20-código bd7 .....	24
Ilustración 21-código bd8 .....	24
Ilustración 22-código bd9 .....	24
Ilustración 23-código bd10 .....	25
Ilustración 24-código bd11 .....	25
Ilustración 25-código bd12 .....	25
Ilustración 26-código agregaradm .....	26
Ilustración 27-código insertad .....	26
Ilustración 28-interfaz de agregaradm .....	27
Ilustración 29-código de login.html .....	27
Ilustración 30-código loginadm1 .....	28
Ilustración 31-código loginadm2 .....	28
Ilustración 32-código loginadm3 .....	29
Ilustración 33-interfaz de login .....	29
Ilustración 34-interfaz de menú .....	30
Ilustración 35-interfaz menú monitores .....	30
Ilustración 36-interfaz menú cpus .....	31
Ilustración 37-interfaz menú impresoras .....	31
Ilustración 38-interfaz menú scanner .....	32

Ilustración 39-código insertacpu .....	32
Ilustración 40-código insertcpu .....	33
Ilustración 41-interfaz de insertar cpu .....	33
Ilustración 42-código insertamon .....	34
Ilustración 43-código insertmon .....	34
Ilustración 44-interfaz insertar monitores .....	35
Ilustración 45-código insertaimp .....	35
Ilustración 46-código insertimp .....	36
Ilustración 47-interfaz insertar impresora .....	36
Ilustración 48-código insertascan .....	37
Ilustración 49-código insertscan .....	37
Ilustración 50-interfaz de insertar scanner .....	38
Ilustración 51-código modimon.php .....	38
Ilustración 52-código modimon .....	39
Ilustración 53-código modimon 3 .....	39
Ilustración 54-interfaz de modimon .....	40
Ilustración 55-código modificarmon.php .....	40
Ilustración 56-código modificarmon 2 .....	41
Ilustración 57-interfaz modificarmon .....	41
Ilustración 58-código modi_mon.php .....	42
Ilustración 59-código modicpu.php .....	42
Ilustración 60-código modicpu 2 .....	42
Ilustración 61-código modicpu 3 .....	43
Ilustración 62-interfaz modicpu .....	43
Ilustración 63-código modificarcpu.php .....	44
Ilustración 64-código modificarcpu 2 .....	44
Ilustración 65-interfaz modificarcpu .....	45
Ilustración 66-código modi_cpu .....	45
Ilustración 67-código modiimp.php .....	46
Ilustración 68-código modiimp 2 .....	46
Ilustración 69-código modiimp 3 .....	46
Ilustración 70-interfaz modiimp .....	47
Ilustración 71-código modificarimp.php .....	47
Ilustración 72-código modificarimpr2 .....	48
Ilustración 73-interfaz modificarimp .....	48
Ilustración 74-código modi_impr .....	49
Ilustración 75-código modiscan.php .....	49
Ilustración 76-código modiscan2 .....	49
Ilustración 77-código modiscan3 .....	50
Ilustración 78-interfaz modiscan .....	50
Ilustración 79-código modificarscan .....	51

Ilustración 80-codigo modificarscan2 .....	51
Ilustración 81-interfaz modificarscan .....	52
Ilustración 82-codigo modi_scan.php .....	52
Ilustración 83-codigo buscarmonitor.....	53
Ilustración 84-codigo buscarmonitor 2.....	53
Ilustración 85-codigo buscarmonitor 3.....	53
Ilustración 86-interfaz buscarmonitor .....	54
Ilustración 87-codigo generadormonitor .....	54
Ilustración 88-pdf monitor.....	55
Ilustración 89-codigo buscarcpu.....	55
Ilustración 90-codigo buscarcpu 2.....	55
Ilustración 91-codigo buscarcpu 3.....	56
Ilustración 92-interfaz buscarcpu .....	56
Ilustración 93-codigo generadorcpu .....	57
Ilustración 94-pdf cpu.....	57
Ilustración 95-codigo buscarimp .....	58
Ilustración 96-codigo buscarimp 2 .....	58
Ilustración 97-codigo buscarimp 3 .....	58
Ilustración 98-interfaz buscarimp.....	59
Ilustración 99-codigo generadorimp.....	59
Ilustración 100-pdf impresora .....	60
Ilustración 101-codigo buscarscan .....	60
Ilustración 102-codigo buscarscan 2 .....	60
Ilustración 103-codigo buscarscan 3 .....	61
Ilustración 104-interfaz buscarscan.....	61
Ilustración 105-codigo generadorscan.....	62
Ilustración 106-pdf scanner .....	62

## ÍNDICE DE TABLAS

Tabla 1-personal involucrado .....	5
Tabla 2-definiciones y abreviaturas.....	6
Tabla 3-referencias .....	6
Tabla 4-característica del usuario 1 .....	7
Tabla 5-característica del usuario 2 .....	7
Tabla 6-RF01 .....	8
Tabla 7--RF02 .....	8
Tabla 8--RF03 .....	8
Tabla 9-RF04.....	9
Tabla 10--RF5 .....	9
Tabla 11--RF06 .....	9
Tabla 12--RF07 .....	10
Tabla 13-RFN01 .....	10
Tabla 14-RFN02 .....	10
Tabla 15-RFN03 .....	11
Tabla 16-RFN04 .....	11
Tabla 17-RFN05 .....	11
Tabla 18-RFN06 .....	12
Tabla 19-RFN07 .....	12
Tabla 20-RFN08 .....	12







## **1.-INTRODUCCIÓN**

### **1.1.-JUSTIFICACION**

En la U.M.F. #43 en el departamento de informática se tiene la necesidad de un sistema en el cual se requiere tener un mejor control de los equipos que conforman la clínica ya que no se tiene un control para poder dividir y ubicar las áreas en las que se encuentran los equipos. Este sistema ayudara a tener un control interno en el departamento y mejorara el rendimiento del mismo

### **1.2.-OBJETIVO**

#### **1.2.1.-OBJETIVO GENERAL**

Desarrollar un sistema de inventarios, el cual permita el almacenamiento, administración y la búsqueda de los equipos de una manera rápida y oportuna, utilizando su código de inventario; agilizando así los procesos administrativos del departamento de informática en la Unidad Médica Familiar #43.

#### **1.2.2.-OBJETIVOS ESPECÍFICOS**

- Diseñar y desarrollar una base de datos que permita el almacenamiento de la información.
- El sistema deberá permitir acceder a dos tipos de usuarios (administrador y usuario general).
- El usuario general solo podrá realizar consultas.
- El sistema contendrá dos apartados:

Administración de la información

Realización de consultas

- El apartado de la administración de la información contendrá lo siguiente:

Un módulo que permita registrar a los usuarios que podrán ingresar al sistema.

Un módulo en el cual se ingresara toda la información requerida de lo que se va a inventariar y se generara un código de inventario.

- Generar los diferentes reportes que requiera el departamento.
- Desarrollar informe técnico del sistema.

### ***1.3. - PROBLEMAS A RESOLVER***

Con este sistema será llevara a cabo la asignación y etiquetado de equipos para tener una mejor información de los equipos ya que no se tiene ningún control de información del equipo ni en qué área se encuentra porque se tiene una gran cantidad de equipos en toda la U.M.F #43. Actualmente solo se tiene un numero de inventario pero es nacional con lo que con este sistema se manejara ese mismo número de inventario y así se ayudara a mejorar el funcionamiento; dicho sistema más adelante se tratara de implementar en todas las áreas de la unidad y así tener un mejor control de todo lo que se pueda manejar .

## **2. PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADA**

## **2.1.- ERS (IEEE 830) – ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.**

### **2.1.1.-INTRODUCCIÓN**

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de inventarios del IMSS –Unidad médica #43. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830.

#### **2.1.1.2.-PROPÓSITO**

El propósito principal de este sistema es mejorar el funcionamiento del departamento de informática y no andar buscando equipo por equipo a la hora de buscar alguno, agilizar el proceso para la obtención de información acerca de algún equipo y generar el control interno con todo lo relacionado con el departamento de informática.

#### **2.1.1.3.-ALCANCE**

Sistema interno de inventario del IMSS - unidad #43 será el nombre que se le dará al sistema ya cuando esté acabado, a este sistema administrara la información relacionada con los equipos de cómputo, impresoras, scanner, servidores, switches entre otros equipos; también se tomara el número de inventario nacional para tener un control más adecuado, y se asignara a un área y a un empleado para tener responsables del mismo.

Tendrá muchos beneficios cuando halla supervisiones, auditorias o solo cuando se pida la información ya que no se tendrá que andar buscando el equipo físicamente solo se tendrá que ingresar al sistema y se obtendrá la información requerida, también uno de los objetivos es ese facilitar la búsqueda y almacenamiento de la información, las metas a cumplir son el mejoramiento del departamento para poder asignar adecuadamente los equipos en cada área y así tener una persona responsable del mismo.

#### **2.1. 1.4.-PERSONAL INVOLUCRADO**

<b>Nombre</b>	Ivan Artiño
<b>Rol</b>	Analista, diseñador y programador
<b>Categoría Profesional</b>	Practicante de Informática
<b>Responsabilidad</b>	Análisis de información, diseño y programación del SII#43
<b>Información de contacto</b>	ivanarti94@gmail.com

**Tabla 1 - Personal involucrado**

### ***2.1. 1.5.-DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS***

<b><i>Nombre</i></b>	<b><i>Descripción</i></b>
<b>Usuario</b>	Persona que usará el sistema para gestionar procesos
<b>SII#43</b>	Sistema de Inventarios del Imss –unidad medica #43

**Tabla 2** - Definiciones y abreviaturas

### ***2.1. 1.6.-REFERENCIAS***

<b>Título del Documento</b>	<b>Referencia</b>
Standard IEEE 830	IEEE

**Tabla 3** - Referencias

### ***2.1.1.7.-RESUMEN***

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

### ***2.1.2.-DESCRIPCIÓN GENERAL***

#### ***2.1.2.1.-PERSPECTIVA DEL PRODUCTO***

El sistema SII#43 será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida y eficaz, además se instalara en el servidor para que después de un tiempo se pudiera utilizar en más departamentos.

#### ***2.1.2.2.-FUNCIONALIDAD DEL PRODUCTO***



**Ilustración 1 - Diagrama ULM**

### **2.1.2.3.-CARACTERÍSTICAS DE LOS USUARIOS**

<b>Tipo de usuario</b>	Administrador
<b>Formación</b>	Ing. Sistemas computacionales
<b>Actividades</b>	Control y manejo del sistema en general

**Tabla 4-característica del usuario 1**

<b>Tipo de usuario</b>	Usuario
<b>Formación</b>	N/A
<b>Actividades</b>	Consulta y generación de código e impresión

**Tabla 5 - Característica del usuario 2**

### **2.1.2.4.-RESTRICCIONES.**

- Interfaz para ser usada con internet.
- Uso de Dominio (X)
- Lenguajes y tecnologías en uso: HTML, PHP,CSS.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

### **2.1.2.5.-SUPOSICIONES Y DEPENDENCIAS**

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

### **2.1.3.-REQUISITOS ESPECÍFICOS**

#### **2.1.3.1.-REQUERIMIENTOS FUNCIONALES**

<b>Identificación del requerimiento:</b>	RF01
--	------

<b>Nombre del Requerimiento:</b>	Identificación de Usuario.
<b>Características:</b>	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
<b>Descripción del requerimiento:</b>	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF05 RNF08
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 6 - RF01**

<b>Identificación del requerimiento:</b>	RF02
<b>Nombre del Requerimiento:</b>	Registrar usuarios
<b>Características:</b>	Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema.
<b>Descripción del requerimiento:</b>	El sistema permitirá al usuario (Administrador y usuario general) registrarse. El usuario debe suministrar datos como: Clave, Nombre, Apellido, E-mail, Usuario y Password.
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF05 RNF08
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 7 - RF02**

<b>Identificación del requerimiento:</b>	RF3
<b>Nombre del Requerimiento:</b>	Ingresar Información.
<b>Características:</b>	El sistema ofrecerá al administrador poder ingresar datos de los diferentes tipos de equipos que tendrá para inventariar en la unidad
<b>Descripción del requerimiento:</b>	El sistema tendrá que pedir ingresar datos para poder funcional el sistema
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF08
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 8 - RF03**

<b>Identificación del requerimiento:</b>	RF04
--	------



<b>Nombre del Requerimiento:</b>	Consultar Información.
<b>Características:</b>	El sistema ofrecerá al usuario información general acerca de los equipos que estén dentro del sistema
<b>Descripción del requerimiento:</b>	Muestra información dependiendo de lo que solicite el usuario
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF08
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 9 - RF04**

<b>Identificación del requerimiento:</b>	RF05
<b>Nombre del Requerimiento:</b>	Modificar Información
<b>Características:</b>	El sistema permitirá solo al administrador modificar los datos de los equipos dentro del sistema
<b>Descripción del requerimiento:</b>	Permite al administrador modificar datos de los equipos
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF05
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 10 - RF5**

<b>Identificación del requerimiento:</b>	RF06
<b>Nombre del Requerimiento:</b>	Generar etiquetas.
<b>Características:</b>	El sistema permitirá generar etiquetas.
<b>Descripción del requerimiento:</b>	Permite al administrador imprimir etiquetas donde vendrá información de los equipos
<b>Requerimiento NO funcional:</b>	RNF01 RNF02 RNF05
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 11 - RF06**

<b>Identificación del requerimiento:</b>	RF07
<b>Nombre del Requerimiento:</b>	Auditoría del sistema

<b>Características:</b>	Garantizar las soluciones de problemas existentes mediante la utilización del sistema.
<b>Descripción del requerimiento:</b>	Evaluar y analizar los procesos del sistema, proponiendo solución de problemas existentes dentro del sistema utilizado.
<b>Requerimiento NO funcional:</b>	RNF03 RNF04 RNF05 RNF06 RNF07 RNF08
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 12 - RF07**

### 2.1.3.2.-REQUERIMIENTOS NO FUNCIONALES

<b>Identificación del requerimiento:</b>	RNF01
<b>Nombre del Requerimiento:</b>	Interfaz del sistema.
<b>Características:</b>	El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
<b>Descripción del requerimiento:</b>	El sistema debe tener una interfaz de uso intuitiva y sencilla.
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 13 - RNF01**

<b>Identificación del requerimiento:</b>	RNF02
<b>Nombre del Requerimiento:</b>	Facilidad de manejo del sistema.
<b>Características:</b>	La interfaz del usuario deberá de presentar un sistema de ayuda para que los mismos usuarios del sistema se les faciliten el trabajo en cuanto al manejo del sistema.
<b>Descripción del requerimiento:</b>	La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones).
<b>Prioridad del requerimiento:</b>	Alta

**Tabla 14 - RNF02**

<b>Identificación del requerimiento:</b>	RNF03
<b>Nombre del Requerimiento:</b>	Mantenimiento.

<b>Características:</b>	El sistema deberá de tener un manual de instalación y manual de usuario para facilitar los mantenimientos que serán realizados por el administrador.
<b>Descripción del requerimiento:</b>	El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 15 - RFN03**

<b>Identificación del requerimiento:</b>	RNF04
<b>Nombre del Requerimiento:</b>	Diseño de la interfaz a la característica de la web.
<b>Características:</b>	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.
<b>Descripción del requerimiento:</b>	La interfaz de usuario debe ajustarse a las características de la web de la institución, dentro de la cual estará incorporado el sistema de gestión de procesos y el inventario.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 16 - RFN04**

<b>Identificación del requerimiento:</b>	RNF05
<b>Nombre del Requerimiento:</b>	Desempeño
<b>Características:</b>	El sistema garantizara a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a esta misma.
<b>Descripción del requerimiento:</b>	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 17 - RFN05**

<b>Identificación del requerimiento:</b>	RNF06
--	-------

<b>Nombre del Requerimiento:</b>	Nivel de Usuario
<b>Características:</b>	Garantizara al usuario el acceso de información de acuerdo al nivel que posee.
<b>Descripción del requerimiento:</b>	Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 18 - RFN06**

<b>Identificación del requerimiento:</b>	RNF07
<b>Nombre del Requerimiento:</b>	Confiabilidad del sistema.
<b>Características:</b>	El sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana. Ya que es un sistema web diseñado para la carga de datos y consulta de la información ingresada.
<b>Descripción del requerimiento:</b>	La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un aviso de la posible falla en cualquiera de sus áreas del sistema.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 19 - RFN07**

<b>Identificación del requerimiento:</b>	RNF08
<b>Nombre del Requerimiento:</b>	Seguridad en información
<b>Características:</b>	El sistema garantizara a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
<b>Descripción del requerimiento:</b>	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
<b>Prioridad del requerimiento:</b> Alta	

**Tabla 20 - RFN08**

#### ***2.1.4.-Requisitos comunes de las interfaces***

##### ***2.1.4.1.-Interfaces de usuario***

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

#### ***2.1.4.2.-Interfaces de hardware***

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Adaptadores de red.
- Procesador de 1.66GHz o superior.
- Memoria RAM mínima 2gb.
- Mouse.
- Teclado.

#### ***2.1.4.3.-Interfaces de software***

- Sistema Operativo: Windows XP o superior.
- Explorador: Mozilla, Chrome.

#### ***2.1.4.4.-Interfaces de comunicación***

Los servidores, clientes y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible.

### ***2.2.-Requisitos funcionales***

#### ***2.2.1.-Requisito Funcional 01- Identificación de Usuarios***

Los usuarios deberán identificarse para acceder a cualquier parte del sistema. El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.

#### ***2.2.2.-Requisito Funcional 02 - Registrar Usuarios***

El sistema permitirá al usuario (Administrador y usuario) registrarse. El usuario debe suministrar datos como: Matricula, Nombre, Apellido, E-mail, Usuario y Contraseña.

#### ***2.2.3.-Requisito Funcional 03 - Ingresar información***

El administrador podrá ingresar toda la información requerida por el sistema para poder asignar los equipos para alguna área específica

#### ***2.2.4.-Requisito Funcional 04 - Consulta de información***

El administrador y el usuario podrá consultar la información dependiendo del nivel de accesibilidad del sistema

#### ***2.2.5.-Requisito Funcional 05 - Modificar información***

El administrador solo podrá modificar la información del sistema.

#### ***2.2.6.-Requisito Funcional 06 - Gestionar etiquetas***

Permite al administrador imprimir etiquetas para cualquier equipo de la unidad.

#### ***2.2.6.-Requisito Funcional 07 - Auditoría***

Evaluar y analizar los procesos del sistema, proponiendo solución de problemas existentes dentro del sistema utilizado

### ***2.3.-Requisitos no funcionales***

#### ***2.3.1.- Requisitos no funcionales - Requisitos de rendimiento***

Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

#### ***2.3.3.-Requisitos no funcionales - Seguridad***

Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta. Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas. Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

#### ***2.3.3.- Requisitos no funcionales - Fiabilidad***

El sistema debe tener una interfaz de uso intuitiva y sencilla. La interfaz de usuario debe ajustarse a las características de la web de la institución, dentro de la cual estará incorporado el sistema de gestión de procesos y el inventario

#### ***2.3.4- Requisitos no funcionales - Disponibilidad***

La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

#### ***2.3.5.- Requisitos no funcionales - Mantenibilidad***

El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.

La interfaz debe estar complementada con un buen sistema de ayuda (la administración puede recaer en personal con poca experiencia en el uso de aplicaciones informáticas).

#### ***2.3.6.- Requisitos no funcionales - Portabilidad***

El sistema será implantado bajo la plataforma de Windows.

## 2.4.-Mockups

### 2.4.1. Mockups de Administrador



**Ilustración 2 - Mockup interfaz principal**

Esta es la interfaz principal para ingresar al sistema ahí se ingresará la matrícula del personal que es única para cada trabajador y la contraseña asignada, la cual tendrá el logotipo del IMSS , el nombre del sistema ,2 textbox uno para la matrícula y el otro para la contraseña y un botón para ingresar al sistema.



**Ilustración 3 - Mockup interfaz de error**

Esta es la interfaz de error al ingresar matrícula o contraseña incorrecta, la cual manda un mensaje de error que se ingresaron mal los datos tendrá un botón y el mensaje avisando el error.



**Ilustración 4 - Mockup interfaz de menú principal**

Esta es la interfaz del menú principal en el cual se podrá ingresar, asignar, buscar o modificar alguna información acerca de los equipos, la cual tendrá el menú en el cual se podrá seleccionar la opción que quisiera realizar ya sea ingresar, asignar, buscar o modificar también tendrá el logo del IMSS y el nombre del sistema .



**Ilustración 5 - Mockup interfaz de ingresar**

Esta es la interfaz para ingresar los datos al sistema todo lo relacionado con los equipos, la interfaz tendrá 3 textbox los cuales se podrán ingresar el número de inventario, el número de serie y los comentarios, 2 combobox en el que se pueda seleccionar la marca y el modelo, un botón para guardar, el logo del IMSS y el nombre del sistema.





**Ilustración 6 - Mockup interfaz de asignar**

Esta es la interfaz para poder asignar los equipos al departamento correspondiente y al personal correspondiente, la cual tendrá 3 textbox en los que se podrá ingresar el número de inventario, los encargados y donde se muestran los resultados, un combobox donde seleccionaremos el departamento o área donde se va a asignar ,2 botones uno para buscar y el otro para guardar lo modificado, el logo del IMSS y el nombre del sistema.



**Ilustración 7 - Mockup interfaz de buscar**

Esta es la interfaz donde se pueden buscar los datos de los equipos y también en esta interfaz se podrá imprimir la información del mismo, la cual tendrá 2 textbox uno para ingresar el número de inventario y el otro para la información del equipo, 2 botones uno para imprimir y el otro para buscar, el logo del IMSS y el nombre del sistema



**Ilustración 8 - Mockup interfaz de modificar**

Esta es la interfaz donde se podrá modificar la información de los equipos que se puedan sesionar o editar su información, la cual tendrá 4 textbox el primero será para el numero de inventario, el segundo será para el número de serie, el tercero será para los encargados y el cuarto para los comentarios, también se tendrá 3 combobox en los cuales se podrá seleccionar la marca, el modelo y el departamento y en la parte inferior el botón de guardar todos los cambios.

#### 2.4.2.-Mockups de Usuario



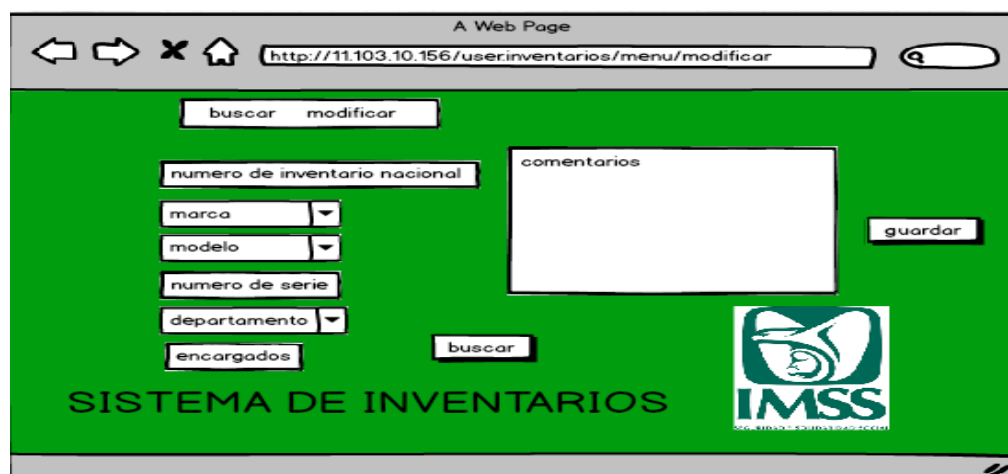
**Ilustración 9 - Mockup interfaz de ingreso de usuario**

Esta es la interfaz donde el usuario general ingresara al sistema y podrá solo hacer lo que le corresponda, la cual tendrá el logotipo del IMSS, el nombre del sistema ,2 textbox uno para la matrícula y el otro para la contraseña y un botón para ingresar al sistema.



**Ilustración 10 - Mockup interfaz de buscar de usuario**

Esta es la interfaz donde el usuario pueden buscar los datos de los equipos y también en esta interfaz se podrá imprimir la información del mismo, la cual tendrá 2 textbox uno para ingresar el número de inventario y el otro para la información del equipo, 2 botones uno para imprimir y el otro para buscar, el logo del IMSS y el nombre del sistema



**Ilustración 11 - Mockup interfaz de modificar de usuario**

Esta es la interfaz donde el usuario podrá modifica la información de los equipos que se puedan sesionar, la cual tendrá 4 textbox el primero será para el numero de inventario, él segundo será para el número de serie, el tercero será para los encargados y el cuarto para los comentarios, también se tendrá 3 combobox en los cuales se podrá seleccionar la marca, el modelo y el departamento y en la parte inferior el botón de guardar todos los cambios y el de buscar el equipo.

### **3.-RESULTADOS, PLANOS, GRÁFICAS, PROTOTIPOS, MAQUETAS Y PROGRAMAS**

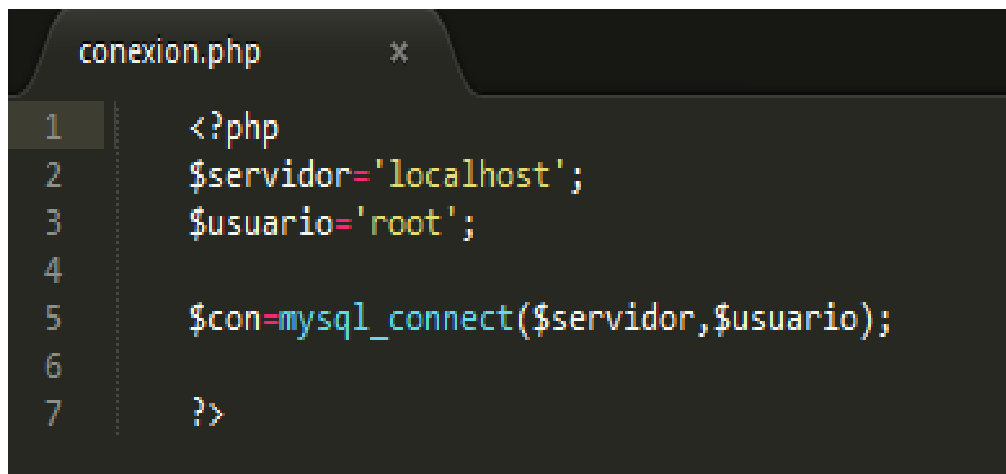
### 3.1.-Resultados del proyecto

A screenshot of a code editor showing the contents of a file named 'index.php'. The code is written in PHP and includes line numbers from 1 to 10. It starts with a PHP opening tag, followed by a 'require' statement for 'conexion.php'. Then, it uses 'mysql\_select\_db' to select a database named 'informatica'. An 'if' statement checks if the database selection was successful; if true, it calls 'header' to redirect to 'login.html', and if false, it calls 'header' to redirect to 'bd.php'. The code ends with a PHP closing tag.

```
1 <?php
2 require('conexion.php');
3 $db = mysql_select_db ( "informatica");
4 if ($db ) {
5     header("Location:login.html");
6 }
7 else {
8     header("Location:bd.php");
9 }
10 ?>
```

**Ilustración 12-codigo índice**

En la imagen se muestra el código del índice en el cual requiere de conexión con el servidor y que está en conexion.php, se selecciona la base y se direcciona si esta creada a login.html de lo contrario se direccionara a crearla a bd.php.

A screenshot of a code editor showing the contents of a file named 'conexion.php'. The code is written in PHP and includes line numbers from 1 to 7. It starts with a PHP opening tag, followed by two assignments: '\$servidor' set to 'localhost' and '\$usuario' set to 'root'. Then, it uses 'mysql\_connect' to establish a connection with the server using the provided credentials. The code ends with a PHP closing tag.

```
1 <?php
2 $servidor='localhost';
3 $usuario='root';
4
5 $con=mysql_connect($servidor,$usuario);
6
7 ?>
```

**Ilustración 13-codigo conexión**

En la imagen se muestra del código de la conexión requerida en el índice que es el nombre del servidor y el usuario del servidor como también la conexión que se requiere para mysql y entrar al servidor.

```

bd.php
1  <?php
2      require('conexion.php');
3      $db = mysql_select_db ( "informatica");
4
5      $create=mysql_query(("CREATE DATABASE `informatica`"),$con);

```

**Ilustración 14-codigo bd 1**

En la imagen se muestra el código donde se requiere conexión y se selecciona la base de datos, si no está creada la base de datos se crea la base de datos llamada informática.

```

mysql_select_db('informatica',$con);
$tabla="CREATE TABLE IF NOT EXISTS `admin` (
    `matricula` varchar(10) NOT NULL,
    `nombre` varchar(55) NOT NULL,
    `puesto` varchar(55) NOT NULL,
    `email` varchar(55) NOT NULL,
    `contraladm` varchar(15) NOT NULL,
    `contra2adm` varchar(15) NOT NULL,
    PRIMARY KEY (`matricula`))";

```

**Ilustración 15-codigo bd 2**

En la imagen se muestra la creación de tabla admin donde las columnas serán matricula varchar 10, nombre varchar 55, puesto varchar 55, email varchar 55, contraladm varchar 15 y contra2adm varchar 15 en la tabla creada la columna de matrícula será llave primaria.

```

$crear_tabla=mysql_query($tabla,$con);
$tabla="CREATE TABLE IF NOT EXISTS `usuario` (
    `matricula` varchar(10) NOT NULL,
    `nombre` varchar(55) NOT NULL,
    `email` varchar(55) NOT NULL,
    `contraladm` varchar(55) NOT NULL,
    `contra2adm` varchar(55) NOT NULL,
    PRIMARY KEY (`matricula`))";

```

**Ilustración 16-codigo bd 3**

En la imagen se muestra la creación de tabla usuario donde las columnas serán matricula varchar 10, nombre varchar 55, puesto varchar 55, email varchar 55, contraladm varchar 15 y contra2adm varchar 15 en la tabla creada la columna de matrícula será llave primaria.

```

$crear_tabla=mysql_query($tabla,$con);
$tabla="CREATE TABLE IF NOT EXISTS `monitores` (
  `cod_inv_mon` varchar(15) NOT NULL,
  `no_serie` varchar(15) NOT NULL,
  `modelo` varchar(8) NOT NULL,
  `departamento` varchar(10) NOT NULL,
  `responsable` varchar(55) NOT NULL,
  `comentario` varchar(55) NOT NULL,
  PRIMARY KEY (`cod_inv_mon`));";

```

**Ilustración 17-código bd4**

En la imagen se muestra la creación de tabla monitores donde las columnas serán cod\_inv\_mon varchar 15, no\_serie varchar 15, modelo varchar 8, departamento varchar 10, responsable varchar 55 y comentario varchar 55 en la tabla creada la columna de cod\_inv\_mon será llave primaria.

```

$crear_tabla=mysql_query($tabla,$con);
$tabla="CREATE TABLE IF NOT EXISTS `cpus` (
  `cod_inv_cpu` varchar(15) NOT NULL,
  `noserie` varchar(15) NOT NULL,
  `modelo` varchar(8) NOT NULL,
  `departamento` varchar(10) NOT NULL,
  `responsable` varchar(55) NOT NULL,
  `direccionip` varchar(55) NOT NULL,
  `nombre_equipo` varchar(55) NOT NULL,
  `cuenta_equipo` varchar(55) NOT NULL,
  `comentario` varchar(55) NOT NULL,
  PRIMARY KEY (`cod_inv_cpu`));";

```

**Ilustración 18-código bd5**

En la imagen se muestra la creación de tabla cpus donde las columnas serán cod\_inv\_cpu varchar 15, noserie varchar 15, modelo varchar 8, departamento varchar 10, responsable varchar 55, direccionip varchar 55,nombre\_equipo varchar 55,cuenta\_equipo varchar 55, comentario varchar 55 en la tabla creada la columna de cod\_inv\_cpu será llave primaria.

```

$crear_tabla=mysql_query($tabla,$con);
$tabla="CREATE TABLE IF NOT EXISTS `impresora` (
  `cod_inv_imp` varchar(15) NOT NULL,
  `no_serie` varchar(15) NOT NULL,
  `modelo` varchar(8) NOT NULL,
  `departamento` varchar(10) NOT NULL,
  `responsable` varchar(55) NOT NULL,
  `comentario` varchar(55) NOT NULL,
  PRIMARY KEY (`cod_inv_imp`));";

```

**Ilustración 19-código bd6**

En la imagen se muestra la creación de tabla impresora donde las columnas serán cod\_inv\_imp varchar 15, no\_serie varchar 15, modelo varchar 8, departamento varchar 10, responsable varchar 55 y comentario varchar 55 en la tabla creada la columna de cod\_inv\_imp será llave primaria.

```

$crear_tabla=mysql_query($tabla,$con);
$tabla="CREATE TABLE IF NOT EXISTS `scanner` (
  `cod_inv_esc` varchar(15) NOT NULL,
  `no_serie` varchar(15) NOT NULL,
  `modelo` varchar(8) NOT NULL,
  `departamento` varchar(10) NOT NULL,
  `responsable` varchar(55) NOT NULL,
  `comentario` varchar(55) NOT NULL,
  PRIMARY KEY (`cod_inv_esc`))";
$crear_tabla=mysql_query($tabla,$con);

```

**Ilustración 20-código bd7**

En la imagen se muestra la creación de tabla scanner donde las columnas serán cod\_inv\_esc varchar 15, no\_serie varchar 15, modelo varchar 8, departamento varchar 10, responsable varchar 55 y comentario varchar 55 en la tabla creada la columna de cod\_inv\_esc será llave primaria.

```

mysql_select_db('informatica',$con);
$add="CREATE PROCEDURE `informatica`.`Insertar_admin` (in matricula varchar(10),
  nombre varchar(55), puesto varchar(55), email varchar(55), contraladm varchar(15), contra2adm varchar(15))
BEGIN
INSERT INTO admin VALUES (matricula,nombre,puesto,email,contraladm,contra2adm);
END";
$crear_add=mysql_query($add,$con);

```

**Ilustración 21-código bd8**

En la imagen se muestra el método para ingresar los datos del administrador llamado insertar\_admin donde se llamara y convertirán los datos ingresados para agregarlos en la base de datos, también en la imagen se muestra donde se selecciona la base de datos y la tabla donde se ingresaran los datos convertidos con el método.

```

mysql_select_db('informatica',$con);
$add="CREATE PROCEDURE `informatica`.`Insertar_cpu` (in cod_inv_cpu varchar(15), noserie varchar(15), modelo varchar(8), departamento varchar(10),
  responsable varchar(55), direccionip varchar(55), nombre_equipo varchar(55), cuenta_equipo varchar(55), comentario varchar(55))
BEGIN
INSERT INTO cpus VALUES (cod_inv_cpu,noserie,modelo,departamento,responsable,direccionip,nombre_equipo,cuenta_equipo,comentario);
END";
$crear_add=mysql_query($add,$con);

```

**Ilustración 22-código bd9**

En la imagen se muestra el método para ingresar los datos de los cpus llamado insertar\_cpu donde se llamara al método y convertirá los datos ingresados para agregarlos en la base de datos, también en la imagen se muestra donde se selecciona la base de datos y la tabla donde se ingresaran los datos convertidos con el método.



```
mysql_select_db('informatica',$con);
$add="CREATE PROCEDURE `informatica`.`Insertar_mon` (in cod_inv_mon varchar(15),no_serie varchar(15),modelo varchar(8), departamento varchar(10),
responsable varchar(55), comentario varchar(55))
BEGIN
INSERT INTO monitores VALUES (cod_inv_mon,no_serie,modelo,departamento,responsable,comentario);
END";
$crear_add=mysql_query($add,$con);
```

**Ilustración 23-codigo bd10**

En la imagen se muestra el método para ingresar los datos de los monitores llamado insertar\_mon donde se llamara al método y convertirá los datos ingresados para agregarlos en la base de datos, también en la imagen se muestra donde se selecciona la base de datos y la tabla donde se ingresaran los datos convertidos con el método.

```
mysql_select_db('informatica',$con);
$add="CREATE PROCEDURE `informatica`.`Insertar_imp` (in cod_inv_imp varchar(15),no_serie varchar(15),modelo varchar(8), departamento varchar(10),
responsable varchar(55), comentario varchar(55))
BEGIN
INSERT INTO impresora VALUES (cod_inv_imp,no_serie,modelo,departamento,responsable,comentario);
END";
$crear_add=mysql_query($add,$con);
```

**Ilustración 24-codigo bd11**

En la imagen se muestra el método para ingresar los datos de las impresoras llamado insertar\_imp donde se llamara al método y convertirá los datos ingresados para agregarlos en la base de datos, también en la imagen se muestra donde se selecciona la base de datos y la tabla donde se ingresaran los datos convertidos con el método.

```
mysql_select_db('informatica',$con);
$add="CREATE PROCEDURE `informatica`.`Insertar_escan` (in cod_inv_esc varchar(15),no_serie varchar(15),modelo varchar(8), departamento varchar(10),
responsable varchar(55), comentario varchar(55))
BEGIN
INSERT INTO escaner VALUES (cod_inv_esc,no_serie,modelo,departamento,responsable,comentario);
END";
$crear_add=mysql_query($add,$con);
```

**Ilustración 25-codigo bd12**

En la imagen se muestra el método para ingresar los datos de los scanner llamado insertar\_escan donde se llamara al método y convertirá los datos ingresados para agregarlos en la base de datos, también en la imagen se muestra donde se selecciona la base de datos y la tabla donde se ingresaran los datos convertidos con el método.

```

<div id="center">
  <div id="logo">
    
  </div>
  <div id="titulo">
    <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
  </div>
  <div id="menu">
  </div>
</div>
<form method="POST" action="insertad.php" class="contacto" enctype="multipart/form-data">

  <div><label>MATRICULA</label><input type="text" id="clv" name="clv"></div>
  <div><label>NOMBRE</label><input type="text" id="nk" name="nk"></div>
  <div><label>PUESTO</label><input type="text" id="usr" name="usr"></div>
  <div><label>E-mail:</label><input type="email" id="em" name="em"></div>
  <div><label>CONTRASEÑA</label><input type="password" id="nom" name="nom"></div>
  <div><label>REPETIR CONTRASEÑA</label><input type="password" id="tp" name="tp" ></div>
  <div><input type="submit" name="submit" id="boton" value="Guardar"/>
  </div>
</form>

```

**Ilustración 26-código agregaradm**

En la imagen se muestra el código de la interfaz para ingresar al administrador general del sistema en donde se le piden los datos como matrícula, nombre, puesto, email, contraseña1 y contraseña2, donde esos datos se envían a un php llamado insertad, la interfaz incluye el logo del IMSS, el nombre del sistema ,6 textbox donde se ingresan los datos y un botón de guardar donde envía los datos insertados al php insertad.

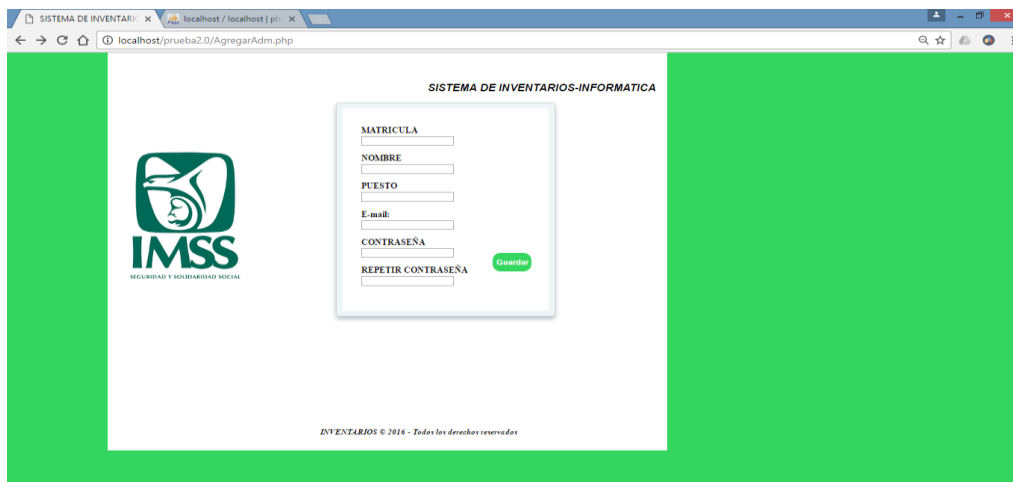
```

<?php
$matricula = $_POST["clv"];
$nombre = $_POST["nk"];
$puesto = $_POST["usr"];
$email = $_POST["em"];
$contraseña = $_POST["nom"];
$contraseña1 = $_POST["tp"];
require('conexion.php');
mysql_select_db('informatica');
$tabla=mysql_query("CALL Insertar_admin('{ $matricula}', '{ $nombre}',
| '{ $puesto}', '{ $email}', '{ $contraseña}', '{ $contraseña1}'"),$con);
header('refresh:3; url=login.html');
?>
</form>

```

**Ilustración 27-código insertad**

En la imagen se muestra el código en donde se recibe la información del administrador y se convierten los datos, después se selecciona la base de datos en la que se van a ingresar y se llama el método insertar\_admin en donde se creó anteriormente y se direcciona a login.html.



**Ilustración 28-interfaz de agregaradm**

En la imagen se muestra la interfaz de agregaradm donde se piden los datos del administrador que se van a guardar en la base de datos, que posterior se le van a pedir para ingresar al sistema. La interfaz contiene el logo del IMSS ,5 textbox para poder ingresar los datos ,5 label donde indica los datos a ingresar, el nombre del sistema y el botón para guardar los datos en la base de datos.

```
<div id="logo">
  
</div>
<div id="titulo">
  <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
</div>
<div class="container">
  <section id="content">
    <form method="POST" action="loginadm.php">
      <h2>Login</h2>
      <div>
        <input type="text" placeholder="MATRICULA" id="usu" name="usu"></input>
      </div>
      <div>
        <input type="password" placeholder="CONTRASEÑA" id="pwd" name="pwd"></input>
      </div>
      <div>
        <input type="submit" value="Log in" ></input>
      </div>
    </form>
  </section>
</div>
```

**Ilustración 29-código de login.html**

En la imagen se muestra el código de login para ingresar al sistema por ello se pide la matrícula y la contraseña que se va a mandar a loginadm.php para verificar si están correcto lo que se ingresó y si coincide con lo que está en la base de datos se podrá ingresar sin problemas de lo contrario hasta ahí podrá ingresar sin poder ingresar al sistema.

```

loginadm.php
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Registro</title>
6      <link rel="stylesheet" href="css/datos.css">
7  </head>
8  <body>
9  <div id="espectro">
10 <?php
11 $usu= $_POST["usu"];
12 $pwd= $_POST["pwd"];
13
14 require('conexion.php');
15 mysql_select_db('informatica',$con);
16

```

**Ilustración 30-código loginadm1**

En la imagen se muestra la conversión de los datos ingresados en el login también se muestra la conexión y la selección de la base de datos para poder verificar los datos ingresados más adelante.

```

    $arrojado=mysql_query("SELECT matricula FROM admin
WHERE matricula='$usu'",$con) or
die("Problemas en el select:".mysql_error());
$resultado = mysql_fetch_array($arrojado);
$valor = $resultado['matricula'];
if($valor == $usu){
    $arrojado=mysql_query("SELECT contraladm FROM admin
WHERE contraladm='$pwd'",$con) or
die("Problemas en el select:".mysql_error());
$resultado = mysql_fetch_array($arrojado);
$valor = $resultado['contraladm'];
if($valor == $pwd){
echo '<div id="usuario" id="usuario">
<br><br><br>
<strong><IMG SRC="images/paloma.jpg" ALIGN=center WIDTH=178 HEIGHT=200 BORDER=2 VSPACE=3 HSPACE=3 ALT="Correcto"></IMG>
<p ALIGN=center><font FACE="impact" SIZE=11 COLOR="#75b6dc">INICIO DE SESION CORRECTO!</p>
<p ALIGN=center><font FACE="impact" SIZE=5 COLOR="#75b6dc">Por favor espere.....Ser&aacute; redireccionado automaticamente</p></div>
header('refresh:3; url=menu.html');

```

**Ilustración 31-código loginadm2**

En esta imagen se muestra validación de la matrícula que este ingresada correctamente con los datos ingresados y la contraseña que estén también correcta para así ingresar al sistema, se manda un mensaje de que esta correcto los datos y se direcciona a menú.html.

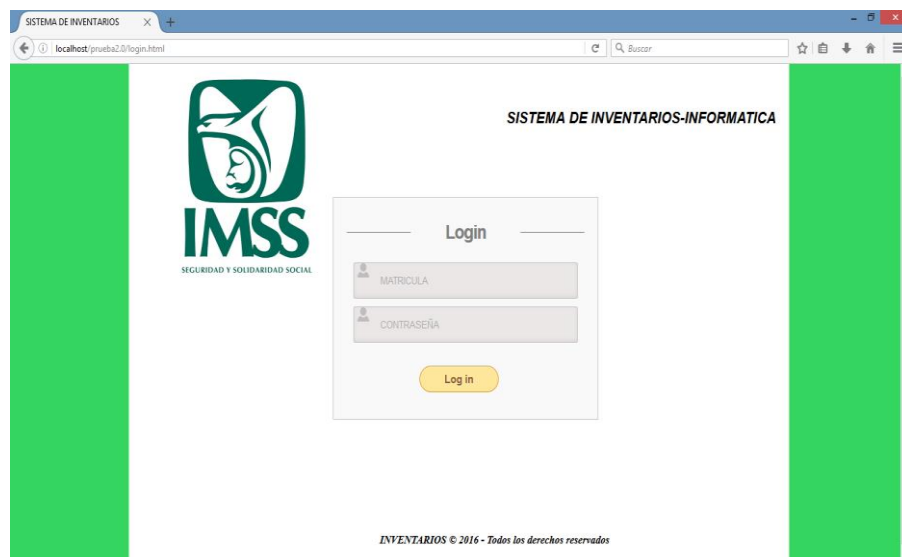
```

}
else {
echo '<div class="alert alert=error">
<br><br><br>
<strong><IMG SRC="images/error.jpg" ALIGN=center WIDTH=178 HEIGHT=200 BORDER=2 VSPACE=3 HSPACE=3 ALT="Correcto"></IMG>
<p ALIGN=center><font FACE="impact" SIZE=11 COLOR="#75b6dc">;USUARIO O CONTRASE&ntilde;A ERRONEA!</p>
<p ALIGN=center><font FACE="impact" SIZE=5 COLOR="#75b6dc">Por favor espere.....Ser&aacute; redireccionado automaticamente</p>
</div>';
header('refresh:3; url=login.html');
}
}
else {
echo '<div class="alert alert=error">
<br><br><br>
<strong><IMG SRC="images/error.jpg" ALIGN=center WIDTH=178 HEIGHT=200 BORDER=2 VSPACE=3 HSPACE=3 ALT="Correcto"></IMG>
<p ALIGN=center><font FACE="impact" SIZE=11 COLOR="#75b6dc">;USUARIO O CONTRASE&ntilde;A ERRONEA!</p>
<p ALIGN=center><font FACE="impact" SIZE=5 COLOR="#75b6dc">Por favor espere.....Ser&aacute; redireccionado automaticamente</p>
</div>';
header('refresh:3; url=login.html');
}
}
?>

```

**Ilustración 32-código loginadm3**

En la imagen se muestra los mensajes de error en los cuales se indica que se ingresó mal la matrícula o la contraseña se volverá a direccionar a login.html de nuevo para poder ingresar al sistema correctamente de lo contrario seguirá saliendo el mensaje hasta que ingrese correctamente los datos.



**Ilustración 33-interfaz de login**

En la imagen se muestra la interfaz de login donde se ingresa la matricula del usuario y la contraseña que anteriormente se ingresa en el registro anterior. El cual tiene que coincidir con lo que se ingresó en la base de datos y se direccionara a loginadm.php. la interfaz tiene el logo del IMSS ,el nombre del sistema ,la pantalla de login que es el título,2 textbox donde se ingresara la matrícula y la contraseña y un botón de login para ingresar al sistema .



**Ilustración 34-interfaz de menú**

En esta imagen se muestra el menú principal del sistema en el cual se podrá elegir entre las imágenes lo que quisiera acceder ya sea monitores, cpus, impresoras y scanner para poder inventariar y al picarle se direccionara a su respectivo menú, en la parte de arriba está el botón de cerrar sesión en el cual se sale del sistema y lo direcciona a login de nuevo.



**Ilustración 35-interfaz menú monitores**

En esta interfaz se muestra el menú de monitores en el cual el primer icono se ingresara la información, en el segundo icono se modificara la información, en el tercer icono se buscara información y por último el cuarto icono direccionara al menú principal, en la parte de arriba también está el botón de cerrar sesión en el cual se sale del sistema y lo direcciona a login de nuevo.



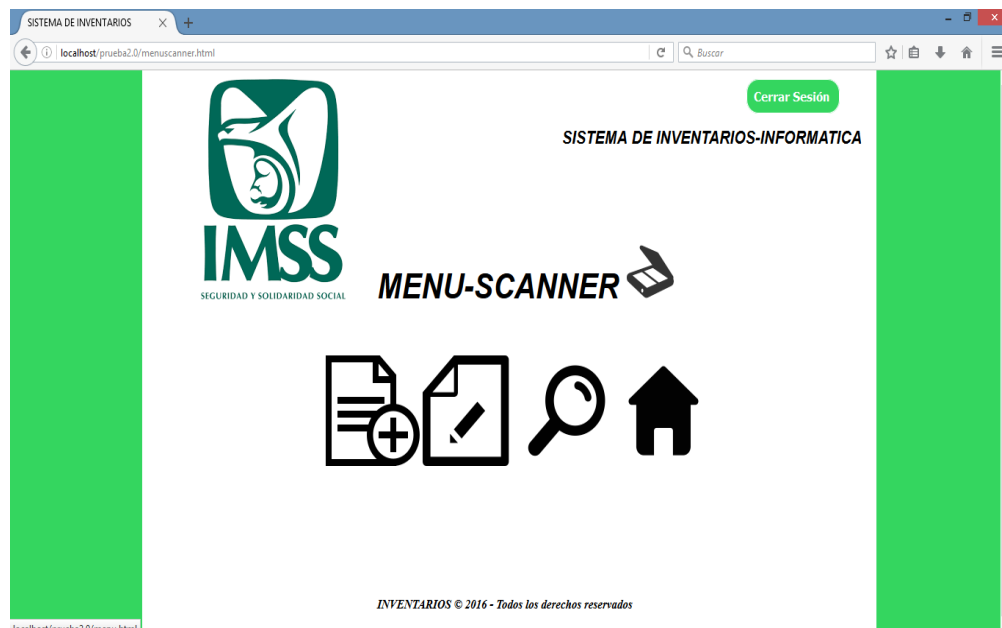
**Ilustración 36-interfaz menú cpus**

En esta interfaz se muestra el menú de cpus en el cual el primer icono se ingresara la información, en el segundo icono se modificara la información, en el tercer icono se buscara información y por último el cuarto icono direccionara al menú principal, en la parte de arriba también está el botón de cerrar sesión en el cual se sale del sistema y lo direcciona a login de nuevo.



**Ilustración 37-interfaz menú impresoras**

En esta interfaz se muestra el menú de impresoras en el cual el primer icono se ingresara la información, en el segundo icono se modificara la información, en el tercer icono se buscara información y por último el cuarto icono direccionara al menú principal, en la parte de arriba también está el botón de cerrar sesión en el cual se sale del sistema y lo direcciona a login de nuevo.



**Ilustración 38-interfaz menú scanner**

En esta interfaz se muestra el menú de scanner en el cual el primer icono se ingresara la información, en el segundo icono se modificara la información, en el tercer icono se buscara información y por último el cuarto icono direccionara al menú principal, en la parte de arriba también está el botón de cerrar sesión en el cual se sale del sistema y lo direcciona a login de nuevo.

```
<div id="titulo">
  <a href="menucpu.html">
  <a href="">
  <a href=""></a>
  <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
</div>
<div id="menu">
</div>
<form method="POST" action="insertcpu.php" class="contacto" enctype="multipart/form-data">

  <div><label>CODIGO DE INVENTARIO</label><input type="text" id="cod" name="cod"></div>
  <div><label>NUMERO DE SERIE</label><input type="text" id="noser" name="noser"></div>
  <div><label>MODELO</label><input type="text" id="mod" name="mod"></div>
  <div><label>DEPARTAMENTO</label><input type="text" id="dep" name="dep"></div>
  <div><label>RESPONSABLE</label><input type="text" id="res" name="res"></div>
  <div><label>DIRECCION IP</label><input type="text" id="ip" name="ip" ></div>
  <div><label>NOMBRE DEL EQUIPO</label><input type="text" id="nomeq" name="nomeq"></div>
  <div><label>CUENTA DEL EQUIPO</label><input type="text" id="cta" name="cta"></div>
  <div><label>COMENTARIO</label><textarea name="comen" rows="5" cols="30"></div>
  <div><input type="submit" name="submit" id="button" value="Guardar"/>
</div>
</form>
```

**Ilustración 39-código insertacpu**

En la imagen se muestra el código de la interfaz de insertar datos de cpu en el cual tiene 9 textbox en los que se le pide al usuario código de inventario, número de serie, modelo, departamento, responsable, dirección ip, nombre del equipo, cuenta

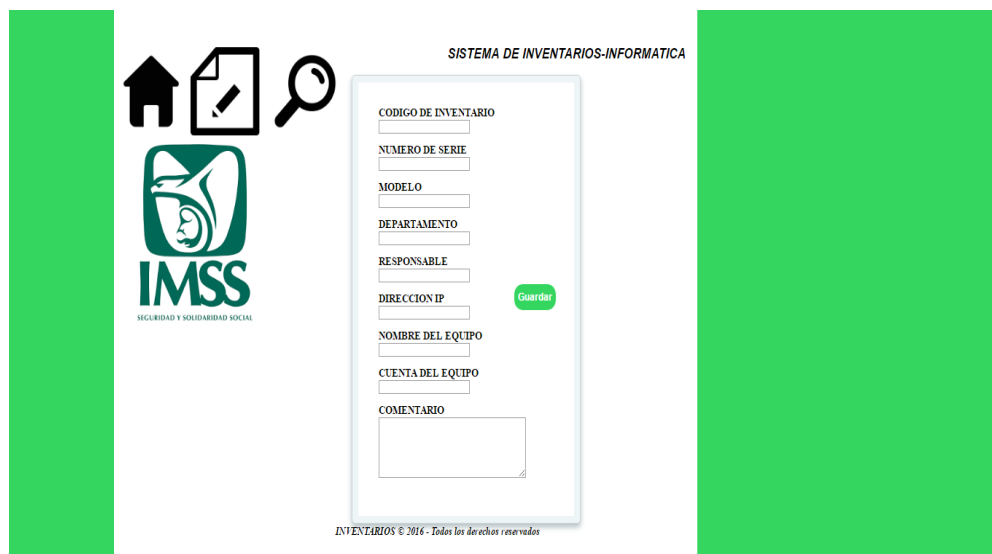


del equipo y comentario, todo esto se envía a insertcpu.php en el que inserta los datos a la base de datos.

```
<body>
<?php
$codigocpu = $_POST["cod"];
$numserie = $_POST["noser"];
$modelo = $_POST["mod"];
$departamento = $_POST["dep"];
$responsable = $_POST["res"];
$direccionip = $_POST["ip"];
$nombeq = $_POST["nomeq"];
$ctaeq = $_POST["cta"];
$comentario = $_POST["comen"];
require('conexion.php');
mysql_select_db('informatica');
$tabla=mysql_query("CALL Insertar_cpu('{$_POST["cod"]}', '{$_POST["noser"]}', '{$_POST["mod"]}',
| '{$_POST["dep"]}', '{$_POST["res"]}', '{$_POST["ip"]}', '{$_POST["nomeq"]}', '{$_POST["cta"]}', '{$_POST["comen"]}', $con);
header('refresh:3; url=insertacpu.html');
?>
</body>
```

**Ilustración 40-código insertcpu**

En la imagen se muestra el código de insertcpu en el cual recibe los datos enviados de la interfaz y los convierte para que se le llame al método creado para ingresar los datos e ingresarlos a la base de datos, se selecciona la base de datos y se llama al método para después se direccionara a la interfaz de insertar cpus.



**Ilustración 41-interfaz de insertar cpu**

En la imagen se muestra la interfaz de insertar cpu en donde el usuario debe de llenar con los datos de cada cpu en los textbox también se puede regresar al menú principal, buscar o editar algún dato erróneo, tiene el logo del IMSS, un botón, los textbox y el nombre del sistema.

```

</div>
<div id="titulo">
  <a href="menumonitores.html">
  <a href="">
  <a href=""></a>
  <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
</div>
<div id="menu">
</div>
<form method="POST" action="insertmon.php" class="contacto" enctype="multipart/form-data">

  <div><label>CODIGO DE INVENTARIO</label><input type="text" id="cod" name="cod"></div>
  <div><label>NUMERO DE SERIE</label><input type="text" id="noser" name="noser"></div>
  <div><label>MODELO</label><input type="text" id="mod" name="mod"></div>
  <div><label>DEPARTAMENTO</label><input type="text" id="dep" name="dep"></div>
  <div><label>RESPONSABLE</label><input type="text" id="res" name="res"></div>
  <div><label>COMENTARIO</label><textarea name="comen" rows="5" cols="30"></textarea></div>
  <div><input type="submit" name="submit" id="botton" value="Guardar"/>
</div>

```

**Ilustración 42-código insertamon**

En la imagen se muestra el código de la interfaz de insertar datos de los monitores en el cual tiene 6 textbox en los que se le pide al usuario código de inventario, número de serie, modelo, departamento, responsable, y comentario, todo esto se envía a insertmon.php en el que inserta los datos a la base de datos.

```

<?php
$codigomon = $_POST["cod"];
$numserie = $_POST["noser"];
$modelo = $_POST["mod"];
$departamento= $_POST["dep"];
$responsable= $_POST["res"];
$comentario= $_POST["comen"];
require('conexion.php');
mysql_select_db('informatica');
$tabla=mysql_query("CALL Insertar_mon('{ $codigomon}', '{ $numserie}', '{ $modelo}',
'{$departamento}', '{ $responsable}', '{ $comentario}'))", $con);
header('refresh:3; url=insertamon.html');
?>
</form>

```

**Ilustración 43-codigo insertmon**

En la imagen se muestra el código de insertmon en el cual recibe los datos enviados de la interfaz y los convierte para que se le llame al método creado para ingresar los datos e ingresarlos a la base de datos, se selecciona la base de datos y se llama al método para después se direccionara a la interfaz de insertar monitores.



**Ilustración 44-interfaz insertar monitores**

En la imagen se muestra la interfaz de insertar monitores en donde el usuario debe de llenar con los datos de cada monitor en los textbox también se puede regresar al menú principal, buscar o editar algún dato erróneo, tiene el logo del IMSS, un botón, los textbox y el nombre del sistema.

```
<div id="titulo">
  <a href="menuscanner.html">
  <a href="">
  <a href=""></a>
  <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
</div>
<div id="menu">
</div>
<form method="POST" action="insertimp.php" class="contacto" enctype="multipart/form-data">

  <div><label>CODIGO DE INVENTARIO</label><input type="text" id="cod" name="cod"></div>
  <div><label>NUMERO DE SERIE</label><input type="text" id="noser" name="noser"></div>
  <div><label>MODELO</label><input type="text" id="mod" name="mod"></div>
  <div><label>DEPARTAMENTO</label><input type="text" id="dep" name="dep"></div>
  <div><label>RESPONSABLE</label><input type="text" id="res" name="res"></div>
  <div><label>COMENTARIO</label><textarea name="comen" rows="5" cols="40"></textarea></div>
  <div><input type="submit" name="submit" id="botton" value="Guardar"/>
</div>
</form>
```

**Ilustración 45-código insertaimp**

En la imagen se muestra el código de la interfaz de insertar datos de las impresoras en el cual tiene 6 textbox en los que se le pide al usuario código de inventario, número de serie, modelo, departamento, responsable, y comentario, todo esto se envía a insertmon.php en el que inserta los datos a la base de datos.

```

<?php
$codigomon = $_POST["cod"];
$numserie = $_POST["noser"];
$modelo = $_POST["mod"];
$departamento= $_POST["dep"];
$responsable= $_POST["res"];
$comentario= $_POST["comen"];
require('conexion.php');
mysql_select_db('informatica');
$tabla=mysql_query("CALL Insertar_imp('{$_POST["cod"]}', '{$_POST["noser"]}',
    '{$_POST["mod"]}', '{$_POST["dep"]}', '{$_POST["res"]}', '{$_POST["comen"]}', $con);
header('refresh:3; url=insertaimp.html');
?>
</form>

```

**Ilustración 46-codigo insertimp**

En la imagen se muestra el código de insertimp en el cual recibe los datos enviados de la interfaz y los convierte para que se le llame al método creado para ingresar los datos e ingresarlos a la base de datos, se selecciona la base de datos y se llama al método para después se direccionara a la interfaz de insertar impresoras.

**Ilustración 47-interfaz insertar impresora**

En la imagen se muestra la interfaz de insertar impresora en donde el usuario debe de llenar con los datos de cada cpu en los textbox también se puede regresar al menú principal, buscar o editar algún dato erróneo, tiene el logo del IMSS, un botón, los textbox y el nombre del sistema.

```

<div id="logo">
    
</div>
<div id="titulo">
    <a href="menuscanner.html">
    <a href="">
    <a href=""></a>
    <h1>SISTEMA DE INVENTARIOS-INFORMATICA</h1>
</div>
<div id="menu">
</div>
<form method="POST" action="insertscan.php" class="contacto" enctype="multipart/form-data">

    <div><label>CODIGO DE INVENTARIO</label><input type="text" id="cod" name="cod"></div>
    <div><label>NUMERO DE SERIE</label><input type="text" id="noser" name="noser"></div>
    <div><label>MODELO</label><input type="text" id="mod" name="mod"></div>
    <div><label>DEPARTAMENTO</label><input type="text" id="dep" name="dep"></div>
    <div><label>RESPONSABLE</label><input type="text" id="res" name="res"></div>
    <div><label>COMENTARIO</label><textarea name="comen" rows="5" cols="30"></textarea></div>
    <div><input type="submit" name="submit" id="botton" value="Guardar"/>
</div>
</form>

```

**Ilustración 48-codigo insertscan**

En la imagen se muestra el código de la interfaz de insertar datos de los scanner en el cual tiene 6 textbox en los que se le pide al usuario código de inventario, número de serie, modelo, departamento, responsable, y comentario, todo esto se envía a insertmon.php en el que inserta los datos a la base de datos.

```

<?php
$codigo = $_POST["cod"];
$numserie = $_POST["noser"];
$modelo = $_POST["mod"];
$departamento= $_POST["dep"];
$responsable= $_POST["res"];
$comentario= $_POST["comen"];
require('conexion.php');
mysql_select_db('informatica');
$tabla=mysql_query("CALL Insertar_escan('{ $codigo}', '{ $numserie}',
    '{ $modelo}', '{ $departamento}', '{ $responsable}', '{ $comentario}'), $con);
header('refresh:3; url=insertscan.html');
?>
</form>

```

**Ilustración 49-código insertscan**

En la imagen se muestra el código de insertscan en el cual recibe los datos enviados de la interfaz y los convierte para que se le llame al método creado para ingresar los datos e ingresarlos a la base de datos, se selecciona la base de datos y se llama al método para después se direccionara a la interfaz de insertar scanner.

**Ilustración 50-interfaz de insertar scanner**

En la imagen se muestra la interfaz de insertar scanner en donde el usuario debe de llenar con los datos de cada cpu en los textbox también se puede regresar al menú principal, buscar o editar algún dato erróneo, tiene el logo del IMSS, un botón, los textbox y el nombre del sistema.

```
<?php
require('conexion1.php');

$query="SELECT cod_inv_mon, no_serie, modelo,
|departamento,responsable,comentario FROM monitores";

$resultado=$mysqli->query($query);

?>
```

**Ilustración 51-codigo modimon.php**

En la imagen el php modimon en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<thead>
  <tr>
    <td><b>codigo de inventario</b></td>
    <td><b>numero de serie</b></td>
    <td><b>modelo</b></td>
    <td><b>departamento</b></td>
    <td><b>responsable</b></td>
    <td><b>comentario</b></td>
    <td></td>

```

**Ilustración 52-codigo modimon**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a modificar.

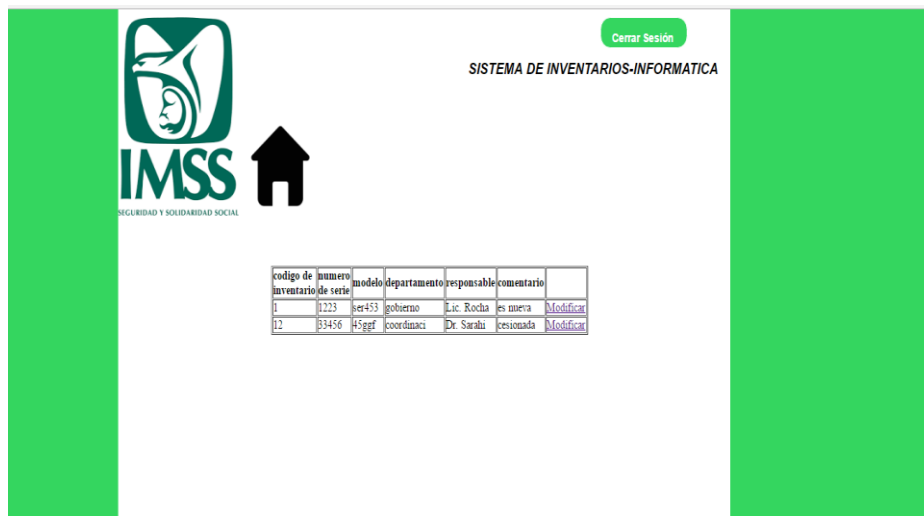
```

<tbody>
  <?php while($row=$resultado->fetch_assoc()){ ?>
    <tr>
      <td><?php echo $row['cod_inv_mon'];?>
      </td>
      <td>
        <?php echo $row['no_serie'];?>
      </td>
      <td><?php echo $row['modelo'];?>
      </td>
      <td>
        <?php echo $row['departamento'];?>
      </td>
      <td><?php echo $row['responsable'];?>
      </td>
      <td>
        <?php echo $row['comentario'];?>
      </td>
      <td>
        <a href="modificarmon.php?cod_inv_mon=<?php echo $row['cod_inv_mon'];?>">Modificar</a>
      </td>
    </tr>
  <?php } ?>

```

**Ilustración 53-codigo modimon 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea modificar y lo direccionara a otro php con una variable que es la de código de inventario .



**Ilustración 54-interfaz de modimon**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser modificada si así se desea, también tiene el botón de home que se regresa al menú anterior.

```
<?php
require('conexion1.php');

$cod_inv_mon=$_GET['cod_inv_mon'];

$query="SELECT no_serie, modelo,departamento,responsable,comentario
|FROM monitores WHERE cod_inv_mon='".$cod_inv_mon'";

$resultado=mysqli->query($query);

$row=$resultado->fetch_assoc();

?>
```

**Ilustración 55-codigo modificarmon.php**

En la imagen se muestra el código de modificarmon.php en el que se requiere la conexión correspondiente para ingresar a la base de datos, también obtiene el código de inventario enviado del php anterior y con ese dato busca específicamente lo que se vaya a modificar y lo guarda en una variable llamada resultado .



```

<tr>
<input type="hidden" name="cod_inv_mon" value="<?php echo $cod_inv_mon; ?>">
<td width="20"><b>numero de serie</b></td>
<td width="30"><input type="text" name="no_serie" size="25" value="<?php echo $row['no_serie']; ?>" /></td>
</tr>
<tr>
<td><b>modelo</b></td>
<td><input type="text" name="modelo" size="25" value="<?php echo $row['modelo']; ?>" /></td>
</tr>
<tr>
<td><b>departamento</b></td>
<td><input type="text" name="departamento" size="25" value="<?php echo $row['departamento']; ?>" /></td>
</tr>
<tr>
<td><b>responsable</b></td>
<td><input type="text" name="responsable" size="25" value="<?php echo $row['responsable']; ?>" /></td>
</tr>
<tr>
<td><b>comentario</b></td>
<td><input type="text" name="comentario" size="25" value="<?php echo $row['comentario']; ?>" /></td>
</tr>
<tr>
<td colspan="2"><input type="submit" name="Guardar" value="Guardar" /></td>
</tr>
</table>
Form>

```

**Ilustración 56-código modificarmon 2**

En la imagen se muestra un método en el que se rellenan los combobox respectivamente para poder editar su contenido y así poder enviar los datos correctos que fueron modificados a otro php para hacer la modificación en la base de datos con otro método.

SISTEMA DE INVENTARIOS-INFORMATICA

IMSS  
SEGURIDAD Y SOLIDARIDAD SOCIAL

numero de serie: 1223  
 modelo: ser453  
 departamento: gobierno  
 responsable: Lic. Rocha  
 comentario: es nueva

Guardar

**Ilustración 57-interfaz modificarmon**

En la imagen se muestra la interfaz en la cual se auto rellenan los combobox de acuerdo a los datos que estén en la base de datos que pueden ser modificados y el botón de guardar donde hace el proceso de modificar y guardar los datos de nuevo en la base de datos.

```

<?php
    require('conexion1.php');

    $codigomon = $_POST["cod_inv_mon"];
    $numserie = $_POST["no_serie"];
    $modelo = $_POST["modelo"];
    $departamento = $_POST["departamento"];
    $responsable = $_POST["responsable"];
    $comentario = $_POST["comentario"];

    $query="UPDATE monitores SET no_serie='$numserie', modelo='$modelo', departamento='$departamento',
    responsable='$responsable', comentario='$comentario' WHERE cod_inv_mon='$codigomon'";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 58-codigo modi\_mon.php**

En la imagen se muestra el método en el que se requiere la conexión a la base de datos también la conversión de los datos enviados del php anterior para poder modificarlos, se muestra el método para reingresar los datos modificados a la base de datos y a la tabla correspondiente.

```

<?php
    require('conexion1.php');

    $query="SELECT cod_inv_cpu, noserie, modelo,departamento,responsable,
    direccionip,nombre_equipo,cuenta_equipo,comentario FROM cpus";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 59-codigo modicpu.php**

En la imagen el php modimon en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<table border=1 width="80%">
    <thead>
        <tr>
            <td><b>codigo de inventario</b></td>
            <td><b>numero de serie</b></td>
            <td><b>modelo</b></td>
            <td><b>departamento</b></td>
            <td><b>responsable</b></td>
            <td><b>direccion ip</b></td>

            <td><b>comentario</b></td>
            <td></td>
        </tr>
    </thead>
    <tbody>

```

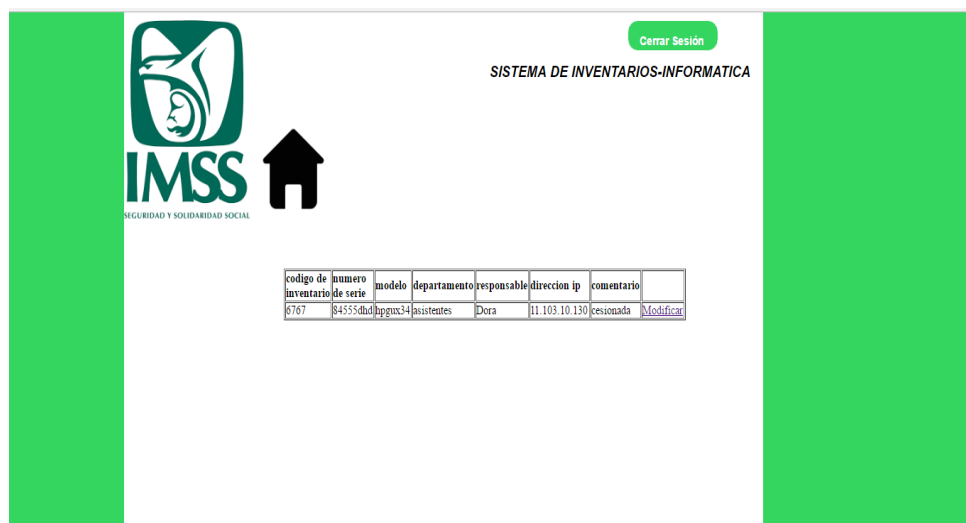
**Ilustración 60-codigo modicpu 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a modificar.

```
<?php while($row=$resultado->fetch_assoc()){ ?>
<tr>
<td><?php echo $row['cod_inv_cpu'];?>
</td>
<td>
<?php echo $row['noserie'];?>
</td>
<td><?php echo $row['modelo'];?>
</td>
<td>
<?php echo $row['departamento'];?>
</td>
<td><?php echo $row['responsable'];?>
</td>
<td><?php echo $row['direccionip'];?>
</td>
<td>
<?php echo $row['comentario'];?>
</td>
<td>
<a href="modificarcpu.php?cod_inv_cpu=<?php echo $row['cod_inv_cpu'];?>">Modificar</a>
</td>
</tr>
<?php } ?>
```

**Ilustración 61-codigo modicpu 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea modificar y lo direccionara a otro php con una variable que es la de código de inventario.



**Ilustración 62-interfaz modicpu**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser modificada si así se desea,

también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

```
<?php
    require('conexion1.php');

    $cod_inv_cpu=$_GET['cod_inv_cpu'];

    $query="SELECT noserie, modelo,departamento,responsable,direccionip,
    nombre_equipo,cuenta_equipo,comentario FROM cpus WHERE cod_inv_cpu='".$cod_inv_cpu."'";

    $resultado=mysqli->query($query);

    $row=$resultado->fetch_assoc();

?>
```

**Ilustración 63-codigo modificarcpu.php**

En la imagen se muestra el código de modificarmon.php en el que se requiere la conexión correspondiente para ingresar a la base de datos, también obtiene el código de inventario enviado del php anterior y con ese dato busca específicamente lo que se vaya a modificar y lo guarda en una variable llamada resultado .

```
<form name="modificar_moniotres" method="POST" action="mod_cpu.php">
    <table width="50%">
        <tr>
            <input type="hidden" name="cod_inv_cpu" value="<?php echo $cod_inv_cpu; ?>" />
            <td width="20%"><b>numero de serie</b></td>
            <td width="30%"><input type="text" name="noserie" size="25" value="<?php echo $row['noserie']; ?>" /></td>
        </tr>
        <tr>
            <td><b>modelo</b></td>
            <td><input type="text" name="modelo" size="25" value="<?php echo $row['modelo']; ?>" /></td>
        </tr>
        <tr>
            <td><b>departamento</b></td>
            <td><input type="text" name="departamento" size="25" value="<?php echo $row['departamento']; ?>" /></td>
        </tr>
        <tr>
            <td><b>responsable</b></td>
            <td><input type="text" name="responsable" size="25" value="<?php echo $row['responsable']; ?>" /></td>
        </tr>
        <tr>
            <td><b>direccion ip</b></td>
            <td><input type="text" name="direccionip" size="25" value="<?php echo $row['direccionip']; ?>" /></td>
        </tr>
        <tr>
            <td><b>comentario</b></td>
            <td><input type="text" name="comentario" size="25" value="<?php echo $row['comentario']; ?>" /></td>
        </tr>
        <tr>
            <td colspan="2"><center><input type="submit" name="Guardar" value="Guardar" /></center></td>
        </tr>
    </table>
</form>
```

**Ilustración 64-codigo modificarcpu 2**

En la imagen se muestra un método en el que se rellenan los combobox respectivamente para poder editar su contenido y así poder enviar los datos correctos que fueron modificados a otro php para hacer la modificación en la base de datos con otro método.

SISTEMA DE INVENTARIOS-INFORMATICA



numero de	84555dhd
serie	
modelo	hpgu34
departamento	asistentes
responsable	Dora
direccion ip	11.103.10.130
comentario	cesionada

**Ilustración 65-interfaz modificarcpu**

En la imagen se muestra la interfaz en la cual se auto rellenan los combobox de acuerdo a los datos que estén en la base de datos que pueden ser modificados y el botón de guardar donde hace el proceso de modificar y guardar los datos de nuevo en la base de datos.

```
<?php
require('conexion1.php');

$codigocpu = $_POST["cod_inv_cpu"];
$numserie = $_POST["noserie"];
$modelo = $_POST["modelo"];
$departamento= $_POST["departamento"];
$responsable= $_POST["responsable"];
$direccionip = $_POST["direccionip"];
$comentario= $_POST["comentario"];

$query="UPDATE cpus SET noserie='$numserie', modelo='$modelo', departamento='$departamento',
responsable='$responsable', direccionip='$direccionip',comentario='$comentario' WHERE cod_inv_cpu='$codigocpu'";
$resultado=mysqli->query($query);
```

**Ilustración 66-codigo modi\_cpu**

En la imagen se muestra el método en el que se requiere la conexión a la base de datos también la conversión de los datos enviados del php anterior para poder modificarlos, se muestra el método para reingresar los datos modificados a la base de datos y ala tabla correspondiente.

```

<?php
    require('conexion1.php');

    $query="SELECT cod_inv_imp, no_serie, modelo,departamento,
    responsable,comentario FROM impresora";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 67-codigo modiimp.php**

En la imagen el php modiimp en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<thead>
<tr>
    <td><b>codigo de inventario</b></td>
    <td><b>numero de serie</b></td>
    <td><b>modelo</b></td>
    <td><b>departamento</b></td>
    <td><b>responsable</b></td>
    <td><b>comentario</b></td>
    <td></td>

<tbody>

```

**Ilustración 68-codigo modiimp 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a modificar.

```

<tbody>
<?php while($row=$resultado->fetch_assoc()){ ?>
    <tr>
        <td><?php echo $row['cod_inv_imp'];?>
        </td>
        <td>
            <?php echo $row['no_serie'];?>
        </td>
        <td><?php echo $row['modelo'];?>
        </td>
        <td>
            <?php echo $row['departamento'];?>
        </td>
        <td><?php echo $row['responsable'];?>
        </td>
        <td>
            <?php echo $row['comentario'];?>
        </td>
        <td>
            <a href="modificarimp.php?cod_inv_imp=<?php echo $row['cod_inv_imp'];?>">Modificar</a>
        </td>
    </tr>
<?php } ?>

```

**Ilustración 69-codigo modiimp 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón

en el que se podrá seleccionar si se desea modificar y lo direccionara a otro php con una variable que es la de código de inventario.



**Ilustración 70-interfaz modiimp**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser modificada si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

```
<?php
require('conexion1.php');

$cod_inv_imp=$_GET['cod_inv_imp'];

$query="SELECT no_serie, modelo,departamento,responsable,comentario
FROM impresora WHERE cod_inv_imp='$cod_inv_imp'";

$resultado=mysqli->query($query);

$row=$resultado->fetch_assoc();

?>
```

**Ilustración 71-código modificarimp.php**

En la imagen se muestra el código de modificarimp.php en el que se requiere la conexión correspondiente para ingresar a la base de datos, también obtiene el código de inventario enviado del php anterior y con ese dato busca específicamente lo que se vaya a modificar y lo guarda en una variable llamada resultado .

```

<table width="90%">
  <tr>
    <input type="hidden" name="cod_inv_imp" value="<?php echo $cod_inv_imp; ?>" />
    <td width="20%"><b>numero de serie</b></td>
    <td width="30%"><input type="text" name="no_serie" size="25" value="<?php echo $row['no_serie']; ?>" /></td>
  </tr>
  <tr>
    <td><b>modelo</b></td>
    <td><input type="text" name="modelo" size="25" value="<?php echo $row['modelo']; ?>" /></td>
  </tr>
  <tr>
    <td><b>departamento</b></td>
    <td><input type="text" name="departamento" size="25" value="<?php echo $row['departamento']; ?>" /></td>
  </tr>
  <tr>
    <td><b>responsable</b></td>
    <td><input type="text" name="responsable" size="25" value="<?php echo $row['responsable']; ?>" /></td>
  </tr>
  <tr>
    <td><b>comentario</b></td>
    <td><input type="text" name="comentario" size="25" value="<?php echo $row['comentario']; ?>" /></td>
  </tr>
  <tr>
    <td colspan="2"><center><input type="submit" name="Guardar" value="Guardar" /></center></td>
  </tr>
</table>

```

**Ilustración 72-codigo modificarimpr2**

En la imagen se muestra un método en el que se rellenan los combobox respectivamente para poder editar su contenido y así poder enviar los datos correctos que fueron modificados a otro php para hacer la modificación en la base de datos con otro método.

SISTEMA DE INVENTARIOS-INFORMATICA

IMSS  
SEGURIDAD Y SOLIDARIDAD SOCIAL

numero de serie: e34334344344334  
 modelo: multifun  
 departamento: gobierno  
 responsable: Bere  
 comentario: 3433443

Guardar

**Ilustración 73-interfaz modificarimp**

En la imagen se muestra la interfaz en la cual se auto rellenan los combobox de acuerdo a los datos que estén en la base de datos que pueden ser modificados y el botón de guardar donde hace el proceso de modificar y guardar los datos de nuevo en la base de datos.



```

<?php
    require('conexion1.php');

    $codigo = $_POST["cod_inv_imp"];
    $numserie = $_POST["no_serie"];
    $modelo = $_POST["modelo"];
    $departamento = $_POST["departamento"];
    $responsable = $_POST["responsable"];
    $comentario = $_POST["comentario"];

    $query="UPDATE impresora SET no_serie='$numserie', modelo='$modelo', departamento='$departamento',
    responsable='$responsable', comentario='$comentario' WHERE cod_inv_imp='$codigo'";

    $resultado=$mysqli->query($query);
?>

```

**Ilustración 74-codigo modi\_impr**

En la imagen se muestra el método en el que se requiere la conexión a la base de datos también la conversión de los datos enviados del php anterior para poder modificarlos, se muestra el método para reingresar los datos modificados a la base de datos y a la tabla correspondiente.

```

<?php
    require('conexion1.php');

    $query="SELECT cod_inv_esc, no_serie, modelo,departamento,responsable,comentario FROM escaner";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 75-codigo modiscan.php**

En la imagen el php modimon en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<thead>
    <tr>
        <td><b>codigo de inventario</b></td>
        <td><b>numero de serie</b></td>
        <td><b>modelo</b></td>
        <td><b>departamento</b></td>
        <td><b>responsable</b></td>
        <td><b>comentario</b></td>
        <td></td>
    </tr>
</thead>
<tbody>
<?php while($row=$resultado->fetch_assoc()

```

**Ilustración 76-codigo modiscan2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a modificar.

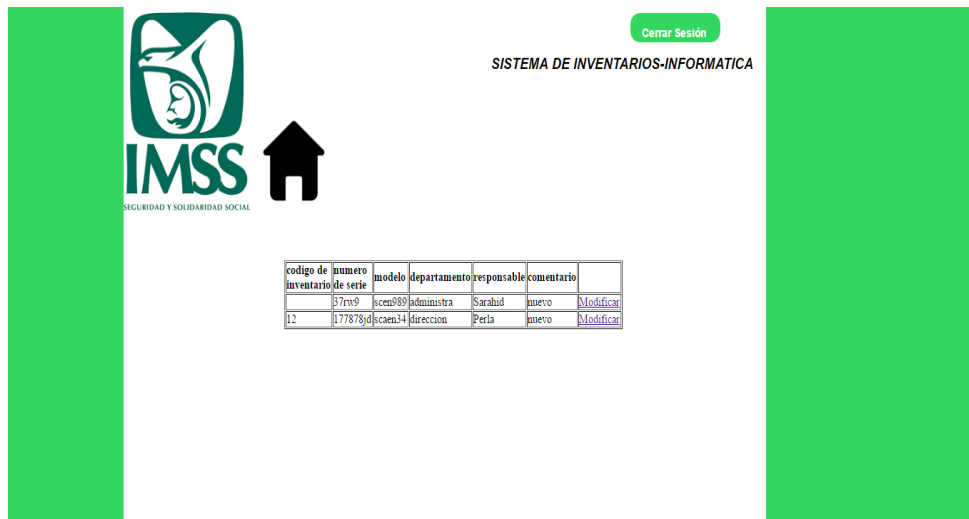
```

<tbody>
<?php while($row=$resultado->fetch_assoc()){ ?>
  <tr>
    <td><?php echo $row['cod_inv_esc'];?>
    </td>
    <td>
      <?php echo $row['no_serie'];?>
    </td>
    <td><?php echo $row['modelo'];?>
    </td>
    <td>
      <?php echo $row['departamento'];?>
    </td>
    <td><?php echo $row['responsable'];?>
    </td>
    <td>
      <?php echo $row['comentario'];?>
    </td>
    <td>
      <a href="modificaresc.php?cod_inv_esc=<?php echo $row['cod_inv_esc'];?>">Modificar</a>
    </td>
  </tr>
<?php } ?>

```

**Ilustración 77-codigo modiscan3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea modificar y lo direccionara a otro php con una variable que es la de código de inventario.



**Ilustración 78-interfaz modiscan**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser modificada si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

```

<?php

require('conexion1.php');

$cod_inv_esc=$_GET['cod_inv_esc'];

$query="SELECT no_serie, modelo,departamento,responsable,comentario
FROM escaner WHERE cod_inv_esc='$cod_inv_esc'";

$resultado=mysqli->query($query);

$row=$resultado->fetch_assoc();

?>

```

**Ilustración 79-codigo modificarscan**

En la imagen se muestra el código de modificarmon.php en el que se requiere la conexión correspondiente para ingresar a la base de datos, también obtiene el código de inventario enviado del php anterior y con ese dato busca específicamente lo que se vaya a modificar y lo guarda en una variable llamada resultado .

```


<table width="50%">
<tr>
<td>
<input type="hidden" name="cod_inv_esc" value="<?php echo $cod_inv_esc; ?>">
<td width="20%"><b>numero de serie</b></td>
<td width="30%"><input type="text" name="no_serie" size="25" value="<?php echo $row['no_serie']; ?>" /></td>
</tr>
<tr>
<td><b>modelo</b></td>
<td><input type="text" name="modelo" size="25" value="<?php echo $row['modelo']; ?>" /></td>
</tr>
<tr>
<td><b>departamento</b></td>
<td><input type="text" name="departamento" size="25" value="<?php echo $row['departamento']; ?>" /></td>
</tr>
<tr>
<td><b>responsable</b></td>
<td><input type="text" name="responsable" size="25" value="<?php echo $row['responsable']; ?>" /></td>
</tr>
<tr>
<td><b>comentario</b></td>
<td><input type="text" name="comentario" size="25" value="<?php echo $row['comentario']; ?>" /></td>
</tr>
<tr>
<td colspan="2"><center><input type="submit" name="Guardar" value="Guardar" /></center></td>
</tr>
</table>

```

**Ilustración 80-codigo modificarscan2**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea modificar y lo direccionara a otro php con una variable que es la de código de inventario.

SISTEMA DE INVENTARIOS-INFORMATICA



numero de	07m9
serie	
modelo	scan989
departamento	administra
responsable	Sarahid
comentario	nuevo

**Ilustración 81-interfaz modificarscan**

En la imagen se muestra la interfaz en la cual se auto rellenan los combobox de acuerdo a los datos que estén en la base de datos que pueden ser modificados y el botón de guardar donde hace el proceso de modificar y guardar los datos de nuevo en la base de datos.

```
<?php
require('conexion1.php');

$codigoesc = $_POST["cod_inv_esc"];
$numserie = $_POST["no_serie"];
$modelo = $_POST["modelo"];
$departamento= $_POST["departamento"];
$responsable= $_POST["responsable"];
$comentario= $_POST["comentario"];

$query="UPDATE escaner SET no_serie='$numserie', modelo='$modelo', departamento='$departamento',
responsable='$responsable', comentario='$comentario' WHERE cod_inv_esc='$codigoesc'";

$resultado=mysqli->query($query);
?>
```

**Ilustración 82-codigo modi\_scan.php**

En la imagen se muestra el método en el que se requiere la conexión a la base de datos también la conversión de los datos enviados del php anterior para poder modificarlos, se muestra el método para reingresar los datos modificados a la base de datos y ala tabla correspondiente.

```

<?php
    require('conexion1.php');

    $query="SELECT cod_inv_mon, no_serie, modelo,
    |departamento,responsable,comentario FROM monitores";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 83-codigo buscarmonitor**

En la imagen el php buscarmonitor en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<thead>
    <tr>
        <td><b>codigo de inventario</b></td>
        <td><b>numero de serie</b></td>
        <td><b>modelo</b></td>
        <td><b>departamento</b></td>
        <td><b>responsable</b></td>
        <td><b>comentario</b></td>
        <td></td>
    </tr>

```

**Ilustración 84-codigo buscarmonitor 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a imprimir.

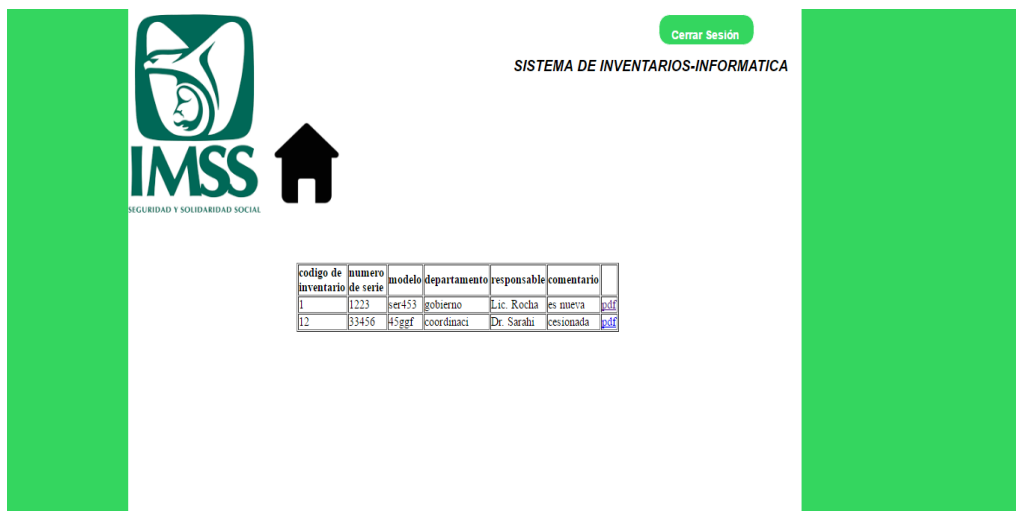
```

<tbody>
    <?php while($row=$resultado->fetch_assoc()){ ?>
    <tr>
        <td><?php echo $row['cod_inv_mon'];?>
        </td>
        <td>
            <?php echo $row['no_serie'];?>
        </td>
        <td><?php echo $row['modelo'];?>
        </td>
        <td>
            <?php echo $row['departamento'];?>
        </td>
        <td><?php echo $row['responsable'];?>
        </td>
        <td>
            <?php echo $row['comentario'];?>
        </td>
        <td>
            <a href="buscarpdfmonitor.php?cod_inv_mon=<?php echo $row['cod_inv_mon'];?>">pdf</a>
        </td>
    </tr>

```

**Ilustración 85-codigo buscarmonitor 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea imprimir y lo direccionara a crear un pdf que se podrá imprimir.



**Ilustración 86-interfaz buscarmonitor**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser creada pdf e imprimir si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

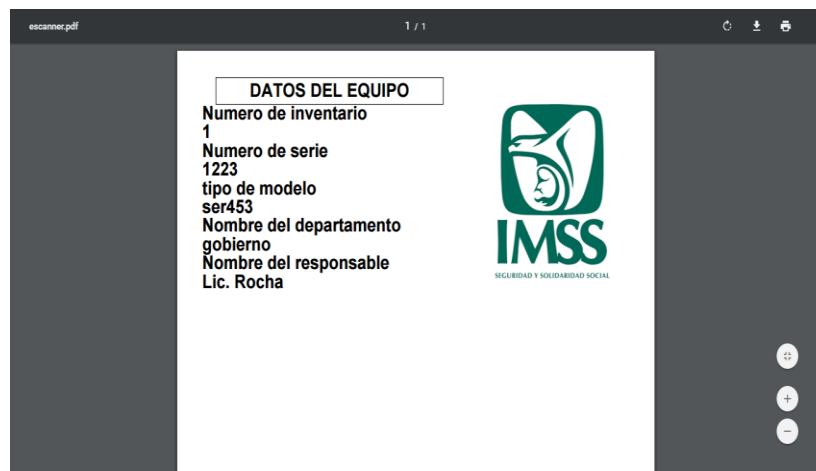
```
<?php
include ('fpdf/fpdf.php');
$pdf = new FPDF();
$pdf->AddPage();

$pdf->SetFont('Helvetica', 'B', 20);
$pdf->Cell(7,85,$pdf->Image('img/logo.png',140,20,50),0,0,'C');
$pdf->Cell(100,10,"DATOS DEL EQUIPO",1,0,'C');
$pdf->Ln();
$pdf->Write (7,"Numero de inventario");
$pdf->Ln();
$pdf->Write (7,$_POST['cod_inv_mon']);
$pdf->Ln();
$pdf->Write (7,"Numero de serie");
$pdf->Ln();
$pdf->Write(7,$_POST['no_serie']);
$pdf->Ln();
$pdf->Write (7,"tipo de modelo");
$pdf->Ln();
$pdf->Write (7,$_POST['modelo']);
$pdf->Ln();
$pdf->Write (7,"Nombre del departamento");
$pdf->Ln();
$pdf->Write (7,$_POST['departamento']);
$pdf->Ln();
$pdf->Write (7,"Nombre del responsable");
$pdf->Ln();
$pdf->Write (7,$_POST['responsable']);
$pdf->Ln();

$pdf->Output("escanner.pdf","F");
echo "<script language='javascript'>window.open('escanner.pdf','_self','');</script>";
exit;
?>
```

**Ilustración 87-codigo generadormonitor**

En la imagen se muestra el código de generadormonitor el cual tiene la creación del pdf de la tabla se desea imprimir y a la vez se puede guardar, el pdf contiene los datos de la tabla y el logo del IMSS.



**Ilustración 88-pdf monitor**

En la imagen se muestra el pdf creado con los datos ingresados en la tabla, se puede imprimir o se puede guardar dependiendo de lo que se quiera hacer con el pdf.

```
<?php
require('conexion1.php');

$query="SELECT cod_inv_cpu, noserie, modelo,departamento,responsable,
|direccionip,nombre_equipo,cuenta_equipo,comentario FROM cpus";

$resultado=$mysqli->query($query);

?>
```

**Ilustración 89-codigo buscarcpu**

En la imagen el php buscarcpu en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```
<table border=1 width="80%">
<thead>
<tr>
<td><b>codigo de inventario</b></td>
<td><b>numero de serie</b></td>
<td><b>modelo</b></td>
<td><b>departamento</b></td>
<td><b>responsable</b></td>
<td><b>direccion ip</b></td>
<td><b>comentario</b></td>
<td></td>
</tr>
<tbody>
```

**Ilustración 90-codigo buscarcpu 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a imprimir.

```

<tbody>
<?php while($row=$resultado->fetch_assoc()){ ?>
<tr>
<td><?php echo $row['cod_inv_mon'];?>
</td>
<td>
<?php echo $row['no_serie'];?>
</td>
<td><?php echo $row['modelo'];?>
</td>
<td>
<?php echo $row['departamento'];?>
</td>
<td><?php echo $row['responsable'];?>
</td>
<td>
<?php echo $row['comentario'];?>
</td>
<td>
<a href="buscarpdfmonitor.php?cod_inv_mon=<?php echo $row['cod_inv_mon'];?>">pdf</a>
</td>
</tr>

```

**Ilustración 91-codigo buscarcpu 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea imprimir y lo direccionara a crear un pdf que se podrá imprimir.



**Ilustración 92-interfaz buscarcpu**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser creada pdf e imprimir si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.



```

<?php
include ('fpdf/fpdf.php');

$pdf = new FPDF();
$pdf->AddPage();

$pdf->SetFont('Helvetica', 'B', 20);
$pdf->Cell(7,85,$pdf->Image('img/logo.png',140,20,50),0,0,'C');
$pdf->Cell(100,10,"DATOS DEL EQUIPO",1,0,'C');
$pdf->Ln();
$pdf->Write (7,"Numero de inventario");
$pdf->Ln();
$pdf->Write (7,$_POST['cod_inv_cpu']);
$pdf->Ln();
$pdf->Write (7,"Numero de serie");
$pdf->Ln();
$pdf->Write(7,$_POST['noserie']);
$pdf->Ln();
$pdf->Write (7,"tipo de modelo");
$pdf->Ln();
$pdf->Write (7,$_POST['modelo']);
$pdf->Ln();
$pdf->Write (7,"Direccion ip");
$pdf->Ln();
$pdf->Write (7,$_POST['direccionip']);
$pdf->Ln();
$pdf->Write (7,"Nombre del departamento");
$pdf->Ln();
$pdf->Write (7,$_POST['departamento']);
$pdf->Ln();

$pdf->Write (7,"Nombre del responsable");
$pdf->Ln();
$pdf->Write (7,$_POST['responsable']);
$pdf->Ln();

$pdf->Output("escanner.pdf",'F');
echo "<script language='javascript'>window.open('escanner.pdf','_self','');</script>";
exit;
?>

```

**Ilustración 93-codigo generadorcpu**

En la imagen se muestra el codigo de generadorcpu el cual tiene la creación del pdf de la tabla se desea imprimir y a la vez se puede guardar, el pdf contiene los datos de la tabla y el logo del IMSS.



**Ilustración 94-pdf cpu**

En la imagen se muestra el pdf creado con los datos ingresados en la tabla, se puede imprimir o se puede guardar dependiendo de lo que se quiera hacer con el pdf.

```

<?php
    require('conexion1.php');

    $query="SELECT cod_inv_imp, no_serie, modelo,departamento,
    responsable,comentario FROM impresora";

    $resultado=$mysqli->query($query);

?>

```

**Ilustración 95-codigo buscarimp**

En la imagen el php buscarimp en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```

<thead>
    <tr>
        <td><b>codigo de inventario</b></td>
        <td><b>numero de serie</b></td>
        <td><b>modelo</b></td>
        <td><b>departamento</b></td>
        <td><b>responsable</b></td>
        <td><b>comentario</b></td>
        <td></td>
    </tr>
</thead>

```

**Ilustración 96-codigo buscarimp 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a imprimir.

```

<tbody>
    <?php while($row=$resultado->fetch_assoc()){ ?>
        <tr>
            <td><?php echo $row['cod_inv_imp'];?>
            </td>
            <td>
                <?php echo $row['no_serie'];?>
            </td>
            <td><?php echo $row['modelo'];?>
            </td>
            <td>
                <?php echo $row['departamento'];?>
            </td>
            <td><?php echo $row['responsable'];?>
            </td>
            <td>
                <?php echo $row['comentario'];?>
            </td>
            <td>
                <a href="buscarpdfimpre.php?cod_inv_imp=<?php echo $row['cod_inv_imp'];?>">pdf</a>
            </td>
        </tr>
    <?php } ?>
</tbody>

```

**Ilustración 97-codigo buscarimp 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea imprimir y lo direccionara a crear un pdf que se podrá imprimir.



**Ilustración 98-interfaz buscarimp**

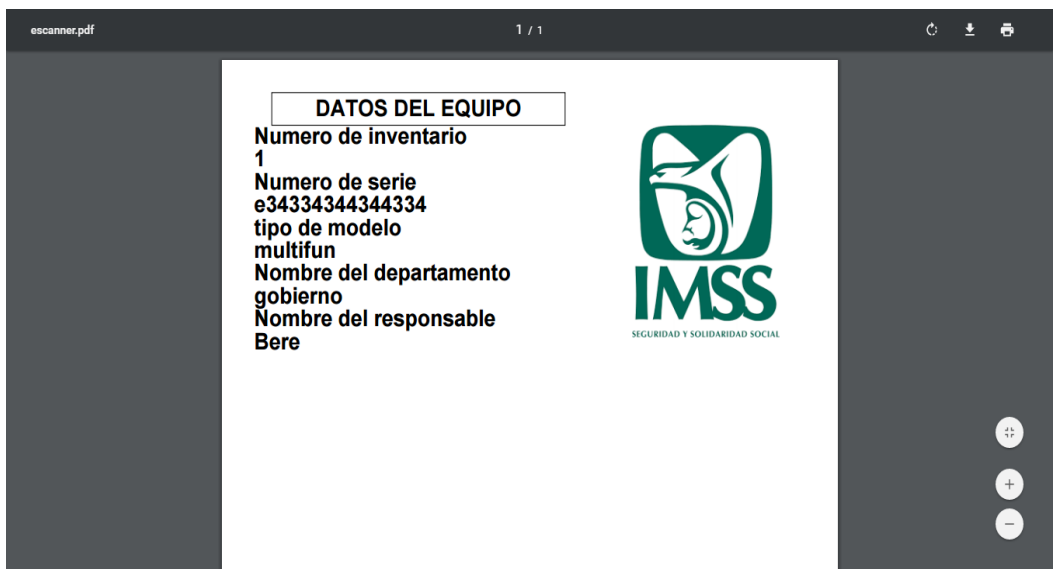
En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser creada pdf e imprimir si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

```
<?php
include ('fpdf/fpdf.php');
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Helvetica', 'B', 20);
$pdf->Cell(7,85,$pdf->Image('img/logo.png',140,20,50),0,0,'C');
$pdf->Cell(100,10,"DATOS DEL EQUIPO",1,0,'C');
$pdf->Ln();
$pdf->Write (7,"Numero de inventario");
$pdf->Ln();
$pdf->Write (7,$_POST['cod_inv_esc']);
$pdf->Ln();
$pdf->Write (7,"Numero de serie");
$pdf->Ln();
$pdf->Write(7,$_POST['no_serie']);
$pdf->Ln();
$pdf->Write (7,"tipo de modelo");
$pdf->Ln();
$pdf->Write (7,$_POST['modelo']);
$pdf->Ln();
$pdf->Write (7,"Nombre del departamento");
$pdf->Ln();
$pdf->Write (7,$_POST['departamento']);
$pdf->Ln();
$pdf->Write (7,"Nombre del responsable");
$pdf->Ln();
$pdf->Write (7,$_POST['responsable']);
$pdf->Ln();

$pdf->Output("escanner.pdf","F");
echo "<script language='javascript'>window.open('escanner.pdf','_self','');</script>"
exit;
?>
```

**Ilustración 99-codigo generadorimp**

En la imagen se muestra el código de generadorimp el cual tiene la creación del pdf de la tabla se desea imprimir y a la vez se puede guardar, el pdf contiene los datos de la tabla y el logo del IMSS.



**Ilustración 100-pdf impresora**

En la imagen se muestra el pdf creado con los datos ingresados en la tabla, se puede imprimir o se puede guardar dependiendo de lo que se quiera hacer con el pdf.

```
<?php
require('conexion1.php');

$query="SELECT cod_inv_esc, no_serie, modelo,departamento,responsable,comentario FROM escaner";

$resultado=mysqli->query($query);

?>
```

**Ilustración 101-codigo buscarscan**

En la imagen el php buscarscan en el cual se muestra que se pide la conexión que se requiere como también la selección de la tabla con sus columnas correspondientes y así mismo se almacena en una variable llamada resultado.

```
<thead>
<tr>
<td><b>codigo de inventario</b></td>
<td><b>numero de serie</b></td>
<td><b>modelo</b></td>
<td><b>departamento</b></td>
<td><b>responsable</b></td>
<td><b>comentario</b></td>
<td></td>
</tr>
</thead>
<tbody>
<?php while($row=mysqli->fetch_assoc())
```

**Ilustración 102-codigo buscarscan 2**

En la imagen se muestra la creación de la tabla donde se mostrara los datos obtenidos en la imagen anteriormente, se mostrara toda la tabla para poder seleccionar el que se vaya a imprimir.

```

<tbody>
<?php while($row=$resultado->fetch_assoc()){ ?>
<tr>
<td><?php echo $row['cod_inv_esc'];?>
</td>
<td>
<?php echo $row['no_serie'];?>
</td>
<td><?php echo $row['modelo'];?>
</td>
<td><?php echo $row['departamento'];?>
</td>
<td><?php echo $row['responsable'];?>
</td>
<td>
<?php echo $row['comentario'];?>
</td>
<td>
<a href="buscарpdfscan.php?cod_inv_esc=<?php echo $row['cod_inv_esc'];?>">pdf</a>
</td>
</tr>
<?php } ?>

```

**Ilustración 103-codigo buscarscan 3**

En la imagen se muestra un método en el cual se introduce cada dato en la tabla ya creada anteriormente así como también a lado de cada fila de la tabla habrá un botón en el que se podrá seleccionar si se desea imprimir y lo direccionara a crear un pdf que se podrá imprimir.

SISTEMA DE INVENTARIOS-INFORMATICA

codigo de inventario	numero de serie	modelo	departamento	responsable	comentario	
37rv9	scen989	administra	Sarahid	nuevo		<a href="#">pdf</a>
12	177878d	scen34	direccion	Perla	nuevo	<a href="#">pdf</a>

**Ilustración 104-interfaz buscarscan**

En la imagen se muestra la interfaz todos los datos que se tiene en la tabla de la base de datos, se puede seleccionar cada fila para ser creada pdf e imprimir si así se desea, también tiene el botón de home que se regresa al menú anterior también tiene el nombre del proyecto y el logo del IMSS.

```

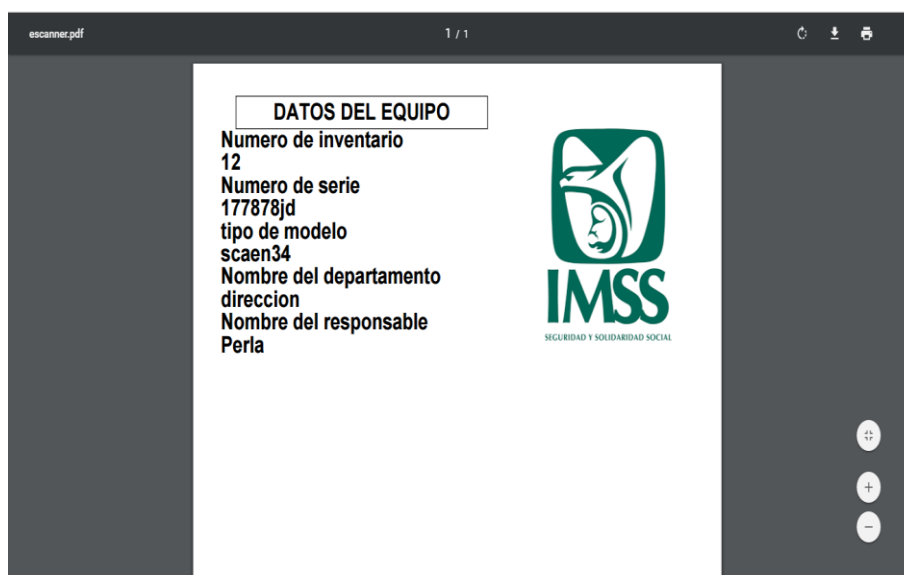
<?php
include ('fpdf/fpdf.php');
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Helvetica', 'B', 20);
$pdf->Cell(7,85,$pdf->Image('img/logo.png',140,20,50),0,0,'C');
$pdf->Cell(100,10,"DATOS DEL EQUIPO",1,0,'C');
    $pdf->Ln();
    $pdf->Write (7,"Numero de inventario");
    $pdf->Ln();
    $pdf->Write (7,$_POST['cod_inv_esc']);
    $pdf->Ln();
    $pdf->Write (7,"Numero de serie");
    $pdf->Ln();
    $pdf->Write(7,$_POST['no_serie']);
    $pdf->Ln();
    $pdf->Write (7,"tipo de modelo");
    $pdf->Ln();
    $pdf->Write (7,$_POST['modelo']);
    $pdf->Ln();
    $pdf->Write (7,"Nombre del departamento");
    $pdf->Ln();
    $pdf->Write (7,$_POST['departamento']);
    $pdf->Ln();
    $pdf->Write (7,"Nombre del responsable");
    $pdf->Ln();
    $pdf->Write (7,$_POST['responsable']);
    $pdf->Ln();

$pdf->Output("escanner.pdf",'F');
echo "<script language='javascript'>window.open('escanner.pdf','_self','');</script>";
exit;
?>

```

**Ilustración 105-codigo generadorscan**

En la imagen se muestra el codigo de generadorscan el cual tiene la creación del pdf de la tabla se desea imprimir y a la vez se puede guardar, el pdf contiene los datos de la tabla y el logo del IMSS.



**Ilustración 106-pdf scanner**

## **4.- CONCLUSIONES Y RECOMENDACIONES**

#### ***4.1.-CONCLUSIONES***

En este proyecto tuve la oportunidad de obtener demasiados conocimientos como también reforzar algunos que ya se tenían, ya que mi proyecto con llevo algunas cosas nuevas que no conocía y tuve que investigar como también algunas cosas que fueron adquiridas en el caminar de la carrera. Se tuvo la satisfacción de cumplir con los objetivos que se propusieron al empezar todo este proyecto en lo personal pensé que no los lograría pero con ayuda de mis asesores pude sacar esto adelante, también es muy importante recalcar que se obtuvo un excelente resultado positivo ya que hubo demasiados comentarios positivos en la empresa ya que se agilizo un proceso que se hacía en Excel anteriormente y se hacia otro proceso aparte y con este proyecto se pudo juntar esos procesos. También se llegó a un acuerdo de seguir con el proyecto en un futuro para que otras áreas también lo pueda utilizar y adaptar a sus necesidades.

#### ***4.2.-RECOMENDACIONES***

Una de las recomendaciones que se pueda dar de este proyecto es de que como lo mencione anteriormente se podrá utilizar y se dará seguimiento e ampliar el campo de este sistema que se le puedan agregar más apartados o procesos al mismo, es muy modificable y adaptable fácilmente mediante el código que no es nada de otro mundo como también este sistema se podría aplicar en cualquier clínica que así lo desee.



**5-.COMPETENCIAS**  
**DESARROLLADAS Y/O**  
**APLICADAS**

### **5.1.-Competencias específicas**

En la creación de las aplicaciones, fueron necesarios conocimientos de programación, del lenguaje PHP, JavaScript, la librería FPDF, bases de datos en mysql, generación de pdf por medio de código y para el diseño de los sitios se utilizó html5 con css3.

Los conocimientos obtenidos durante el transcurso de la residencia fueron el producto de amplias investigaciones sobre el manejo de los lenguajes mencionados ya que al comenzar no se tenía conocimiento sobre JavaScript y la librería FPDF permitiendo obtener nuevos conocimientos y fomentar el aprendizaje continuo.

### **5.2.-Competencias Genéricas**

#### **5.2.1.-Instrumentales**

- Capacidad de organización y planificación
- Conocimientos generales básicos
- Comunicación oral y escrita en la lengua nativa
- Resolución de problemas
- Capacidad de gestión de la información
- Toma de decisiones

#### **5.2.2.-Sistémicas**

- Capacidad de aplicar los conocimientos en la práctica
- Habilidades de investigación
- Capacidad de aprender
- Capacidad para adaptarse a nuevas situaciones
- Creatividad
- Habilidad para trabajar de forma autónoma
- Iniciativa y espíritu emprendedor

#### **5.2.3.-Interpersonales**

- Desarrollar la capacidad de comunicación con un grupo de trabajo.
- Desarrollar empatía en el grupo.
- Capacidad de organizar.
- Desarrollar la habilidad de honestidad

## **6.-REFERENCIAS, BIBLIOGRÁFICAS Y VIRTUALES**

## Página web

- Login en PHP y MySQL, (21 abr. 2015)-Fecha de Consulta: 24 de Agosto de 2016. Disponible en:

<https://www.youtube.com/watch?v=JQVfFFaLlag>

- Listado de ejemplos, (n.d)-Fecha de Consulta: 2 de Septiembre de 2016. Disponible en:

<http://php.net/manual/es/indexes.examples.php>

- Realizando un CRUD (listar, registrar, actualizar, eliminar) con PHP, (21 Enero de 2016)-Fecha de Consulta: 5 de Octubre de 2016. Disponible en:

<http://anexsoft.com/p/57/realizando-un-crud-listar-registrar-actualizar-eliminar-con-php>

- Exportar HTML a PDF en PHP de manera fácil, (21 Enero de 2016)-Fecha de Consulta: 13 de Octubre de 2016. Disponible en:

<http://anexsoft.com/p/62/exportar-html-a-pdf-en-php-de-manera-facil>

- FPDF, (n.d)-Fecha de Consulta: 14 de Octubre de 2016. Disponible en:

<http://www.fpdf.org>