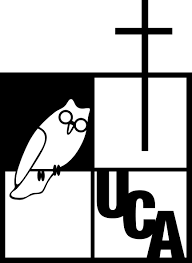
**UNIVERSIDAD CENTROAMERICANA**

**JOSÉ SIMEÓN CAÑAS**

****

**EXTENSIÓN DE APLICACIÓN PARA DISPOSITIVOS MÓVILES SISTEMA DE ATENCIÓN PRE HOSPITALARIA Y CONTROL DE AMBULANCIAS DEL SISTEMA DE**

**EMERGENCIAS MÉDICAS 132, PARA LA GESTIÓN DE INSUMOS Y MEDICAMENTOS EN AMBULANCIAS**

**TRABAJO DE GRADUACIÓN PREPARADO PARA LA**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**PARA OPTAR AL GRADO DE**

**LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

**POR:**

**RENÉ EDGARDO MASIN NÚÑEZ**

**CÉSAR ELMER PLEÍTEZ MEJÍA**

**JOSÉ ORLANDO POLANCO ZELADA**

**JESÚS EDUARDO RODRÍGUEZ MARTÍNEZ**

**OCTUBRE 2018**

**ANTIGUO CUSCATLÁN, EL SALVADOR, C.A.**

**RECTOR**

**ANDREU OLIVA DE LA ESPERANZA, S.J.**

**SECRETARIA GENERAL**

**SILVIA ELINOR AZUCENA DE FERNÁNDEZ**

**DECANO DE LA FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**CARLOS GONZALO CAÑAS GUTIÉRREZ**

**DIRECTOR DE LA CARRERA**

**ÁNGEL FERNANDO DUARTE NOVOA**

**DIRECTOR DEL TRABAJO**

**ERICK GIOVANNI VARELA GUZMÁN**

**LECTOR(A)**

**CECILIA AYALA**

**RESUMEN**

El Servicio de Emergencia Médicas del Ministerio de Salud, implemento una aplicación móvil con el nombre SAPHCA en el año 2017, para el uso de la aplicación se utilizan Tablet que cada ambulancia dispone, ya que anteriormente cuando se atendía una emergencia, los datos de la persona o las personas que requerían ayuda eran llenados en una ficha física, junto con la dirección a la que la ambulancia debía asistir. Este proceso quedo prácticamente obsoleto con la implementación de la aplicación móvil, debido a que el centro de emergencias médicas también dispone de un moderno software que recibe las llamadas y sus datos son ingresados en dicha aplicación, que a su vez traslada estos datos a la aplicación móvil, junto con la dirección en que debe ser atendida la emergencia, como se sabe al momento de atender una emergencia se es requerido de medicamentos para atender a la víctima.

Cada ambulancia recibe cierto inventario de medicamentos, pero el problema es que cuando se hace uso de estos, es que no existe un control de inventario que permita, por ejemplo, que en la Tablet se carguen todos los datos del inventario, y que estos solo sean descontados del inventario que posee cada ambulancia, ya que esto permitiría un mejor control de estos medicamentos. También permitirá actualizar el inventario de manera rápida, así los médicos que atienden la ambulancia estarían al tanto de cuando algún medicamento posea pocas unidades, y en caso de atender una emergencia no tenga la dificultad de quedarse sin poder administrar medicamentos a las víctimas, por ello el MINSAL requiere de un módulo de control de inventario que facilite todas estas funciones integrado en la aplicación SAPHCA.

Por ello en los siguientes capítulos se plantea desde el planteamiento del problema, los objetivos que se trazaron para poder realizar este módulo en la aplicación, las posibles limitantes al momento de realizar el proyecto. Además de que se realizó una investigación para estar más familiarizado con lo que son los dispositivos móviles, es decir, el tipo de dispositivos móviles que existen, la función de cada uno de estos, los sistemas operativos con los que funcionan, los tipos de sistemas operativos que existen para estos dispositivos, sus ventajas y desventajas. También se abordó el lenguaje de programación en el que se desarrollan estos sistemas operativos y las aplicaciones que son instaladas en estos.

Se ahonda un poco más también en los programas especializados que sirven para el desarrollo de estas aplicaciones para dispositivos móviles, además se investigó un poco más acerca la información que manejan estas aplicaciones, y como en manipulada, y recopilada en un solo lugar, este lugar donde se recopila toda la información, es una base de datos, y en estas se representan entidades de la vida real con atributos, por ejemplo la entidad persona en una base de datos podría recabar la información de esta como su estatura, su edad, su sexo, su comida favorita, entre otros, todos estos datos serían atributos, ya que cada persona tiene distintos atributos. También se habla un poco acerca programas que permiten el manejo de las bases de datos, y los gestores que permiten visualizar toda esta gran cantidad de información que pueden poseer las bases de datos.

Ya entrando en lo que sería el módulo, se planteó para este sus requerimientos, todos estos obtenidos a través de información recopilada mediante la investigación, es decir, reuniones que se tuvo con las personas que usan esta aplicación. Con estos datos se plantearon posibles soluciones, hasta que se pudo llegar a uno, para ello se hicieron muchos diseños previos de cómo sería el flujo de los componentes del módulo, las distintas pantallas con las que contaría, sus funciones, sus limitantes y como poder evitar posibles errores generados por las mismas limitantes. Se plantearon posibles escenarios al momento que el usuario este haciendo uso del módulo.

Se planteó también el diseño del sistema, es decir la manera en que se vería una vez estuviera listo, la información que necesita para funcionar, el mejor entorno para poder hacer posible el desarrollo del módulo y por supuesto una etapa de pruebas para verificar su correcto funcionamiento una vez esté en etapa de depuración. Por lo que es muy recomendable que MINSAL le saque provecho a este tipo de tecnología, y de paso haga más ágil el proceso de atención de las emergencias, claro que la capacitación del personal en el uso de estas tecnologías es clave, para que se haga buen uso de la misma.

**ÍNDICE**

RESUMENi

ÍNDICE DE FIGURASv

ÍNDICE DE TABLASvi

SIGLASvii

CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA 1

1.1 Generalidades1

1.2 Propuesta de solución2

1.3 Objetivos3

1.3.1 Objetivo general3

1.3.2 Objetivos específicos4

1.4 Alcances4

1.5 Limitantes5

1.6 Antecedentes5

CAPÍTULO 2. MARCO TEÓRICO6

2.1 Dispositivos móviles6

2.2 Sistemas operativos para dispositivos móviles10

2.2.1 Clasificación de SO10

2.2.2 Tipos de sistemas operativos para dispositivos móviles11

2.3 Lenguajes de desarrollo para aplicaciones móviles25 2.3.1 JAVA para Android25

2.3.1 Kotlin26

2.4 Bases de datos31

2.4.1 Sistema de administración de base de datos32

2.4.2 Tipos de base de datos34

2.5 Servicios Web38

2.6 Inventario42

CAPÍTULO 3. REQUERIMIENTOS DEL SISTEMA43

3.1 Flujo general del módulo en la aplicación SAPHCA44

3.2 Flujo del sistema de control de inventario de ambulancia 45

3.3 Requerimientos funcionales47

3.4 Requerimientos no funcionales50

CAPÍTULO 4. DISEÑO DEL SISTEMA52

4.1 Diseño del módulo y sistema web SICIA52

4.2 Base de datos52

4.2.1 Tablas utilizadas por el Modulo de control inventario de el SAPHCA53

4.2.2 Diccionario de Datos de las Tablas del Módulo de control de inventario 53

4.2.3. Diagrama de la Base de Datos 55

4.3 Diseño de interfaces56

4.3.1 Diseño de interfaces del módulo 56

4.3.2 Diseño de interfaces del sistema web SICIA 62

CAPÍTULO 5. DESARROLLO Y PRUEBAS DEL SISTEMA71

5.1 Metodología del desarrollo71

5.1.1. Distribución de labores de programación 71

5.1.2 Herramientas de desarrollo 72

5.2 Etapas de pruebas del sistema73

5.2.1 Metodología de la prueba73

CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES74

6.1 Conclusiones74

6.2 Recomendaciones75

REFERENCIAS76

ANEXOS

ANEXO A: Solicitud y vale de salida de material interno y hoja de transferencia de inventario

ANEXO B: Casos de uso

ANEXO C: Diccionario de datos

ANEXO D: Diagrama de la base de datos de la aplicación SAPHCA

ANEXO E: Pruebas unitarias

ANEXO F: Manual de usuario

ANEXO G: Manual técnico

**ÍNDICE DE FIGURAS**

Figura 2.1 Dispositivo móvil de datos limitados.7

Figura 2.2 Dispositivo móvil de datos básicos.7

Figura 2.3 Dispositivo móvil de datos mejorados. 8

Figura 2.4 Ejemplo de dispositivos móviles 9

Figura 2.5 Línea de tiempo de las versiones del sistema operativo Android.19

Figura 2.6 Línea de tiempo de las versiones del sistema operativo IOS.23

Figura 2.7 Línea de tiempo de las versiones del sistema operativo Windows Phone.25

Figura 2.8 Pantalla inicial de Android Studio.28

Figura 2.9 Presentación de todos los archivos de un proyecto de Android Studio.29

Figura 2.10 Ventana principal de Android Studio30

Figura 2.11 Ejemplo de los diferentes tipos de usuario que acceden a una base de datos.32

Figura 2.12 Representación de dos tablas en una base de datos. 35

Figura 2.13 Workbench de eclipse42

Figura 3.1 Flujo general del módulo de control de inventario de ambulancias en la aplicación SAPHCA. 44

Figura 3.2 Diagrama de flujo de sistema SICIA45

Figura 4.1 Parte del diagrama de la base de datos de SAPHCA 56

Figura 4.2 Diseño de la pantalla inicial del módulo de inventario58

Figura 4.3 Pantalla inicial del módulo de inventario 58

Figura 4.4 Diseño de la pantalla de ajuste por incidencia60

Figura 4.5 Pantalla de ajuste de inventario por incidencia.60

Figura 4.6 Diseño de la pantalla del ingreso de consumo.61

Figura 4.7 Pantalla del ingreso de consumo 62

Figura 4.8 Diseño de pantalla de inicio de sesión del sistema web SICIA.63

Figura 4.9 Pantalla de inicio de sesión del sistema web SICIA.63

Figura 4.10 Diseño de la pantalla principal del sistema web SICIA.64

Figura 4.11 Pantalla principal del sistema web SICIA.65

Figura 4.12 Diseño de la pantalla asignar inventario66

Figura 4.13 Pantalla asignar inventario67

Figura 4.14 Diseño de pantalla del módulo Consultar inventarios del sistema web SICIA.68

Figura 4.15 Pantalla del módulo Consultar inventarios del sistema web SICIA.68

Figura 4.16 Diseño de pantalla del módulo Consultar operaciones de inventario del SICIA.70

Figura 4.17 Pantalla del módulo Consultar operaciones de inventario del SICIA 70

**ÍNDICE DE TABLAS**

Tabla 3.1 Estructura de casos de uso 47

Tabla 3.2 Caso de uso “Acceder al módulo de control de inventario (SAPHCA) 48

Tabla 3.3. Listado de casos de uso 49

Tabla 4.1. Listado de tablas que se utilizarán en el módulo53

Tabla 4.2. Detalle de los campos de la tabla “operacion”54

Tabla 4.3. Detalles de los campos de la tabla “inventario” 54

Tabla 5.1. Formato de pruebas unitarias. 74

Tabla 5.2. Prueba unitaria “Acceso al módulo de control de inventario (SAPHCA)” 74

**SIGLAS**

AEP: Android Extension Pack (Extensión de paquete Android)

AOSP: Android Open Source Project (Proyecto Android de Código Abierto)

AOT: Ahead-Of-Time (Compilación anticipada)

API: (Application Programming Interface) Interfaz de Programación de Aplicaciones

APK: Android Package (Aplicación de Paquetes de Android)

ART: Android Runtime (Tiempo de ejecución de Android)

BD: Base de datos

BOSEM: Base Operativa del Sistema de Emergencias Médicas

CRUD: Create, Read, Update and Delete (Crear, leer, actualizar y borrar)

DBMS: Sistemas de Administración de Bases de Datos (Database Management System)

DCL: Data Control Language (Lenguaje de control de datos)

DDL: Data Definition Language (Lenguaje de definición de datos)

DML: Data Manipulation Language (Lenguaje de manipulación de datos)

DQL: Data Query Language (Lenguaje de consulta de datos)

DTIC: Dirección de Tecnologías de Información y Comunicaciones

FOSALUD: Fondo Solidario para la Salud

GC: Garbage Collector (Recolector de basura)

GPS: Global Positioning System (Sistema de Posicionamiento Global)

GUI: Graphic User Interface (Interfaces gráficas de usuario)

HTTP: Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto)

IDE: Integrated Development Environment (Entornos de desarrollo integrado)

JSF: JavaServer Faces (Herramienta para aplicaciones Java basadas en web)

JSON: JavaScript Object Notation (Notación de Objetos de JavaScript)

JVM: Java Virtual Machine Máquina Virtual de Java

MINSAL: Ministerio de Salud

MSQL: Microsoft SQL Server (Servidor de Microsoft SQL)

ORACLE: Oracle Corporation (Corporación Oracle)

PDA: Personal Digital Assistant (Asistente Personal Digital)

RAM: Random Access Memory (Memoria de Acceso Aleatorio)

REST: Representational State Transfer (Transferencia de Estado Representacional)

SDK: Software Development Kit (Kit de Desarrollo de Software)

SEM: Sistema de Emergencias Médicas

SICIA: Sistema web de control de inventario de ambulancia (SICIA)

SINAB: Sistema Nacional de Abastecimientos

SMS: Short Message Service (Mensajes cortos de texto)

SO: Sistema operativo

SQL: Structured Query Language (Lenguaje de consulta estructurado)

TSQL: Transact-SQL

WAP: Wireless Application Protocol (Protocolo de Aplicaciones Inalámbricas)

WSDL: Web Services Description Language (Lenguaje de Descripción de Servicios Web)

W3C: World Wide Web Consortium (Consorcio Mundial de la Red)

XML eXtensible Markup Language (Lenguaje de Marcas Extensible)

**CAPÍTULO 1. PLANTEAMIENTO DEL PROBLEMA**

* 1. **PROBLEMÁTICA**

El Ministerio de Salud (MINSAL) actualmente ha desarrollado una aplicación móvil para el Sistema de Emergencias Médicas (SEM), que realiza la función de coordinar, a través de recursos medios y tecnológicos provistos por Fondo Solidario para la Salud (Fosalud), con el fin de proveer atención a víctimas que han sufrido lesiones o alguna enfermedad repentina, para que puedan ser atendidos en el sitio de la emergencia o trasladar al paciente al hospital que corresponda de forma gratuita.

Fosalud tiene la función de proveer de recursos tanto físicos como tecnológicos, así también, personal para ambulancias, medicamento e insumos para pacientes, entre otros, al SEM. MINSAL, tiene la función de proveer recursos informáticos (y darle soporte a los mismos), tales como, un sistema de control de llamadas, una aplicación móvil (en la cual se hará énfasis), entre otros.

El proceso actual para la atención de una emergencia dentro del SEM es el siguiente:

Se realiza una llamada por parte de la víctima o paciente al centro de llamada de emergencias del SEM, está llamada o emergencia es registrada en el sistema de control de llamadas, una vez registrados los datos necesarios de la emergencia (datos del paciente, ubicación, entre otros.) se le envía una solicitud para atender la emergencia a una ambulancia disponible; cada ambulancia, tiene un dispositivo electrónico, en este caso una Tablet, con la aplicación móvil descrita anteriormente; cuando el SEM realiza la solicitud, esta llega automáticamente a alguna ambulancia disponible mediante la aplicación móvil, si no hubiese alguna disponible, la solicitud queda en modo de espera. Una vez realizada la solicitud, la ambulancia se dirige al lugar de destino a atender la emergencia.

Anteriormente, se llevaba un formulario en formato físico, que contiene la información del paciente o víctima, así como los medicamentos e insumos consumidos al atender la emergencia (si se usasen).

La aplicación móvil facilita el proceso anteriormente descrito, ya que permite ingresar los datos necesarios para el formulario dentro de la aplicación, mediante diferentes módulos (u opciones), consolidando toda la información necesaria en un reporte digital, conocido dentro de la institución como registro médico.

La aplicación móvil permite llevar un mejor control del registro médico, así como presentarlo en forma digital.

Esta investigación estará enfocada en el inventario de medicamentos dentro de la ambulancia; actualmente, existe un módulo en la aplicación móvil en el que el personal de la ambulancia ingresa el medicamento consumido al momento de la emergencia; este medicamento pertenece al inventario de medicamentos de la ambulancia, el cual, al mismo tiempo, pertenece al inventario de medicamentos de las bodegas del SEM (BOSEM), existiendo un total de cuatro bodegas ubicadas estratégicamente para cubrir la zona rural y urbana de San Salvador.

Cada ambulancia tiene asignada una BOSEM, y esta se encarga de abastecer a las ambulancias de los medicamentos que se utilizan al atender una emergencia; a su vez, cada BOSEM es abastecida por Fosalud. El SEM, mediante un acuerdo con Fosalud, le permite realizar la compra directa de medicamentos e insumos, también puede abastecer a cada BOSEM independientemente.

Cada BOSEM maneja su inventario (de bodega) mediante una aplicación web conocida como SINAB, esta aplicación no tiene relación alguna con la aplicación móvil, al mismo tiempo, en la aplicación móvil no está contemplado ningún tipo de inventario para la ambulancia, por lo que cada salida de inventario ingresada en SINAB, se maneja como una salida tal cual, debiéndose manejar como una salida de inventario de BOSEM y una entrada para inventario de ambulancia.

El control de inventario de cada ambulancia se maneja en formato físico, dependiendo de cada encargado de ambulancia, esto quiere decir que al registrar una salida de inventario dentro de SINAB, se registra manualmente como una entrada para el inventario de ambulancia, por lo que cada salida de inventario (o consumo) dentro de la ambulancia también debe registrarse manualmente.

En cada BOSEM hay asignadas dos ambulancias, cada una con su encargado, por lo que se manejan formatos distintos de inventario de ambulancias para una misma BOSEM; al mismo tiempo, se necesita llevar un control de inventario para el turno matutino, así como para el turno vespertino en un mismo día, por lo que el trabajo para los encargados de bodega se duplica.

* 1. **PROPUESTA DE SOLUCIÓN**

Con la intención de proveer un control de inventario de medicamentos dentro de la ambulancia para las BOSEM, se plantea el análisis y desarrollo de un módulo de inventario de medicamentos de ambulancias que será incluido en la actual aplicación que ya utilizan. El módulo contará con las opciones de consulta de inventario de la ambulancia, así como un módulo de ajustes de inventario, para casos de medicamentos dados de baja por fecha de vencimiento o accidentes al momento de manipularlos, además contará con la opción de registrar un consumo medicamento.

El ingreso de medicamentos al inventario de ambulancias se realizará automáticamente, mediante un servicio web, al registrar una salida de inventario de BOSEM dentro de la aplicación web del Sistema Nacional de Abastecimientos (SINAB).

La salida de medicamentos por consumo, serán registradas desde una pantalla, que podrá ser accedida desde una de las opciones que mostrará el módulo de inventario que se desarrollará, en ella el usuario deberá seleccionar el medicamento que se consumió y que cantidad fue consumida, al mismo tiempo se podrán agregar más medicamentos para los casos en los que se desee registrar más de un consumo. Finalmente, el usuario deberá presionar un botón con la opción registrar, para los casos de ajuste de inventario se procederá de la misma manera, pero con la diferencia de que en estos casos el usuario deberá ingresar la justificación correspondiente, es decir deberá especificar si se trata de ajuste por vencimiento o por una mala manipulación de los mismos.

Adicionalmente se propone el análisis y desarrollo de una aplicación web, cuyo fin principal será proveer de informes que contengan el detalle del inventario de una ambulancia específica, seleccionada por el usuario de esta aplicación, además de proveer informes con los detalles de consumos de medicamentos registrados desde el módulo de inventario que se desarrollará para la aplicación SAPHCA. Estos informes se podrán generar en formato PDF y XLS. Los usuarios que podrán acceder a esta aplicación serán aquellos que la Dirección de Tecnologías de Información y Comunicaciones del MINSAL tenga almacenados en su base de datos correspondiente.

* 1. **OBJETIVOS**
     1. **General**

Desarrollar un módulo para la aplicación SAPHCA implementada por la Dirección de Tecnologías de Información y Comunicaciones (DTIC) del MINSAL que permita un control de inventario de medicamentos de las ambulancias que pertenecen al Sistema de Emergencias Médicas por medio del ingreso y registro de formularios de salidas de medicamentos, así como también desarrollar una aplicación web para generar informes sobre inventarios.

* + 1. **Específicos**
* Conocer y analizar el proceso que actualmente se utiliza como mecanismo de control de inventario de las ambulancias, con el propósito de llevar este proceso a una solución computacional óptima y eficaz.
* Adquirir los conocimientos necesarios sobre las herramientas a utilizar para el correcto desarrollo e integración del módulo con la aplicación actual.

* Conocer el estado actual y el funcionamiento de la aplicación Android a la que se deberá realizar el módulo, por medio del estudio del código fuente y logrando ejecutarla en el entorno de desarrollo del equipo de investigación.

* Establecer requerimientos que el módulo a desarrollar deberá satisfacer y que serán prioridad en el proceso de desarrollo.
  1. **ALCANCES**

El proyecto consta del análisis, diseño, desarrollo e integración de un módulo para la aplicación actual, siendo su principal requerimiento el registro de salidas de medicamentos en las ambulancias del SEM con el objetivo de mantener un control sobre el inventario de medicamentos y con esto ayudar al personal del SEM a obtener informes de forma automatizada y en un menor tiempo comparado con el proceso actual.

El módulo deberá ofrecer una pantalla donde se muestre un formulario donde se ingrese la información sobre salidas o consumo de medicamento en las ambulancias. Además, deberá ser capaz de enviar y almacenar la información en un servidor remoto donde se tendrá la base de datos, eso mientras tenga acceso a internet. Una vez ingresado un formulario, no debe ser posible su modificación por lo que se debe hacer un mecanismo de confirmación antes de ingresar un formulario.

Para ver los registros y generar informes de entradas y salidas de medicamento de las ambulancias se hará una pequeña aplicación web desarrollada con JAVA la que permitirá generar archivos Excel o PDF creados a partir de parámetros de acotamiento, como determinadas fecha o ambulancias.

* 1. **LIMITANTES**
* Los requerimientos del proyecto no están bien definidos, por lo tanto; puede que en el desarrollo del proyecto aparezcan nuevos requerimientos que entorpezcan el resultado final.

* Si bien el principal objetivo de este proyecto es el análisis, diseño e implementación de un módulo en la APP actualmente utilizada, no se plantea brindar capacitaciones al personal que utilizará el módulo.

* El cambio del lenguaje de programación, así como cambios en el modelo o arquitectura de programación, cuando el proyecto ya esté iniciado, podría afectar de manera drástica el resultado esperado por el MINSAL y los tiempos previamente estimados. Además, que el proyecto debe adaptarse a las tecnologías que se utilizan por el sistema informático del MINSAL.
  1. **ANTECEDENTES**

Como parte del desarrollo del nuevo módulo para la App SAPHCA, se cuenta con información de la misma, está como ya se ha dicho antes, pertenece al DTIC del MINSAL, pero, fue desarrollada por una empresa consultora llama “Administraciones y Sistemas S.A” en 2017.

La APP fue creada con el objetivo de poder enviar estado de las ambulancias (si está disponible, si está asignada, cuando se dirige al lugar, cuando está en el lugar, en carga, en atención, cuando se dirigen hacia el hospital, cuando se está trasladando al paciente de la ambulancia al hospital, entre otros) al SEM y así llevar un mejor control; de poder georreferenciar las ambulancias que están conectadas, desde el SEM enviar la ubicación del destino al que deben dirigirse que se envía mediante Web y así que dicha ambulancia llegue al lugar exacto. También recolectar la información de los pacientes (nombres, apellidos, dirección, sexo, edad) y los medicamentos que se le fueron aplicados e insumos que fueron utilizados, todo para ser enviado en tiempo real al SEM.

**CAPÍTULO 2. MARCO TEÓRICO**

* 1. **DISPOSITIVOS MÓVILES**

Es un dispositivo electrónico de tamaño muy reducido creado con el fin de ser portable, estos dispositivos cuentan con las siguientes características:

* Capacidad de procesamiento de información.
* Capacidad de conectarse a una red de manera temporal o permanente.
* Con cierta capacidad de almacenamiento limitado.
* Con otras funciones dependiendo del uso al que el dispositivo está destinado.

Dichos dispositivos suelen poseer una pantalla para visualizar contenido, además de que pueden poseer un teclado para el ingreso de datos u otra información; este teclado suele ser acorde al tamaño del dispositivo e incluso táctil en la pantalla, además a la mayoría de estos dispositivos se les puede conectar algún medio periférico como es el caso de teclado, mouse, entre otros. Otras características que esta clase de dispositivos posee son que algunos de ellos tienen un precio muy asequible con respecto a una computadora, este puede ser actualizable o no, en caso que no se pueda actualizar este con el paso de los años deberá ser reemplazado por uno nuevo, la forma de manejarlos puede ser más sencilla comparada con otros dispositivos, y su funcionalidad puede llegar a ser la misma que la de otros dispositivos de mayor tamaño.

En el año 2005 unas empresas propusieron ciertos estándares para la definición de dispositivos móviles. Dichos estándares son los siguientes:

* Dispositivo móvil de datos limitados: Es un dispositivo que posee una pantalla pequeña, que se basa en un tipo de pantalla tipo texto con servicios de datos limitados por lo general a Mensajes cortos de texto (SMS) y acceso a Protocolo de Aplicaciones Inalámbricas (WAP) como ejemplo básico de estos dispositivos están los teléfonos móviles clásicos. En la figura 2.1 se muestra un ejemplo de dispositivo móvil de datos limitados.



Figura 2.1. Dispositivo móvil de datos limitados.

* Dispositivo móvil de datos básico: Es un dispositivo que posee una pantalla mediana; aproximadamente de 120 x 120 y 240 x 240 píxeles, además se caracteriza por disponer de un menú o navegación que se basa en iconos a los cuales se accede mediante un cursor táctil en la pantalla, o con botones físicos como se solía implementar. Dicho dispositivo ofrece acceso al correo electrónico, lista de direcciones, navegador de web básico, SMS. Como principal ejemplo de dichos dispositivos están los teléfonos inteligentes. En la figura 2.2 se muestra un ejemplo de dispositivo móvil de datos básicos.



Figura 2.2. Dispositivo móvil de datos básicos.

* Dispositivo móvil de datos mejorados: Estos dispositivos poseen una pantalla por encima de los 240 x 120 píxeles; la navegación en estos equipos se caracteriza por ser de tipo “Stylus” es decir mediante un lápiz táctil, este posee las mismas características de los dos tipos de dispositivos mencionados con anterioridad, además de aplicaciones corporativas, y los más importante: sistemas operativos específicos para el correcto funcionamiento de estos dispositivos. En la figura 2.3 se muestra un ejemplo de dispositivo móvil de datos mejorados.



Figura 2.3. Dispositivo móvil de datos mejorados.

Existen muchos tipos de dispositivos móviles, algunos de estos son: paginadores, teléfonos móviles, sistemas de televisión e internet, sistema de navegación de automóviles, sistemas de entretenimiento, Asistente Personal Digital (PDA), en la figura 2.4 se pueden ver algunos ejemplos.

Un paginador es un dispositivo muy simple, ya que es similar a un radio, con la excepción que este escucha un solo canal todo el tiempo; mientras que un radio transmisor se encarga de emitir la señal para los diferentes paginadores que están en esa red específica. Todo esto se logra gracias a que posee un receptor por el cual recibe la señal mediante un código, el paginador escucha este código que es único para cada dispositivo, una vez recibe este código, activa una alerta que informa al usuario del paginador que ha recibido una señal.

El teléfono móvil es un dispositivo electrónico portátil, con la característica de ser inalámbrico, y permite tener acceso a la red de telefonía celular o móvil; la comunicación mediante llamadas de voz, haciendo uso de los SMS; además brinda acceso a internet mediante conexión inalámbrica, y otras grandes funcionalidades que con el pasar de los años estos dispositivos han adquirido.

Sistema de televisión e internet es un dispositivo electrónico que se conecta a un televisor que posee conector de audio y video; dicho dispositivo permite la navegación por internet usando un mando a distancia, este mando puede actuar como un mouse y/o un teclado inalámbrico; por ello permite asimilar algunas funcionalidades de una computadora en la televisión.

El sistema de navegación de automóviles es un Sistema de Posicionamiento Global (GPS), creado para la navegación de los vehículos en las carreteras, todo con el fin de proveer a los automóviles, autobuses y camiones de información para que se puedan dirigir a ciertos lugares, y también indicando las mejores rutas para el desplazamiento de estos automotores; todo este proceso se hace de manera remota.

El PDA es un dispositivo móvil que es conocido también como un organizador de bolsillo; este es un ordenador que está compuesto por un procesador, almacenamiento interno limitado, Memoria de Acceso Aleatorio (RAM), acceso a una red de manera inalámbrica, y cuenta con una pantalla táctil. Este es un asistente que ofrece múltiples funciones como, por ejemplo: agenda, administrador de tareas, libreta de direcciones y el uso del correo electrónico mediante una aplicación interna.



Figura 2.4. Ejemplo de dispositivos móviles: a) teléfono móvil, b) Sistema de navegación de automóviles, c) paginador, d) Sistema de televisión e internet, e) PDA.

* 1. **SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES**

Un sistema operativo(SO) es una herramienta de software, diseñada y creada para cumplir con la función de administrar los recursos (o hardware) del dispositivo sobre el cual se ejecuta, de manera que los procesos de programas externos al SO, como por ejemplo un explorador de internet, tengan acceso controlado por medio de la asignación de los recursos que necesita la aplicación para funcionar de forma correcta, logrando así evitar que una aplicación haga uso completo de todos los recursos disponibles en el dispositivo y dejando a otras sin estos.

Los recursos a los que se hace referencia pueden ser tiempo de procesador, memoria, entre otros. Además, se encarga de la gestión de entradas y salidas de periféricos a través de controladores para facilitar la interacción de los periféricos con el SO.

Un controlador es un programa que indica al SO la manera en que se debe comunicar con un periférico y cómo controlarlo de forma correcta para lograr un funcionamiento idóneo y sin errores.

Por otra parte, un SO también se encarga de la seguridad de los recursos, es decir, que garantiza que estos sean utilizados y accedidos únicamente por aquellos programas y usuarios que cuentan con los permisos necesarios, esto es importante ya que se evita que un programa malicioso tenga acceso a estos.

Un ejemplo de programa malicioso podría ser uno cuyo propósito sea utilizar el 100% de los recursos, lo que significa que el dispositivo deje de funcionar completa o parcialmente, debido a que el SO no podrá contar los recursos necesarios para cumplir con las tareas de asignación de recursos a los demás procesos que los necesiten.

Otra función de un SO es la de gestionar la lectura y escritura del sistema de archivos a la vez que proporciona un sistema de autorización de acceso a archivos de los programas y usuarios del dispositivo.

* + 1. **Clasificación de SO**

Los SO se pueden clasificar por: el modo de administración de tareas, modo de administración de usuarios y modo de manejo de recursos.

Modo de administración de tareas: dentro de esta clasificación se encuentra los mono tarea, son aquellos SO que únicamente pueden ejecutar un programa a la vez.

Este tipo de SO, con el tiempo fueron reemplazados por SO de tipo multitarea que, contrario a los mono tarea, permiten la ejecución de varios programas o procesos, esto se logra compartiendo tiempo de procesador, de forma que hay una cola de procesos esperando para utilizar el procesador.

Un proceso ocupa el procesador cierta cantidad de tiempo, pasado ese tiempo se le permite al siguiente proceso de la cola, utilizar el procesador y así sucesivamente, actualmente esto sucede tan rápido que da la sensación de que todo se ejecuta al mismo tiempo.

Modos de administración de usuarios: los SO también pueden ser clasificados por la cantidad de usuarios que pueden utilizarlo, monousuario es aquel SO que permite la ejecución de programas de un solo usuario a la vez, por otra parte, están los SO multiusuario que, como su nombre indica, permite la ejecución de programas de varios usuarios a la vez y que comparten los mismos recursos.

Modo de manejo de recursos: en esta clasificación se encuentran los SO centralizados y distribuidos, su principal diferencia es que los centralizados hacen uso y gestionan los recursos de un único dispositivo o computadora, y los distribuidos se encargan de gestionar recursos de varios dispositivos o computadoras lo cual significa una ventaja pues permite dividir la carga de trabajo entre los dispositivos que gestiona.

* + 1. **Tipos de sistemas operativos para dispositivos móviles**

Con el surgimiento de dispositivos móviles, como celulares y tabletas, fue necesaria la creación de un SO orientado para móviles, ya que estos dispositivos al ser móviles tienen otras exigencias como conexiones inalámbricas, por lo que, los fabricantes de estos dispositivos destinaron recursos para crear SO que se adaptan de mejor forma a sus dispositivos, fue así como surgieron los siguientes sistemas operativos:

1. Android.
2. IOS.
3. BlackBerry OS.
4. Symbian.
5. Windows Phone.
6. Firefox OS.

A continuación, se detallan las principales características de estos SO.

* **Sistema operativo Android**

Este SO en principio fue desarrollado por Android Inc. con el apoyo económico de la empresa Google, empresa que un tiempo después compraría a Android Inc. Google es una empresa especializada en ofrecer productos y servicios asociados a internet, software, dispositivos electrónicos entre otras tecnologías, que a día de hoy es de las más conocidas a nivel mundial.

Este SO fue presentado por primera vez en el año 2007 en su versión beta, su versión más básica es conocida como Android Open Source Project(AOSP) y el primer celular que lo utilizó fue el HTC Dream, celular que fue sacado a la venta en el año 2008, por la empresa HTC Corporation, este celular contaba con la versión 1.0 de Android. Es importante mencionar que Google decidió liberar gran parte del código de Android bajo la licencia Apache, una licencia libre y de código abierto, esto implica que el código puede ser copiado, modificado y distribuido sin ningún tipo de restricción por parte de Google en este caso.

Con el lanzamiento de Android al mercado de los dispositivos móviles, empieza a competir con otros SO y esto provoca que se buscará la forma de dar más y mejores funcionalidades a los dispositivos que hacían uso de él, lo que se traduciría en una evolución del SO, a través del lanzamiento de versiones cada cierto tiempo, cuyos nombres hacen alusión a postres. A continuación, se detallan las versiones y los cambios más importantes entre versiones:

Android 1.1 Apple Pie: El lanzamiento de esta versión surgió el 23 de septiembre de 2008, se presenta como un SO totalmente gratuito, y tenía mucho margen de mejora y apenas inquietó a las competencias, pero ya iniciaba un cambio que con los años posteriores se convertirían en estándar para los SO móviles, las características principales eran las siguientes:

* Pestañas de notificaciones.
* Android Market.
* Aplicación para Gmail.
* Navegador web, calculadora, reloj.
* Mapas de Google.
* Era capaz de actualizarse a través de conexión a internet.
* Reproductor multimedia.
* Conexión Wi-Fi y Bluetooth.
* Mensajes de texto y llamadas.

Android 1.5 Cup cake: Su lanzamiento fue el 27 de abril de 2009, y con esta versión Android mostraría grandes mejoras, principalmente se hicieron cambios casi inesperados sobre algunos elementos de la interfaz, pero las novedades más interesantes las encontraríamos a nivel de usabilidad con nuevas características:

* Teclado táctil QWERTY en pantallas.
* Soporte de Widget.
* Grabación y reproducción de videos MPEG-4 y 3GP.
* Opciones copiar y pegar en el navegador web.

Android 1.6 Donut: Su lanzamiento fue el 15 de septiembre de 2009, y apareció con algunas novedades adicionales, ya que toda la interfaz sufrió cambios visuales y se agregaron algunos elementos, pero lo realmente atractivo se encontraba en el núcleo del SO en los cuales encontraríamos los detalles siguientes:

* Mejora en la búsqueda por entrada de texto y voz.
* Galería, cámara y videocámara con mejor integración, con rápido acceso a la cámara.
* Soporte para resoluciones de pantalla con 480 pixeles de altura.
* Mejoras de velocidad en búsqueda y aplicaciones de cámara.
* Framework de gestos ampliados y una nueva herramienta de desarrollo GestureBuilder.

Android 2.0 Eclair: El Lanzamiento fue el 26 de octubre de 2009, se realizaron cambios más sustanciales tanto a nivel de diseño como de arquitectura interna, era una versión dirigida a dispositivos con mayor tamaño. Para la versión 2.1 se vieron cambios mínimos, ya que mantuvo las mismas características de la versión 2.0 y solo se corrigieron algunos fallos, pero su uso fue mayor entre los fabricantes respecto a las versiones anteriores. Principales características:

* Sincronización cuenta expandida.
* Soporte intercambio de correo.
* Soporte Bluetooth 2.1.
* Habilidad para tocar una foto de un contacto.
* Nuevas características para la cámara, flash, zoom digital, modo escena, entre otros.
* Mejorada velocidad de tipeo en el teclado virtual.
* Optimización en velocidad de hardware y una interfaz de usuario renovada.
* Soporte para más tamaños de pantalla y resoluciones, con mejor ratio de contraste.
* Mejorado Google Maps 3.1.2.
* Clase MotionEvent mejorada para rastrear eventos multi-touch.

Android 2.2 Froyo: Su lanzamiento fue el 20 de mayo de 2010, esta nueva versión trajo numerosos cambios, se puede decir que, copiados de otros ROMS, y otros con la mentalidad en una figura empresarial. One fue el primer Móvil que se actualizo. También se realizaron actualizaciones en esta versión 2.2.1, 2.2.2, 2.2.3 para corregir los errores en todo el 2010, algunas de sus principales características fueron:

* Optimizaciones en velocidad, memoria y rendimiento.
* Mejoras adicionales de rendimiento de aplicación.
* Integración del motor de JavaScript V8.
* Funcionalidad de anclaje de red por USB y Wi-Fi hotspot.
* Cambio rápido entre múltiples lenguajes de teclado y diccionario.
* Dictado por voz e intercambio de contactos por Bluetooth.
* Soporte para docks Bluetooth-habilitado para autos y de escritorio.
* Soporte para contraseñas numéricas y alfanuméricas.
* Soporte para instalación de aplicaciones en la memoria expandible.
* Soporte para Adobe Flash.

Android 2.3 Gingerbread: Su lanzamiento fue el 06 de diciembre de 2010, se establece como una continuación de políticas en las actualizaciones de SO en la compañía por el lanzamiento de nuevos teléfonos esta vez aliados con la marca SAMSUNG, y desde entonces esta versión de Android es la más extendida durante los siguientes años, tuvo actualizaciones para esta versión para corregir errores. Características:

* Actualización del diseño de interfaz.
* Soporte para pantalla extra grande.
* Soporte mejorado para desarrollo de código nativo.
* Recolección de eventos concurrentes.
* Un administrador de descarga.

Android 3.1 Honeycomb:Su lanzamiento fue el 22 de febrero de 2011, y se trata de una versión específica para tabletas, no compatibles con teléfonos, que introducía la línea maestra de la interfaz en el futuro. Las versiones 3.1 y 3.2 fueron básicamente un conjunto de correcciones:

* Mejor soporte para tabletas.
* Mejoras en el navegador web.
* Sistema multitarea mejorado.
* Soporte para videochat.
* Soporte para una variedad de periféricos.
* Accesorios con conexión USB.

Android 4.0 Ice Cream Sandwich: Lanzado el 19 de octubre de 2011, es una versión basada en la versión anterior (Honeycomb) pero optimizada para Smartphone, esta es una versión con los cambios más importantes sufridos por Android en teléfonos, y es considerado como la versión principal de cara al futuro, y se convirtió en el SO móvil más usado en el mundo. Sus principales características son:

* Interfaz limpia y moderna
* Gestor de tráfico de datos de internet
* Corrector de texto rediseñado y mejorado
* Capturas de pantallas con 2 botones
* Reconocimiento de voz del usuario

Android 4.1 Jelly Bean: Se lanzó el 30 de junio de 2012, mediante una conferencia y fue una actualización incremental con el enfoque primario de mejorar la funcionalidad y el rendimiento de la interfaz, la estrategia de Android en tabletas, añadiendo nuevas características como:

* Mejora la fluidez de estabilidad.
* Dictado por voz mejorado.
* Notificaciones mejoradas con acceso más rápido.
* Google Chrome se convierte en el navegador por defecto de Android.

Android 4.4 kitkat: El lanzamiento fue el 31 de octubre de 2013, ofrece nuevas posibilidades al usuario al mismo tiempo que corrige varios defectos de su versión anterior como, por ejemplo:

* Se sustituyen elementos de la interfaz de azul a blanco hecho por Guillo
* Las horas del reloj ya no se muestran con números en negrita, tanto minutos como horas son finos.
* Transparencias en la barra de estado y barra de navegación.
* Introducción del Modo inmersivo en el que tanto la barra de estado como la barra de navegación se oculta en determinadas aplicaciones para una visualización a pantalla completa.
* Optimización y rendimiento en dispositivos de especificaciones técnicas comedidas, así como la implementación de memorias Random Access Memory (RAM).
* Posibilidad de impresión mediante WiFi.
* Navegador interno basado en el motor de Chromium.
* Nuevo marco de transiciones y efectos visuales.
* Desactivado el acceso a las estadísticas de batería a aplicaciones de terceros.

Android 5.1 Lollipop:Lanzamiento 03 de noviembre de 2014, en esta versión uno de los rasgos más llamativos fue la inclusión de un nuevo lenguaje de diseño que unificó la experiencia de uso de cualquier tipo de dispositivo. Sus características y mejoras:

* Android Runtime (ART) con compilación anticipada (AOT) y mejora de la recogida de basura (GC).
* Soporte para CPU con arquitectura de 64 de bits.
* Opengl ES 3.1 y Android Extension Pack (AEP) en configuraciones de Unidad de procesamiento gráfico (GPU) soportadas.
* Pantalla de actividades recientes con tareas en lugar de aplicaciones, hasta un máximo configurado de tareas por aplicación.
* Vectoriales dibujables, que escala sin perder definición.
* Soporte para vistas previas de impresión.
* Pantalla de bloqueo refrescada y ya no soporta widgets.
* Bandeja de notificación refrescada y configuraciones rápidas desplegable.
* Project Volta, para las mejoras de la vida de la batería.

Android 6.0 Marshmallow:Lanzamiento el 05 de octubre de 2015, en al mismo tiempo se daba a conocer una nueva generación de dispositivos Nexus, el cual, junto con sus generaciones anteriores, fueron los primeros en recibir Android Marshmallow. Entre los cambios a destacar en la versión Marshmallow de Android encontramos:

* Sistema de Permisos rediseñado. Ahora sólo hay 8 categorías de permisos.
* Los usuarios pueden conceder o denegar permisos individuales a las aplicaciones cuando lo requieran.
* Soporte nativo para reconocimiento de huellas dactilares.
* Nuevo Sistema de administración de energía llamado “Doze”.
* Compatibilidad con USB Tipo-C.
* Capacidad de Carga hasta 5 veces más rápida.
* Introducción de enlaces verificados.

Android 7.0 Nougat: Lanzamiento el 22 de agosto de 2016, como siempre viene con numerosas novedades y mejoras tanto en el apartamento funcional como el estético que se resume a continuación:

* Se mejoran las animaciones.
* Se incorpora la opción de multi-ventana de forma nativa.
* Es posible arrastrar contenido de una aplicación a otra.
* Optimización de uso de RAM de aplicaciones en segundo plano.
* Las instalaciones se realizan más rápidamente gracias a que no se compilan durante la instalación, sino en ejecución de la misma o en reposo.
* Nueva API Vulkan con mejor rendimiento gráfico.
* Nuevo economizador de consumo de datos.
* Usa Chrome como WebView lo que ahorra consumo en RAM ya que las aplicaciones no tendrán que cargar este componente.
* Posibilidad de ajustar la importancia de las notificaciones.
* Incorporación de JAVA 8.
* Posibilidad de volver a abrir una aplicación directamente cuando ha tenido un fallo.
* Nuevo centro de notificaciones. (Se pueden cambiar directamente los iconos que se deseen ver al deslizar los dedos hacia abajo).
* Las notificaciones entrantes se pueden programar para que no aparezcan de forma continua o evitarlas por un lapso de tiempo.
* Es posible responder mensajes de texto desde la notificación.

Android 8.0 Oreo:Esta nueva versión no presenta grandes cambios a nivel de interfaz, pero sí promete dar respuesta a una de los problemas que aquejan los clientes: la gestión de la batería. Será más inteligente, pero, sobre todo, más rápido. Principales características:

* Los celulares podrán encenderse hasta dos veces más rápido.
* Los emoticones se actualizan con un nuevo diseño.
* La selección de texto usará inteligencia artificial para no sólo seleccionar una palabra cuando mantienes presionada la pantalla, sino que selecciona nombres completos, direcciones y números telefónicos.
* El menú emergente que aparece cuando seleccionas texto ahora entenderá el contexto para abrir Google Maps cuando seleccionas una dirección, o abrir la app de teléfono cuando seleccionas un número.
* Google Play Protect te mantendrá más informado de la seguridad de tus apps.
* Pantalla siempre ambiental muestra iconos similares a los que ofrece el LB G6 y los Samsung Galaxy S8 en su pantalla siempre encendida.
* Nuevo fondo de pantalla.
* La app de descargas se convierte en app de archivos y funciona como un administrador de documentos y archivos.
* En las opciones ocultas, es posible personalizar los botones de navegación de Android, permitiendo inclinarnos hacia un lado o añadir hasta dos nuevos botones más.
* Además, en estas opciones ocultas, es posible personalizar los atajos de la pantalla de bloqueo.
* En los Google Pixel, los botones de navegación se oscurecen cuando abres el cajón de apps.
* En los Pixel, tienes la opción de cambiar el tema del celular.

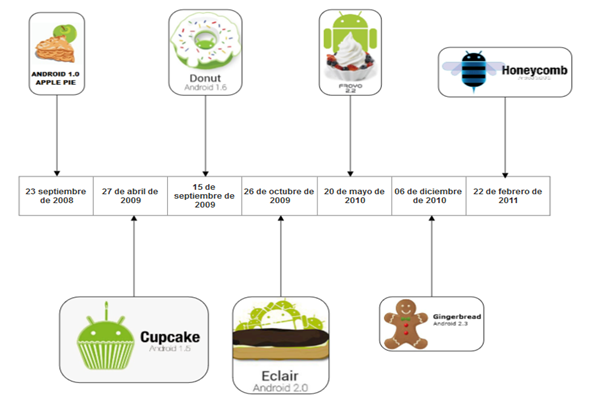
.

Figura 2.5. Línea de tiempo que muestra las fechas, nombres y logos de las versiones del sistema operativo Android.

* **Sistema operativo IOS**

iPhone OS (IOS) es un SO desarrollado por Apple, originalmente para su teléfono inteligente, pero al final decidieron implementarlo en todos sus productos, como el iPod Touch, iPad y Apple TV, la empresa no permite que este SO sea instalado en dispositivos de terceras compañías. En el año 2007 se dio a conocer la existencia de este SO para entonces aún no contaba con un nombre oficial, pero fue hasta el siguiente año donde se le empezó a llamar iPhone OS. Es importante mencionar que en la actualidad es el principal rival del SO Android, ya que entre ellos se reparten el mayor porcentaje del mercado de SO.

Características del sistema: En la pantalla principal es donde se encuentra la ubicación del icono de las aplicaciones y el dock en la parte inferior de la pantalla donde se pueden anclar aplicaciones de uso frecuente, precisamente un dock es un elemento gráfico que permite a los usuarios iniciar, cambiar y monitorear las aplicaciones que tienen instaladas en sus dispositivos.

Versiones de iOS:

* IOS 1:iPhone OS 1 se lanzó el 29 de junio de 2007 con un conjunto limitado de funciones y algunas aplicaciones como Correo, iPod, Calendario, Fotos, Reloj, Texto, Safari, Notas, YouTube, Calculadora, Mapas, Configuraciones, Cámara, Stocks, y teléfono Esta versión de iOS no tenía App Store o la aplicación iTunes Store, aplicaciones que actualmente permiten a los usuarios comprar y descargar otras aplicaciones.

* IOS 2: Esta versión de iOS se lanzó el 11 de julio de 2008 con una de las principales novedades, la "App Store". La tienda de aplicaciones tenía soporte para aplicaciones de terceros que daban acceso a los usuarios a las aplicaciones creadas por los desarrolladores. El correo electrónico y las páginas de la pantalla de inicio fueron una de las adiciones significativas al iPhone OS 2.0.

* IOS 3: No hubo cambios importantes en esta versión de iOS, pero tuvo muchas pequeñas mejoras críticas. IOS 3.0 lanzado el 17 de junio de 2009, tenía la capacidad de cortar, copiar y pegar. Apple mejoró la precisión de su GPS agregando una brújula magnética a la aplicación iPhone Map. El iPhone equipado con iOS 3.0 podría grabar videos, una función que no estaba disponible en sus predecesores. La aplicación del sistema de mensajes se renombró a Mensajes con una adición de soporte para mensajería multimedia. El 3 de abril de 2010 se lanzó una actualización incremental iOS 3.2 que permite cambiar los fondos de la pantalla de inicio y la pantalla de inicio en modo horizontal.

* IOS 4:La característica más importante introducida en iOS 4 fue la capacidad de video chat de Face-Time. La versión fue lanzada el 17 de junio de 2010. Apple también lanzó iBooks, fue la primera vez que se introdujo el soporte nativo para iBooks en dispositivos con iOS. La creación de carpetas en la pantalla de inicio para las aplicaciones fue proporcionada por iOS 4. IOS 4 también presentó Game Center.

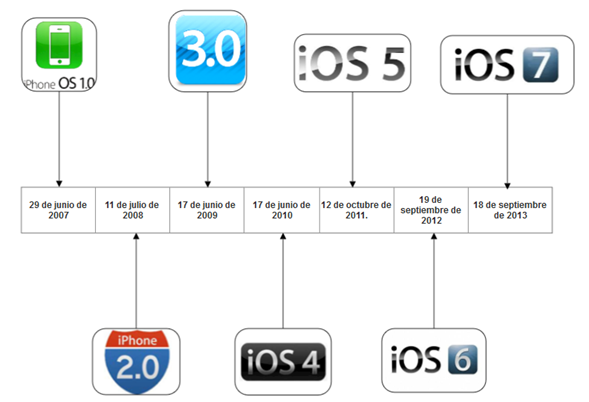
* IOS 5:Fue lanzado 12 de octubre de 2011. Esta versión de iOS podría considerarse la mayor actualización de iOS en comparación con las versiones anteriores. Se presentaron más de 200 nuevas funciones. iOS 5 albergó un nuevo centro de notificaciones y lanzó Messenger con iOS 5. La aplicación de la cámara se actualizó con la capacidad de editar fotos. Los dispositivos iOS se hicieron independientes de la PC al introducir la sincronización inalámbrica. La integración de Twitter para compartir contenido fue otra característica presentada con esta versión.

* IOS 6:Fue lanzado 19 de septiembre de 2012. Con esta versión de iOS, Apple creó su propia solución de Maps que proporciona navegación paso a paso y modelos tridimensionales en ciertas ciudades. El iOS ya no dependía de Google Maps, pero el lanzamiento de los mapas de Apple fracasó y resultó en el despido del jefe de iOS, Scott Forstall. La App Store obtuvo un importante rediseño que también se reflejó en iTune Store e iBookstore. La integración de Facebook se introdujo con iOS 6. La aplicación del teléfono se actualizó con un nuevo teclado y se presentó la posibilidad de responder a las llamadas entrantes a través de mensajes de texto.

* IOS 7:Lanzamiento en 18 de septiembre de 2013. El iOS 7 tuvo el mayor rediseño visual desde el lanzamiento de iOS en 2007. La interfaz de usuario de iOS 7 tiene degradados coloridos, gráficos llamativos y paneles transparentes deslizantes. Cada elemento fue rediseñado para dar una apariencia y sensación simples. El Centro de control se lanzó con la capacidad de alternar entre WiFi, Bluetooth, modo de avión y otras configuraciones. También permite iniciar rápidamente la linterna, la calculadora o la cámara. iTunes Radio fue otra característica importante introducida en iOS 7. Apple también agregó AirDrop para compartir imágenes, indicaciones de manejo, video, entre otros.
* IOS 8: Fue lanzado 2 de junio de 2012. El iOS 8 introdujo Family Sharing, que permite a los usuarios de Apple compartir todo el contenido que han descargado de iTunes. El contenido se puede compartir con hasta seis miembros familiares designados. En la aplicación de mensajería incorporada, Apple agregó una función para enviar mensajes de audio y video. Una de las principales actualizaciones fue widgets que permiten a las aplicaciones compartir datos y funciones con otras aplicaciones. IOS 8 agregó iCloud Drive, que ofrece almacenamiento en la nub. IOS 8 también introdujo tipo predictivo en su teclado en pantalla que sugiere varias opciones para la próxima palabra.

* IOS 9: Fue lanzado el 16 de septiembre de 2015. En iOS 9, se introdujeron las extensiones de bloqueo de contenido para Safari, lo que les dio a los usuarios finales el control sobre qué contenido se está cargando y cómo se está rastreando. IOS 9 incluía una nueva aplicación de Noticias que mostraba noticias de fuentes como New York Times, CNN, Wired y ESPN. Los mapas de Apple incluyen indicaciones de tránsito para muchas ciudades de los EE. UU. Y varias ciudades de China. Maps también hizo recomendaciones a un punto de interés. iOS también agregó una serie de funciones al iPad para mejorar la productividad y proporcionar una multitarea mejorada.

* IOS 10: Fue presentado el 13 de septiembre de 2016. La pantalla de bloqueo en iOS 10 tiene un aspecto nuevo y fresco. La pantalla se iluminará cuando un usuario la levante a la cara. La notificación tiene un nuevo aspecto en iOS 10. El toque 3D funciona en los elementos presentes en el centro de control. En la última versión de iOS, los usuarios pueden enviar dibujos, latidos y otros toques con Digital Touch. Los mapas de Apple tienen un aspecto nuevo que ofrece búsquedas recientes de fácil acceso y lugares visitados con más frecuencia. En iOS 10, Apple Music tiene un rediseño importante con navegación más simple, fácil acceso al contenido descargado.
* IOS 11: Fue lanzado el 5 de junio de 2017. La nueva versión del sistema operativo iOS 11 se calificó como “un paso gigante para el iPhone y un monumental salto para el iPad”. Este nuevo sistema operativo brinda un rediseño completo del App Store, un funcionamiento más inteligente y proactivo para Siri, que es una aplicación que cumple funciones de asistente; también se agregaron mejoras para la cámara y fotos, así como tecnologías de realidad aumentada para permitir experiencias más envolventes. Es también, la versión con más contenido para iPad. el nuevo centro de control, teclado para uso con una sola mano, Apple Pay para realizar pagos entre amigos, optimizaciones en la cámara y fotos, nuevo diseño para invertir los colores, función para grabar la pantalla, entre otras actualizaciones.



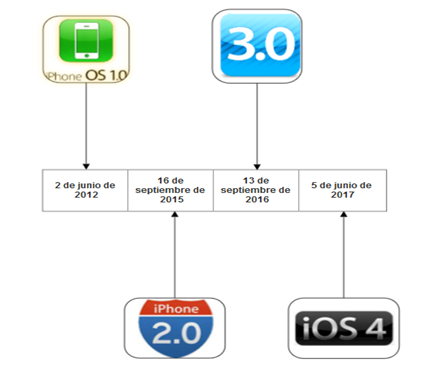


Figura 2.6. Línea de tiempo que muestra las fechas, nombres y logos de las versiones del sistema operativo IOS.

* **Sistema operativo Windows Phone**

El lanzamiento al mercado de este SO se dio el 21 de octubre de 2010 para los países europeos y 8 de noviembre del mismo año para los Estados Unidos. Se desarrolló desde cero con la intención de mejorar problemas que había presentado su antecesor Windows Mobile, por lo que se puede decir que se creó para sustituir al Windows Mobile.

En la pantalla actual nos muestra la fecha, y la información general del dueño, como citas, mensajes, correos, en la parte inferior aparece una barra con dos botones, también incluye una barra de iconos para bluetooth.

Versiones de Windows Phone

1. Windows Phone 7: Este sistema operativo fue lanzado el 21 de octubre de 2010. Esta es la primera versión generada como Windows Phone, en esta versión el SO carecía de muchas características que se fueron adquiriendo con las actualizaciones, esta versión se generaron 4 actualizaciones en las cuales se fueron corrigiendo las pequeñas fallas o errores.

1. Windows Phone 8.0: Se lanzó al mercado el 29 de octubre de 2012. Windows Phone 8 es la segunda versión lanzada a finales del 2012 con nuevas características, como, nueva pantalla de inicio y de bloqueo, más personalizado, nuevo núcleo de Windows NT con nuevo procesador para soporte de varios núcleos, en esta versión se generaron 3 actualizaciones en las cuales corrigieron y agregaron nuevas características.

1. Windows Phone 8.1: Esta versión fue lanzada el 14 de abril de 2014. Windows Phone 8.1 es la versión más actualizada de este SO, fue lanzada a mediados del 2014 con características como mejoras en el centro de notificaciones, sensor de WI-FI, sensor de datos, sensor de batería y mejoras en la pantalla de inicio, esta versión también se le han realizado actualizaciones para mejora de apariencia y rendimiento.

**21 de octubre de 2010**

**29 de octubre de 2012**

**14 de abril de 2014**



Figura 2.7. Línea de tiempo que muestra las fechas, nombres y logos de las versiones del sistema operativo Windows Phone.

* 1. **LENGUAJES DE DESARROLLO PARA APLICACIONES MÓVILES**

Antes de comenzar se debe tener claro que es un lenguaje de programación:

Un lenguaje de programación, es un conjunto de instrucciones, formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen una estructura y el significado de sus elementos y expresiones. Con el fin que las instrucciones puedan ser ejecutadas por un equipo para poder resolver un problema específico.

Hoy en día, existen diversidad de herramientas, lenguajes y entornos que se pueden elegir para poder desarrollar una aplicación móvil. Elegir uno puede depender para que SO se realizará la aplicación. En este caso que se trabajará con dispositivos móviles Android se hablará de sus dos lenguajes nativos: JAVA y Kotlin.

* + 1. **JAVA para Android**

JAVA es un lenguaje de programación orientado a objetos, multiplataforma y bastante potente. Es un lenguaje bastante popular porque cuenta con muchas herramientas de desarrollo y lo mejor de todo de forma gratuita, ya que la mayor parte de su código es libre.

Para poder desarrollar o modificar aplicaciones para Android será necesario tener un Kit de Desarrollo de Software (SDK por sus siglas en inglés), el SDK de Android contiene diversas librerías, estas se dividen en:

* + Plataformas
  + Herramientas
  + Actualizaciones

Las Plataformas de SDK, permiten tener vistas previas de las versiones de Android (Nougat, Marshmallow, KitKat, Jelly Bean, etc.). Se recomienda tener al menos desde la Interfaz de Programación de aplicaciones (API por sus siglas en inglés) 14, Android 4.0 (Ice Cream Sandwich) hasta la API 28, Android P recientemente anunciado por Google, esto para poder visualizar como se verá la aplicación en cada versión del SO, de esta manera se puede especificar cuál será la versión mínima, para que la app pueda ser instalada de manera óptima.

Las Herramientas del SDK son librerías necesarias para facilitar el desarrollo de las aplicaciones, entre las herramientas que se pueden encontrar están:

* Google Web Driver
* Google Play Services
* Android Auto API Simulators
* GPU Debugging tools
* Soporte de Repositorios
* Entre otras

Actualizaciones del SDK, sirve para mantener todas las Plataformas y herramientas del SDK actualizadas o para agregar librerías extra que no están disponibles en las anteriores.

* + 1. **Kotlin**

Kotlin es un lenguaje que funciona sobre la Máquina Virtual Java (JVM, por sus siglas en inglés), y que también puede ser compilado a código fuente de JavaScript, es una alternativa bastante moderna a Java, que garantiza en un 100% la interoperabilidad con JAVA. Este intenta solucionar los problemas más comunes que tiene JAVA, siendo el principal, la protección frente a un null (Null Pointer Exception) ya que kotlin te obliga a tenerlo en cuenta. Una de las mayores características de este lenguaje es la asignación automática del tipo de dato a una variable o función en el caso que el programador no se lo asigne.

Este lenguaje fue desarrollado por un equipo de programadores de JetBrains en 2011, San Petersburgo y su nombre proviene de la Isla de Kotlin, cerca de dicha ciudad. En enero de 2012 fue nombrado Lenguaje del Mes de la revista Dr. Dobb.

El 17 de mayo del 2017, Google anunció que Kotlin sería oficialmente soportado como lenguaje para desarrollar aplicaciones en Android, gracias a esto, Google y JetBrains trabajaron para crear una perfecta integración con Kotlin y no se use como un plugin, siendo este ahora un lenguaje de primer nivel que le permite invocar objetos que forman parte de la SDK de Android.

**Entornos de desarrollo integrado para desarrollo de aplicaciones móviles en Android**

Para poder desarrollar las aplicaciones cada Sistema Operativo cuenta con Entornos de desarrollo integrado (IDE por sus siglas en inglés). Estos consisten básicamente en un editor de código, que tal y como su nombre lo indica permite al programador crear y modificar el código de sus proyectos; un compilador que se encarga de traductor el código fuente a lenguaje de máquina para que puede ser entendido por el equipo o dispositivo que se encargará de ejecutarlo; un depurador que permite ejecutar el programa paso a paso o instrucción por instrucción para poder lo que está pasando en cada una de ellas y así verificar errores o que el programa se ejecute de la manera deseada; y un constructor de interfaz gráfica. Todos estos que han sido empaquetados en una sola aplicación, este provee una interfaz de trabajo amigable para el programador, en la mayoría de los casos un IDE puede funcionar como un sistema en tiempo de ejecución y suelen trabajar con varios lenguajes de programación, estos pueden venir integrados o pueden ser agregados mediante el uso de plugins. Estos entornos de desarrollo están diseñados para maximizar la productividad de los programadores ya que proporcionan los componentes necesarios y una interfaz gráfica que se caracterizan por ser interactivas y fáciles de utilizar.

**Android Studio**

Este se puede decir que es por excelencia el IDE más óptimo para el desarrollo de aplicaciones para Android ya que ha sido desarrollado por el propio Android, este está basado en Android IntelliJ IDEA que analiza el código, buscando conexiones en todos los archivos del proyecto y lenguajes, con esta información, proporciona asistencia de codificación y profundidad, navegación rápida, análisis inteligente de errores.

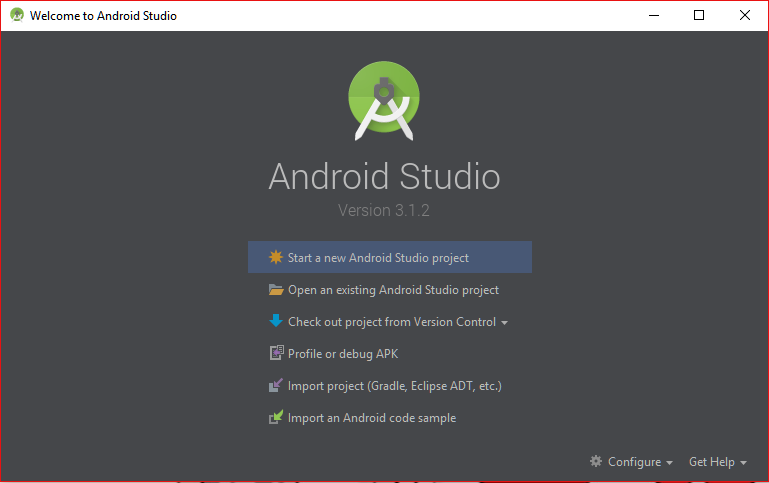


Figura 2.8. Pantalla inicial de Android Studio.

Además del potente editor de códigos y las herramientas de IntelliJ, Android Studio también ofrece más funciones que aumentan la productividad durante la compilación de las apps, tiene un entorno unificado en que se puede realizar desarrollos para todos los dispositivos Android, cuenta con Instant Run para aplicar cambios mientras la app se ejecuta sin necesidad de compilar un nuevo Paquete de Aplicaciones de Android (APK).

Android Studio trae integrado plantillas de código y Github, para poder compilar funciones comunes de las apps e importar ejemplos de código. Herramientas Lint que detectan problemas de rendimiento, usabilidad, compatibilidad de versiones, entre otros.

**Estructura del proyecto**

Cada proyecto contiene uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen módulos de apps para Android, módulos de bibliotecas y módulos de Google App Engine.

De manera predeterminada, Android Studio muestra todos los archivos del proyecto en la vista de proyectos de Android, como se muestra en la siguiente imagen:

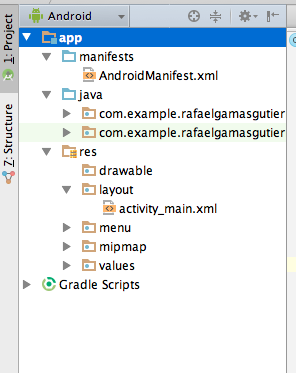


Figura 2.9. Presentación de todos los archivos de un proyecto de Android Studio.

* **manifests:** en esta carpeta se encuentra el archivo AndroidManifest.xml, este describe el proyecto, su nombre, paquetes, icono, los permisos que la aplicación requerirá, versión mínima de Android para poder ejecutarla, versión de la aplicación, entre otras características.
* **java:** contiene el código fuente de la aplicación.
* **res:** contiene los recursos necesarios para la aplicación, como imágenes, animaciones, layouts, iconos, entre otros.
* **Gradle Scripts:** contiene ficheros. gradle, estos permiten compilar y construir la aplicación.

Esta vista se organiza en módulos para proporcionar un rápido acceso a los archivos de origen clave del proyecto.

**Interfaz de usuario**

La ventana principal de Android Studio consta de varias áreas lógicas que se identifican en la siguiente imagen:

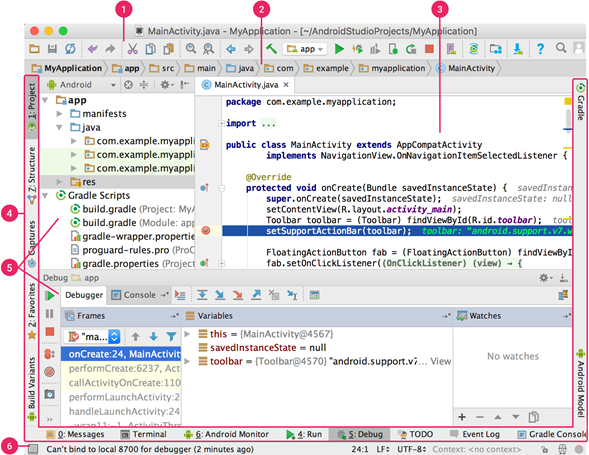


Figura 2.10. Ventana principal de Android Studio

1. La **barra de herramientas** permite realizar muchas acciones, como abrir o crear un proyecto, la ejecución de la app, guardar los cambios realizados.
2. La **barra de navegación** ayuda a explorar el proyecto y abrir archivos para poderlos editar. Proporciona una vista más compacta de la estructura del proyecto que la que se muestra en la ventana Project.
3. La **ventana del editor** es el área donde se puede crear y modificar código. Según el tipo de archivo actual, el editor puede cambiar. Por ejemplo, cuando se visualiza un archivo de diseño, el editor muestra el editor de diseño.
4. La **barra de la ventana de herramientas** se extiende alrededor de la parte externa de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
5. Las **ventanas de herramientas** permiten acceder a tareas específicas, como la administración de proyectos, las búsquedas, los controles de versión, el comportamiento de la aplicación cuando se ejecuta el modo debug, entre otros. Se puede expandir y contraer.
6. En la **barra de estado**, se muestra el estado del proyecto y del IDE en sí, como también cualquier advertencia o mensaje.

Se puede organizar la ventana principal para tener más espacio en pantalla ocultando o desplazando barras y ventanas de herramientas. También se puede usar combinaciones de teclas para acceder a la mayoría de las funciones del IDE.

* 1. **BASES DE DATOS**

Actualmente en el mundo, día a día, minuto a minuto, las personas manejan un gran volumen de datos, de todo tipo, estos datos muy a menudo, sino es que siempre, necesitan estar correctamente organizados, ordenados y almacenados en algún tipo de registro, para luego poder ser usados o analizados por cualquier entidad.

Para esto se crearon grandes almacenes de datos conocidos como base de datos.

Se entiende por base de datos a una colección de información correctamente organizada, dicha información puede ser fácilmente accesible, así como administrada y actualizada.

Podemos entender una base de datos como un gran almacén o librería de datos, estos datos son congruentes entre sí, pertenecientes a un mismo contexto.

Una base de datos almacena información real que puede ser administrada en cualquier herramienta informática.

La información dentro de una base de datos está organizada en filas, columnas y tablas, esto hace más fácil la búsqueda de información relevante dentro de la misma.

El principio fundamental de una base de datos se garantiza mediante ACID:

* Atomicidad: Asegura que una operación se haya o no realizado.
* Consistencia: Asegura que solo se empiece aquello que se puede acabar.
* Aislamiento: Una operación no puede afectar a otras.
* Escalabilidad: Una vez realizada la operación, ésta persistirá.

Las bases de datos normalmente son usadas para tipos específicos de registros de información, tales como:

* Cualquier tipo de transacciones.
* Catálogo de productos, inventarios.
* Información de personas.
* Entre otros.

Llevar un registro conciso de los tipos de información anteriormente mencionados sería imposible sin una base de datos.

Para acceder a la información contenida en una base de datos, se usan herramientas informáticas conocidas como Sistemas de Administración de Bases de Datos (DBMS por sus siglas en inglés), estas herramientas son las que nos permiten un total control sobre la base de datos, así como acceso a reportes y análisis de la información contenida en la misma.

La administración, tanto como la comunicación hacia una base de datos no se hace directamente a la misma, sino a través de un DBMS, para cualquier tipo de usuarios.

En el siguiente diagrama se muestra un ejemplo de los diferentes tipos de usuarios que desean acceder a una base de datos, todos se comunican a la base de datos mediante un DBMS.

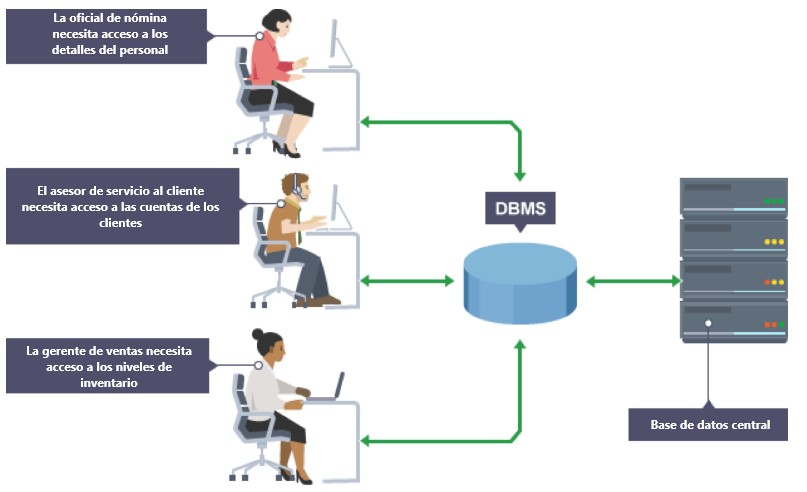


Figura 2.11. Ejemplo de los diferentes tipos de usuario que acceden a una base de datos.

* + 1. **Sistema de administración de base de datos**

Un Sistema de administración de base de datos es un programa informático que controla la organización, almacenamiento, seguridad (aunque esta también depende de la administración), así como la integridad de los datos en una base de datos, el DBMS también permite acciones conocidas como crear, leer, actualizar y borrar (CRUD) registros en una base de datos.

Un DBMS permite presentar la información requerida al momento de consultar una base de datos en formatos variados, dependiendo de la necesidad, esto nos facilita la elaboración de reportes al consultar una base de datos.

Tareas como la recuperación de datos tras un fallo de cualquier tipo también son permitidas por un DBMS, así como elaborar copias de seguridad de la información contenida dentro de la base de datos, o cifrar datos,cabe recordar que muchas veces la información contenida en una base de datos contienen datos que no pueden ser públicos (información de personas, contraseñas, entre otros).

Existen varios DBMS, dependiendo del tipo de lenguaje de la base de datos, algunos de los fabricantes más conocidos de DBMS son, Oracle, Microsoft, entre otros.

**Cómo se comunica un DBMS con la base de datos**

Generalmente, los DBMS acceden a la información contenida dentro de la base de datos mediante lenguajes de base de datos (database language en inglés), estos son lenguajes de alto nivel; los lenguajes de alto nivel se caracterizan por ser más agradables al usuario, así como expresar su algoritmo de una manera más entendible a la capacidad humana, que a la máquina o computador en sí; los lenguajes de alto nivel simplifican la tarea al momento de construir aplicaciones que involucren una base de datos.

El lenguaje de base de datos se refiere también al lenguaje de creación de nuestra base de datos, al mismo tiempo permiten cumplir con diferentes tipos de tareas, entre las categorías destacadas de tipos de lenguajes de base de datos están:

Lenguaje de control de datos (DCL): Permite controlar los diferentes tipos de accesos a la información contenida en la base de datos, pudiendo así fortalecer la seguridad de la misma.

Lenguaje de definición de datos (DDL): Permite estructuras de datos, como tablas, para ordenar la información, al mismo tiempo permite crear o eliminar relaciones entre tablas dentro de la base de datos.

Lenguaje de manipulación de datos (DML): Esta categoría de lenguaje de base de datos permite las acciones CRUD dentro de la base de datos, con la información contenida en la misma.

Lenguaje de consulta de datos (DQL): Facilita hacer búsquedas directas de información proveniente de nuestra base de datos.

* + 1. **Tipos de base de datos**
* Base de datos no relacionales: En informática, este tipo de base de datos también es llamada NoSQL. Lenguaje de consulta estructurado (SQL) es el lenguaje principal para consultas dentro de una base de datos a través del DBMS, por lo que al referirse al término NoSQL, esto quiere decir efectivamente que este tipo de base de datos no usan el lenguaje SQL para administrar su base de datos.

Los datos almacenados dentro de este tipo de bases no requieren estar estructurados mediante tablas, de ahí el término “no relacionales”, al mismo tiempo, este tipo de base de datos no garantiza el acrónimo ACID visto anteriormente, garantiza una base de datos concisa y bien estructurada.

* Base de datos relacionales: Una base de datos relacional se refiere a una base de datos digital, usando un modelo relacional, es decir, la información se organiza mediante una o más tablas (llamadas también relaciones), cada una con un identificador único uno para cada fila (o para cada registro).

Las filas dentro de las tablas de la base de datos son también llamadas registros, así como las columnas son conocidas como atributos. Generalmente, cada tabla representa un tipo de entidad, adecuada al mundo real, tales como entidad categoría de producto, o entidad producto, en la figura 2.12 se puede ver un ejemplo de cómo estas dos entidades mencionadas se relacionan en una base de datos.

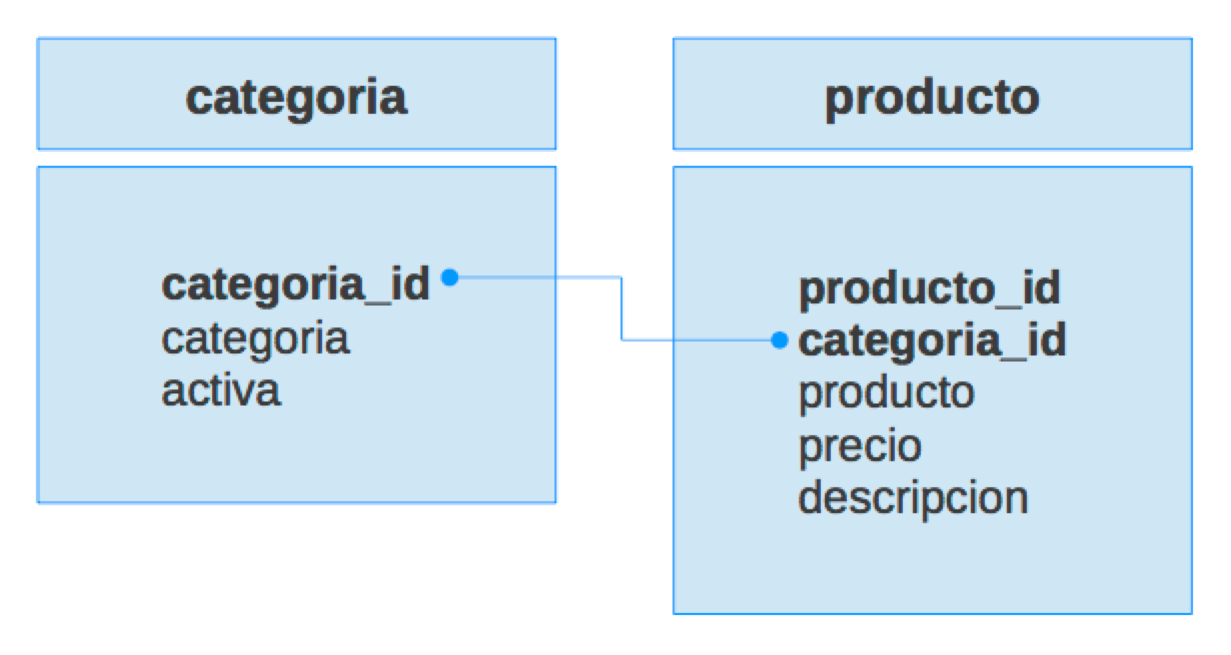


Figura 2.12. Representación de dos tablas en una base de datos.

Las filas representan instancias para este tipo de entidades; en el caso de la entidad categoría, un registro podría ser “Electrónicos”; para la entidad producto, un registro o fila podría ser “Computadora”, las columnas restantes dentro del registro pueden representar atributos asociados a este registro en particular.

Las bases de datos relacionales, al contrario de las no relacionales, si usan SQL como su lenguaje principal de consultas.

**SQL**

SQL es un lenguaje específico de dominio, el cual amplía o da función a las bases de datos relacionales. SQL es un lenguaje estándar e interactivo que brinda funcionalidad a nuestra base de datos, SQL utiliza o maneja características básicas de álgebra y cálculo (ya que está basado en las mismas), que nos permiten elaborar y efectuar consultas, con el fin de obtener información de forma sencilla, así como ingresar nueva información a la base de datos, actualizarla y eliminarla, lo que serían las opciones CRUD mencionadas anteriormente, así mismo, permite recuperar información en caso de pérdidas.

SQL es tanto un lenguaje de definición de datos como de manipulación y control de los mismos, al mismo tiempo es considerado el lenguaje universal para una base de datos.

SQL puede integrarse fácilmente a otros lenguajes de programación, tales como ASP, PHP o JAVA, así mismo, puede usarse con cualquier tipo de gestor de base de datos (DBMS).

**DBMS y SQL**

Los DBMS usan SQL como lenguaje principal para la administración de la base de datos.

Entre los DBMS con soporte para SQL más utilizados se encuentran:

**Oracle**

Oracle no es solo un DBMS para SQL (Oracle Database), sino que es el nombre de una compañía de nivel mundial, conocida como Oracle Corporation (abreviado únicamente Oracle), que se especializa en el desarrollo de soluciones informáticas, así como de soluciones en la nube y de escritorio.

**Microsoft SQL Server**

Microsoft SQL Server (MSQL) es un sistema de manejo de base de datos para el tipo de datos de base relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado por la línea de comandos de MSQL se denomina Transact-SQL (TSQL).

MSQL se compone de las siguientes características:

* Soporta transacciones.
* Incluye en entorno gráfico para la administración de la base de datos, de manera que facilita la misma.
* Soporta funciones, procedimientos almacenados y triggers.
* Permite trabajar de modo cliente-servidor, de manera que la información y datos se alojan en un servidor de base de datos, para que los clientes puedan acceder desde cualquier entorno.
* Permite hacer uso tanto del entorno gráfico como de la consola de administración para nuestras consultas.

**PostgreSQL**

PostgreSQL es un DBMS para bases de datos relacionales, una de las características principales del PostgreSQL es que es libre y de código abierto, publicado bajo una licencia publicada bajo su mismo nombre, esto quiere decir que PostgreSQL no es manejado por una empresa, sino por toda una comunidad de desarrolladores.

Características de PostgreSQL:

Permite que mientras un proceso se escribe en una o varias tablas, los usuarios puedan acceder a la misma sin bloqueo alguno.

Soporte nativamente para direcciones IP y MAC.

Permite a usuarios crear sus propios tipos de datos.

Así como beneficios PostgreSQL también presenta irregularidades, su defecto principal es que no cuenta con un gestor de defectos, por lo que hace muy difícil conocer el estado de los mismos.

Sin duda alguna, PostgreSQL es el DBMS más usado para ambientes que enfaticen el software libre.

**SQLite**

SQLite es una librería que implementa un DBMS para bases de datos transaccionales SQL, sin necesidad de servidores o configuración alguna para la misma.

SQLite es el tipo de base de datos más usada en el mundo, con grandes cantidades de aplicación corriendo su librería interna.

Este no cuenta con procesos dentro del servidor de aplicaciones, sino que ejecuta acciones de lectura y escritura directamente en el disk file (archivos de sistema que pueden guardar datos en nuestros dispositivos de almacenamiento).

SQLite es una librería compacta, aun así, actúa como un DBMS común y corriente, con todas las características de uno y con las mismas funcionalidades.

Existe una compensación de SQLite entre uso de memoria y velocidad, SQLite generalmente trabaja de manera más rápida entre más memoria se le conceda, aun así, la velocidad de procesamiento es considerablemente buena, aun en ambientes con baja memoria.

Las bases de datos SQLite son perfectas en dispositivos con bajas características en hardware, como sistemas operativos no tan exigentes, pero que aun así requieren cierta exigencia para ejecutar sus aplicaciones, por lo que, el caso ideal para usar una base de datos SQLite seria en aplicaciones para dispositivos móviles, dispositivos con bajas características si lo comparamos con un ordenador, pero que aun así necesitan tener un balance entre velocidad de procesamiento, batería y rendimiento a la hora de ejecutar una aplicación.

**SQLite y Bases de datos móviles**

Se entiende por base de datos móvil, a una base de datos que es total y completamente alojada dentro del dispositivo móvil, sin depender de ningún servidor; se pueden mencionar ejemplos como una lista de contactos, distancia recorrida, mensajes móviles, entre otros.

Muchos dispositivos móviles, quizá necesiten acceder a la información contenida en una base de datos o un repositorio, para luego operar con la misma y realizar transacciones incluso aunque ya no se encuentren en conexión con la fuente de información.

SQLite es considerado el lenguaje estándar para una base de datos relacional en un dispositivo móvil, todo esto debido a que ofrece:

* No necesita dependencias, está incluida en sistemas operativos Android o iOS.
* Desarrolladores de la base de datos pueden definir exactamente el esquema de base de datos que deseen, así mismo tienen total control sobre la misma.
* Acepta cualquier tipo de consultas o queries.
* Basado en SQL.
* Depuración de datos, así como análisis.

Existen otros tipos de base de datos para dispositivos móviles (como Oracle Lite, SQL Server Express, entre otros), pero SQLite es la DBMS más usada por desarrolladores para dispositivos móviles.

* 1. **SERVICIOS WEB**

El Consorcio Mundial de la Red (W3C) ha propuesto la definición de un Servicio Web (WebService) como un sistema de software designado que brinda soporte a la interacción de máquina a máquina, y permite el intercambio de información por medio una red. Un Servicio Web puede realizar una o muchas tareas, dicha tarea se describe mediante una notación XML (Lenguaje de Marcas Extensible) estándar conocida como Lenguaje de Descripción de Servicios Web (WSDL), el uso de esta descripción del servicio provee todos los detalles que se necesitan para la interacción con el Servicio Web, además que se incluye los formatos de mensaje; y estos a su vez detallan las operaciones que se realizarán, los protocolos de transporte y la ubicación.

La interfaz WSDL es muy útil, debido a que oculta los detalles de cómo está implementado el servicio; dicho servicio es utilizable de manera independiente de la plataforma de hardware o software en la que el servicio es implementado, además de ser independiente del lenguaje de programación en el que está escrito dicho servicio.

Los Servicios Web suelen ser muy útiles en el día a día, ya que por ejemplo cuando alguien realiza una búsqueda sencilla en internet, dicha consulta se hace a miles de servicios web que proveen miles de respuestas a la vez, otro ejemplo es cuando una persona quiere realizar un viaje, pero utiliza el internet como medio de búsqueda para encontrar sitios web que proporcionan los precios más accesibles, estos sitios web que proporcionan dicha información se comunican por medio del servicio web a distintos sitios que ofrecen servicios de viajes y sus costos, de esta manera el cliente; en este caso el usuario, encuentra los precios más accesibles por medio de una sencilla búsqueda en internet.

Existen diversas maneras para realizar intercambios de datos en los servicios web, una de las más importantes es la Transferencia de Estado Representacional (REST). Después de su lanzamiento en el año 2000 se abrió paso como un nuevo enfoque que permite el desarrollo de proyectos y servicios web; este fue definido por Roy Fielding. Dicho protocolo permite el intercambio y la manipulación de datos en los servicios de internet. En la actualidad, es el protocolo más utilizado para el desarrollo de servicios de aplicaciones. REST es cualquier interfaz entre sistemas que utilice el Protocolo de transferencia de hipertextos (HTTP) para la obtención o generación de operaciones de datos en cualquier formato posible como, por ejemplo: XML y JSON; estos formatos sirven para la representación de información estructurada en la web en forma de documentos.

REST se ha convertido en el estándar más lógico, eficiente y habitual en la creación de API para servicios web, además de ser una alternativa a otros protocolos estándar de intercambio de datos, ya que permite una solución más sencilla a la hora de manipular datos. Esta manipulación de datos se hace mediante distintas operaciones, las más importantes que relacionan REST y HTTP son:

1. Creación de datos (POST).
2. Lectura y consulta de datos (GET).
3. Edición de datos (PUT).
4. Eliminación de datos (DELETE).

Como se mencionó anteriormente, existen formatos que sirven para la representación de información estructurada en REST, como es el caso de la Notación de Objetos de JavaScript (JSON)**,** este permite el intercambio de datos en un formato muy ligero. La lectura y escritura de este formato es muy simple para las personas a la vez que es muy fácil de ser interpretado y generado por las computadoras. Este es un formato de texto que es totalmente independiente de cualquier lenguaje de programación, no obstante, utiliza algunas convenciones que son usadas en la familia de lenguajes de programación C, que incluye el mismo C, C++, C#, Java, JavaScript, entre otros. Dichas propiedades hacen propicio el intercambio de datos en JSON. JSON se constituye de dos estructuras:

1. Una colección de pares nombre/valor, en múltiples lenguajes esta colección de pares es conocido como un objeto, estructura, estructura y de otras maneras.
2. Una lista de valores ordenados. En múltiples lenguajes, esta lista se conoce como arreglos, listas, vectores.

Estas son básicamente estructuras universales, ya que en teoría la mayoría sino es que todos los lenguajes de programación las deberían de soportar de una forma u otra. Es muy razonable que al momento de intercambiar datos de manera independiente al lenguaje de programación se utilice dicha notación.

JavaServer Faces (JSF), es una infraestructura de interfaz de usuario o API, permite que el desarrollo de aplicaciones web basadas en el lenguaje de programación Java sea más fácil. JSF facilita el desarrollo de Interfaces gráficas de usuario (GUI) en aplicaciones de la edición de Empresarial de Java. Dicha tecnología surgió como solución al problema de la presentación y el comportamiento de una aplicación Web, por lo que permite que haya una separación de las actividades realizadas por los autores de la página con las actividades realizadas por los desarrolladores de la lógica de esta.

Así como existen entornos de desarrollo de aplicaciones de móviles, existen entornos especializados en el desarrollo de aplicaciones web; que proporcionan herramientas para la elaboración de proyectos web. Uno de estos entornos de desarrollo y de los más utilizados en la creación de proyectos web es Eclipse, su plataforma de desarrollo está estructurada como subsistemas (sistemas más pequeños) que se implementan en uno o más complementos, dichos subsistemas se construyen sobre un motor de ejecución en lapsos muy cortos de tiempo.

**Workbench de Eclipse**

El término “Workbench” se refiere al entorno de desarrollo de escritorio; es decir en una computadora cualquiera. El Workbench de Eclipse tiene como objetivo integrar de una manera perfecta las herramientas y una apertura controlada al proporcionar un paradigma común para la creación, administración y navegación de los recursos del Workbench. Cada ventana del Workbench contiene una o más perspectivas. Las perspectivas contienen vistas y editores, además controlan lo que se muestra en ciertos menús y barras de herramientas. Puede existir más de una ventana de Workbench en el escritorio en cualquier momento.

Cuando el Workbench es lanzado, lo primero que se observa es un pequeño cuadro de diálogo que permite seleccionar el área donde debe ubicarse el trabajo. El espacio de trabajo es el directorio donde se ubicará todo el trabajo que se desarrolle. Después de escoger el directorio donde se albergará el espacio de trabajo, se mostrará solo una ventana del Workbench, como se mencionó anteriormente se pueden abrir varias ventanas del mismo al mismo tiempo. En la figura 2.18 se muestra el entorno de desarrollo de Eclipse ya habiendo escogido la ubicación del espacio de trabajo.

Es muy importante conocer los diversos elementos con los que cuenta un Workbench, como se mencionó anteriormente, algunos de ellos consisten en: perspectivas, vistas y editores. Una perspectiva es un grupo de vistas y editores en la ventana del Workbench, en una sola ventana del Workbench pueden existir una o varias perspectivas.

En una ventana cada perspectiva puede tener un conjunto diferente de vistas, pero todas las perspectivas comparten el mismo conjunto de editores. Una vista es un componente visual dentro del Workbench, por lo general es utilizada para navegar por una lista o jerarquía de información como por ejemplo los recursos del Workbench, inclusive mostrar las propiedades del editor que está activo. Las modificaciones realizadas en una vista son guardadas de forma inmediata. Al igual que una vista, un editor es un componente visual dentro del Workbench, normalmente se utiliza para editar o explorar un recurso, la representación de este es texto o un diagrama de manera visual. Las modificaciones hechas en un editor se apegan al cierto modelo parecido a un ciclo de vida, es decir abrir-guardar-cerrar.

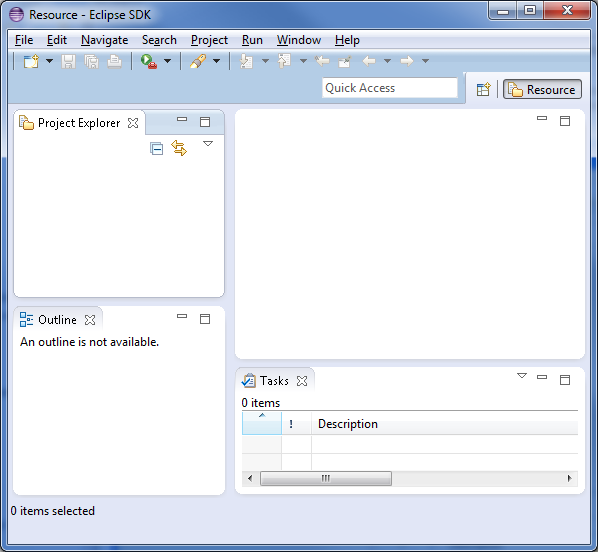


Figura 2.13. Workbench de Eclipse IDE.

**2. 6 INVENTARIO**

Debido a que el proyecto se enfoca en la creación de un módulo que permitirá la gestión de medicamentos, que básicamente es un inventario, a continuación, se detallará un poco más acerca de este tema.

Inventario son registros documentados de cualquier tipo de objetos que una empresa o entidad tiene en su pertenecía, este facilita el poder encontrar cada objeto de manera, más rápida y sencilla y saber que objetos están disponibles y en muchos casos saber la ubicación especifica donde se encuentra.

Se debe destacar que los inventarios se llevan a cabo en lugares que disponen de muchas cantidades de objetos pertenecientes, y esto es así porque es una herramienta muy eficiente a la hora que se quiere organizar muchas cosas, verificar y saber cuándo entran, cuándo salen, los costos que implicaron sus ingresos y egresos, la cantidad de disponibilidad de un mismo objeto, en algunos casos fecha de vencimiento, entre otras cuestiones.

El inventario nos provee de orden y de organización, además de permitirnos saber lo que hay disponible en un lugar específico.

También se puede decir por lo antes mencionado, que, “un inventario es la comprobación y recuento, tanto cualitativo como cuantitativo de las existencias físicas con las teóricas que fueron oportunamente documentadas”.

**2.6.1 Clases de inventarios**

Existe una numerosa cantidad de tipos de inventarios, entre los más recurrentes se cuentan: inventarios finales (se lleva a cabo cada vez que se cierra el período fiscal, generalmente el 31 de diciembre), inventarios periódicos (se realiza cada determinado tiempo), inventarios iniciales (se registran todos los bienes de la empresa), inventarios de liquidación legal, inventario de materias primas, inventario de seguridad, inventario de gestión, inventario físico, entre otros.

Entre las razones que existen en las empresas para realizar un inventario podemos citar: para reducir costos de adquisición, para reducir costos de calidad por arranque, para reducir costos vinculados al material faltante y para reducir costos de pedidos.

Aunque, también existen otras tantas razones que convierten al inventario en una cuestión engorrosa, compleja para la empresa en cuestión como ser: costos por almacenaje, dificultades para responder a los clientes, los costes de coordinar su producción, los costos de los productos defectuosos al tratarse de lotes grandes y los costos asociados a la reducción de capacidad.

La incorporación de la tecnología ha hecho que esta tarea de inventariar empresas, u otros espacios, sea mucho más sencilla, ya que existen softwares especialmente desarrollados para tales menesteres, como ser las bases de datos, que facilitan enormemente esta tarea, que anteriormente era más engorrosa ya que debía hacerse por escrito y en libros especiales.

**CAPÍTULO 3. REQUERIMIENTOS DEL SISTEMA**

En este capítulo se detalla el análisis de la información de los requerimientos del módulo de control de inventario de ambulancia, así como el de administración de inventario para ambulancias, su funcionamiento, procesos y variables relacionadas con el mismo, a través de la presentación del flujo general del módulo de la aplicación y su integración con la App SAPHCA, los casos de uso para definir los requerimientos funcionales de cada una de las acciones del módulo y los atributos de calidad para su creación.

A continuación, se presentan los flujos de las aplicaciones SAPHCA y el Sistema web de control de inventario de ambulancia (SICIA) para la administración de inventario de ambulancias dentro de SAPHCA.

**3.1 FLUJO GENERAL DEL MÓDULO EN LA APLICACIÓN SAPHCA**



Figura 3.1. Flujo general del módulo de control de inventario de ambulancias en la aplicación SAPHCA.

En la figura 3.1 se presenta un diagrama esquemático del flujo general del módulo de control de inventario de ambulancias, dentro de la app SAPHCA, las direcciones de las fechas representan la trayectoria del flujo en el diagrama.

A continuación, se muestra la descripción del módulo, según la figura 3.1:

**1.0 Pantalla de inicio sesión (SAPHCA):** Es la pantalla de inicio de la app móvil, si y sólo si es la primera vez que se inicia sesión, caso contrario la sesión queda guardada dentro de la app móvil, y no es necesario volver a iniciar sesión, a menos que esta se haya cerrado al salir de la aplicación.

**1.1 Menú principal de app SAPHCA:** Presenta los módulos accesibles dentro de la app. Actualmente la app cuenta con 4 módulos, más el módulo de control de inventario de ambulancias.

La pantalla de inicio de sesión, el menú principal y sus 4 módulos descritos anteriormente, ya se encuentran desarrollados por parte de MINSAL, sin embargo, son la base o el requisito para acceder al módulo de inventario a desarrollar en este proyecto.

**1.1.1 Módulo de control de inventario de ambulancias:** Al ingresar a este módulo se muestra en una tabla el detalle del inventario correspondiente a la ambulancia, además se muestran dos opciones, que corresponden a registro de consumo de medicamentos y a ajuste de inventario, todo esto siempre y cuando el dispositivo tenga asignado un inventario.

**1.1.1.1 Ajustes de inventario para ambulancia:** Está opción permite hacer ajustes manuales en inventario en caso de una mala manipulación o vencimiento de medicamentos, esto se hará seleccionando el medicamento e ingresando la cantidad a ajustar, entendiendo como ajuste la eliminación del medicamento en la cantidad especificada, dentro del inventario correspondiente a la ambulancia.

**1.1.1.2 Ingreso de consumo de medicamentos:** En esta opción se registrará el consumo de los medicamentos utilizados en las emergencias médicas que se reporten; todo con el fin de mantener un inventario actualizado.

**3.2 FLUJO DEL SISTEMA DE CONTROL DE INVENTARIO DE AMBULANCIA**

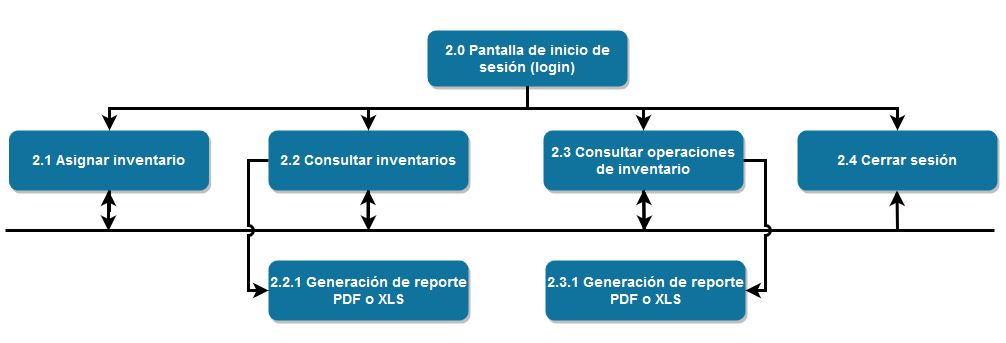
****

Figura 3.2. Diagrama del flujo del sistema SICIA.

En la figura 3.2 se presenta un diagrama esquemático del flujo general del sistema web para el control de inventario de ambulancias, así como pantallas para consultas y generar reportes, las direcciones de las fechas representan la trayectoria del flujo en el diagrama.

A continuación, se muestra la descripción del sistema web, según la figura 3.2:

**2.0 Pantalla de inicio se sesión:** Esta es la pantalla inicial de la aplicación web, el inicio de sesión consiste básicamente en el ingreso de las credenciales del usuario, posteriormente el sistema hace las validaciones correspondientes para dar acceso al sistema.

**2.1 Módulo de asignación de inventario:** Este módulo muestra en una tabla aquellas ambulancias que no tienen asignado un inventario, cada fila de la tabla muestra información de la ambulancia y se presenta la opción que permite la asignación del inventario.

**2.2 Módulo de consulta de inventario de ambulancias:** Esta opción permite visualizar el contenido de medicamentos del inventario actual de las ambulancias, pudiendo filtrar por las mismas. La información del inventario de la ambulancia, previamente seleccionada, se muestra en formato de tabla.

**2.2.1 Generación de reporte PDF o XLS:** Permite generar un archivo PDF o XLS que contiene la información del inventario que se muestra en pantalla.

**2.3 Módulo consulta de operaciones de inventario:** Este módulo permite visualizar las entradas, salidas, o bien las entradas y salidas a su vez de los medicamentos de la ambulancia que ha sido seleccionada; además de la cantidad, fecha en que se realizó dicha operación y algunos otros datos.

**2.3.1 Generación de reporte PDF o XLS:** Permite generar un archivo PDF O XLS que contiene la información generada a partir de las consultas realizadas en el módulo de consulta de operaciones de inventario.

**2.4 Cerrar sesión:** Esta opción permite finalizar la sesión iniciada previamente por un usuario del sistema de administración de inventario.

Aunque los sistemas se muestren separados, ambos sistemas están relacionados mediante la base de datos, ya que la aplicación SAPHCA(diagrama 3.1) realiza funciones de CRUD a los registros de la base, así como de consulta de datos, consultas que pueden resultar molestas para un tamaño de pantalla pequeño con las que cuentan los teléfonos móviles, es por esto, que se modelo y diseño la alternativa de desarrollar un aplicativo web (figura 3.2) para una extensión al módulo de consultas de SAPHCA, este aplicativo permite a usuarios del SEM, tanto operativos y administrativos, poder obtener información acerca del inventario actual y las operaciones de consumo de medicamento de las ambulancias, que han sido registradas desde la aplicación SAPHCA, así como generar archivos PDF o XLS según se requiera.

**3.3. REQUERIMIENTOS FUNCIONALES**

Los requerimientos funcionales especifican y detallan aquellas funciones que son necesarias que un sistema o aplicación sea capaz de llevar a cabo para cumplir con las necesidades del usuario. La forma más fácil de describir estas funciones son los casos de uso, ya que en ellos se describe quién será el usuario y que acciones debería realizar para interactuar con el sistema, para que este último sea capaz de dar respuesta a las necesidades del usuario.

En la tabla 3.1 se muestra y detalla la estructura a utilizar para describir los casos de uso.

Tabla 3.1 Estructura de casos de uso

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | Número identificador único del caso de uso. | | |
| **Nombre del caso de uso:** | Nombre asignado al caso de uso. Se suele designar la acción a realizar mediante un verbo en infinitivo. | | |
| **Creado por:** | Creador del caso de uso. | **Fecha de creación:** | Fecha en la que se creó el caso de uso. |
| **Participantes:** | Usuario del sistema o entidad externa al sistema que interactúa con él para cumplir un objetivo. | | |
| **Descripción:** | Breve descripción de la razón de ser del caso de uso. | | |
| **Pre-condiciones:** | Actividades que deban realizarse o condiciones que deban cumplirse para que el caso de uso suceda (pueden ser referencias a otros casos de uso). | | |
| **Post-condiciones:** | Cambios relevantes que puedan darse en el estado del sistema al finalizar el caso de uso. | | |
| **Flujo Principal:** | Provee una lista detallada de los pasos a realizar para llevar a cabo la tarea. Describe cada una de las acciones de los usuarios y las respuestas del sistema bajo condiciones normales y esperadas (caso de éxito). | | |
| **Flujo Alterno:** | Provee una lista de los pasos a seguir en caso de que se de alguna condición especial o no esperada. Debe especificarse el paso en que se dio la condición, y se detallan los pasos a realizar. | | |
| **Excepciones:** | Describe las condiciones de error previsibles que podrían ocurrir en la ejecución de la tarea. Para cada flujo se coloca el paso del flujo principal en el cual se dio el error. | | |
| **Supuestos:** | Lista cualquier supuesto que se haya hecho en la descripción o el flujo del caso de uso. | | |
| **Notas:** | Lista cualquier comentario adicional o algún aspecto que deba ser determinado. | | |

La columna del lado izquierdo corresponde a los elementos del caso de uso y la columna de la derecha presenta una breve descripción de los elementos. El primer caso de uso correspondiente al módulo es “Acceder al módulo de control de inventario”, dicho caso se observa en la tabla 3.2.

Tabla 3.2. Caso de uso “Acceder al módulo de control de inventario (SAPHCA)

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 1 | | |
| **Nombre del caso de uso:** | Acceder al módulo de control de inventario (SAPHCA) | | |
| **Creado por:** | César Mejía | **Fecha de creación:** | 17/06/2018 |
| **Participantes:** | Encargado de la ambulancia del SEM | | |
| **Descripción:** | En primer lugar, el acceso al módulo de control de inventario permitirá en al usuario visualizar el detalle del inventario de la ambulancia, además mostrará las opciones para ingresar al ajuste de inventario de ambulancia y registro de consumo de medicamento. | | |
| **Pre-condiciones:** | El usuario deberá haber iniciado sesión exitosamente para ingresar al módulo de control de inventario, además que la ambulancia debe tener un inventario asignado. | | |
| **Post-condiciones:** | El usuario podrá ver el detalle del inventario correspondiente a la ambulancia y las opciones del módulo de control de inventario. | | |
| **Flujo Principal:** | 1. Ingresar exitosamente a la aplicación SAPHCA. 2. Acceder al módulo de control de inventario. | | |
| **Flujo Alterno:** | No aplica. | | |
| **Excepciones:** | No aplica. | | |
| **Supuestos:** | No aplica. | | |
| **Notas:** | No aplica. | | |

En la tabla 3.3 se muestra la lista de todos los casos de uso del módulo de inventario y el sistema SICIA. La descripción completa del resto de casos de uso ver ANEXO B.

Tabla 3.3. Listado de casos de uso

|  |  |
| --- | --- |
| **Identificador del caso de uso** | **Nombre del caso de uso** |
| 1 | Acceder al módulo de control de inventario (SAPHCA) |
| 2 | Registrar el consumo de medicamentos (SAPHCA) |
| 3 | Realizar ajuste de inventario de ambulancia (SAPHCA) |
| 4 | Acceder al sistema web (SICIA) |
| 5 | Realizar asignación de inventario (SICIA) |
| 6 | Consultar inventario de ambulancias (SICIA) |
| 7 | Consultar operaciones de inventario (SICIA) |
| 8 | Generar reporte (SICIA) |
| 9 | Cerrar sesión (SICIA) |

**3.4. REQUERIMIENTOS NO FUNCIONALES**

Los requerimientos no funcionales o atributos de calidad son propiedades requeridas que surgen de toda la solución y no hacen referencia a acciones específicas dentro la aplicación. La especificación de estos requerimientos debe describir los atributos que el cliente requiere e incluir aquellos que son apropiados. Los atributos requeridos para el módulo de control de inventario de ambulancias y que cumplen con ser adecuados para este, se definen a continuación:

**Confiabilidad**

Atributos que intervienen en la función de la aplicación, manteniendo el nivel de rendimiento bajo requerimientos que se definieron inicialmente, así como el hecho de incluir documentación técnica, manual de usuario para dar mayor facilidad al uso de esta. El módulo de control de inventario de ambulancias debe contar con un mínimo de estas características y requisitos para lograr una garantía que dicho producto sea lo suficientemente confiable.

**Eficiencia**

Atributos que influyen entre el nivel de rendimiento que se tiene de la aplicación y la cantidad de recursos que se han utilizado, es por ello que la eficiencia del módulo control de inventario de ambulancias estará definida mediante el uso adecuado de los recursos, evitando saturar la memoria de los dispositivos móviles con uso multimedia de excesivo tamaño, optimizando tiempo de ejecución ayudando a tener una mejor experiencia del usuario/ aplicación.

**Escalabilidad**

El módulo debe ser diseñado de tal manera que permita que su desarrollo sea incremental, permitiendo la capacidad de inclusión de nuevas funcionalidades o modificación de las pertenecientes, sin la necesidad de ocasionar grandes cambios en el código existente y evitando comprometer su funcionamiento y calidad, para aumentar la capacidad de tamaño o de trabajo del sistema.

**Usabilidad**

El módulo deberá proporcionar al usuario un entorno visual agradable y amigable en el que se facilite su uso haciéndolo de forma natural e intuitiva. Facilitando un manual de usuario bien estructurado, mostrando mensajes de error pertinentes que informen y orienten al usuario, y apoyándose de elementos gráficos tales como iconos representativos, botones de fácil acceso, controles intuitivos.

**Mantenibilidad**

El módulo y cada uno de sus componentes deberán ser debida y ampliamente documentados en el código fuente con objetivo de hacer rápida su comprensión y facilitar posibles cambios que este requiera en el futuro, además de un manual técnico que permita obtener referencias claras.

**Arquitectura**

El módulo deberá asegurar compatibilidad con la plataforma Android, en su mayoría de versiones, orientado a dispositivos móviles en orientación horizontal. Debe utilizarse el lenguaje de programación Java orientado a dispositivos móviles; y en el caso de sistema web también debe utilizarse el lenguaje de programación Java orientado a la web.

**Interfaz**

La interfaz gráfica es la carta de presentación del módulo a desarrollar, esta contará con iconos intuitivos y grandes para evitar confusiones al momento de su utilización; además el tipo de letra será de un tamaño acorde al utilizado en la App SAPHCA, para mantener el estándar de la misma, en el caso de sistema SICIA se mantiene una interfaz gráfica excelente con iconos y letra los suficientemente grande para su visualización.

**Integración**

El módulo de control de inventario de ambulancias se deberá integrar adecuadamente a la App SAPHCA desarrollada por terceros para el MINSAL, para el éxito de la integración se añadirá todo el código fuente del módulo desarrollado del proyecto, al código fuente de la App SAPHCA desde las primeras etapas del desarrollo, con la reutilizando el código fuente existente y añadiendo de forma apropiada los métodos e interfaces (layouts). Se añadirá un botón desde el menú principal de la App SAPHCA para llamar al módulo de control de inventario de ambulancias.

**CAPÍTULO 4. DISEÑO DEL SISTEMA**

Ahora que el análisis se ha realizado y que los requerimientos del sistema han sido establecidos, se prosigue a definir la presentación de los datos, interfaces y demás componentes del sistema con el fin de establecer algunos requerimientos y otras características con poseerá el software; todo con el fin que este sea lo más amigable, visualmente atractivo y confiable para que el usuario no tenga ninguna dificultad para manipularlo. Con el diseño se pretende dar una idea al usuario de lo que será el módulo de control de inventario de ambulancias y el sistema web SICIA; una vez este sea implementado.

**4.1 DISEÑO DEL MÓDULO Y SISTEMA WEB SICIA**

El módulo de control de inventario de ambulancias se integrará a la aplicación móvil SAPHCA, que ya está siendo utilizada por parte del SEM pero no se le está sacando el máximo provecho posible; dicho módulo estará disponible en el menú principal de la aplicación, al ingresar al módulo se contará con 3 opciones disponibles: ajustar el inventario, ingresar consumo y consultar inventario. Por otra parte, el sistema web SICIA será muy agradable a la vista del usuario y además será muy simple de manejar; ya que solo contará con 5 pantallas que son: inicio de sesión, pantalla principal, asignar inventarios, consultar inventarios y consultar operaciones de inventario.

**4.2. BASE DE DATOS**

El desarrollo del módulo de administración de inventario ha sido implementado utilizando una base de datos (BD), esta es parte integral del funcionamiento de la APP y el sistema web. La base de datos ha sido facilitada por la DTIC del MINSAL y cumple con la estructura lógica del modelo relacional, cuenta con setenta y cuatro tablas; pero solo se hará uso de tres de ellas. A dicha base se le adicionará 3 tablas más para poder llevar el inventario y así cumplir con el propósito del proyecto.

**4.2.1. Tablas utilizadas por el Modulo de control inventario de el SAPHCA**

Del conjunto de tablas que componen la base de datos del SAPHCA, el módulo de control de inventario utilizará el listado que se muestra en la Tabla 4.1, en donde se describe el nombre de la tabla de la base de datos y la información que almacena.

Tabla 4.1. Listado de tablas que se utilizarán en el módulo

|  |  |
| --- | --- |
| **Tabla de la Base de Datos** | **Información que almacena** |
| ambulancia | Esta tabla contiene las diferentes ambulancias con las que cuenta el SEM para asignarlas a las demandas. |
| medicamento | Tabla donde se registran los medicamentos con los que cuenta el personal para suministrar dentro de las emergencias. |
| operacion | Tabla que contiene el registro de entradas o salidas del inventario; según sea el caso. |
| inventario | Tabla donde se registrará a que ambulancia pertenece cierto inventario. |
| detalle\_inventario | Tabla que contiene los detalles del inventario que posee cada ambulancia del SEM |

**4.2.2. Diccionario de Datos de las Tablas del Módulo de control de inventario.**

El diccionario de datos que se presenta a continuación está enfocado únicamente a mostrar el detalle de dos tablas de la base de datos. Estas fueron diseñadas exclusivamente para el manejo del inventario en cada ambulancia del SEM; en estas se manejan detalles como un identificador numérico único para cada ambulancia, producto e inventario, entre otros que son vitales para identificar el inventario de cada ambulancia. En las tablas 4.2 y 4.3 se presentan los nombres de columnas, tipo de datos, datos nulos y una descripción de cada campo en de las tablas.

Tabla 4.2. Detalle de los campos de la tabla “operacion”

|  |  |  |  |
| --- | --- | --- | --- |
| **operacion** | | | |
| Columna | Tipo de dato | Datos nulos | Descripción |
| id\_operacion | SERIAL | NO | Identificador del registro de una operación realizada. |
| id\_ambulancia | INTEGER | NO | Identificador que posee las diferentes ambulancias con las que contará SEM para asignarlas a las demandas. |
| id\_producto | INTEGER | NO | Identificador que posee que cada producto registrado en el inventario. |
| cantidad | INTEGER | NO | Cantidad del producto registrada en la operación. |
| fecha\_operacion | DATE | NO | Fecha en que se realiza la operación. |
| fec\_venc\_producto | DATE | NO | Fecha de caducidad de cada producto utilizado en la operación. |
| tipo\_operacion | CHAR | NO | Tipo de operación que se realizará identificada por un carácter único. |

Tabla 4.3. Detalles de los campos de la tabla “inventario”

|  |  |  |  |
| --- | --- | --- | --- |
| **inventario** | | | |
| Columna | Tipo de dato | Datos nulos | Descripción |
| id\_inventario | SERIAL | NO | Identificador de cada inventario. |
| id\_ambulancia | INTEGER | NO | Identificador que posee las diferentes ambulancias con las que contará SEM para asignarlas a las demandas. |

En el ANEXO C se encuentra la definición del resto de tablas a utilizar en esta investigación.

**4.2.3. Diagrama de la Base de Datos**

Describe la estructura de la base de datos en un diagrama que presenta todo el conjunto de tablas, los campos en cada una de ellas y las relaciones existentes entre sí. En la Figura 4.1 se muestra una parte del diagrama de la base de datos completa del SAPHCA, mostrando solo las tablas utilizadas por el módulo de control de inventario. El diagrama completo se encuentra en el ANEXO D.

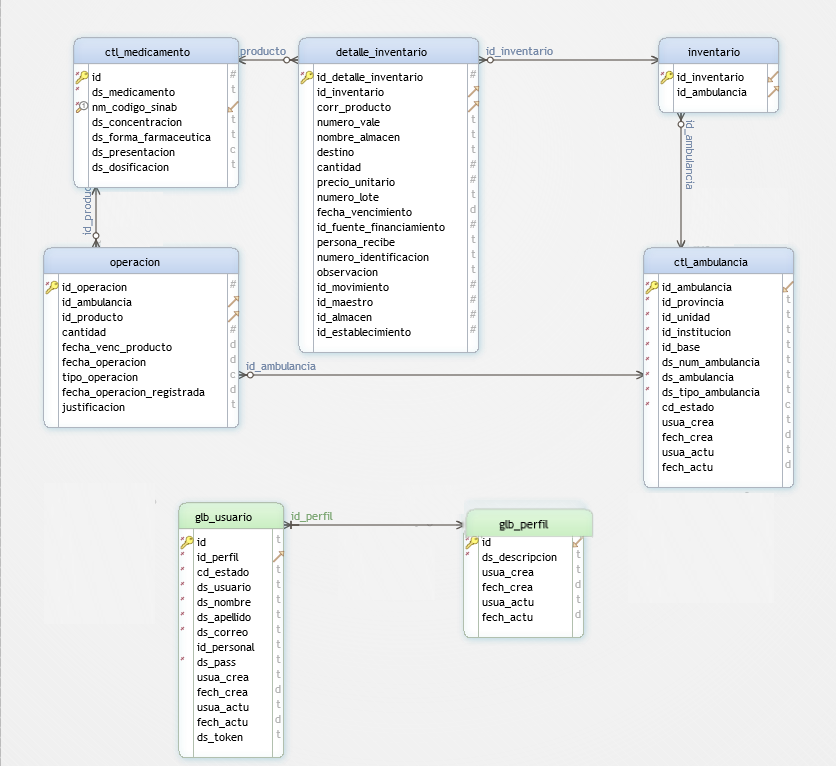


Figura 4.1. Parte del diagrama de la base de datos de SAPHCA.

**4.3 DISEÑO DE INTERFACES**

**4.3.1 Diseño de interfaces del módulo**

El diseño de interfaces tiene como objetivo hacer que la aplicación sea atractiva y fácil de usar para el usuario final, además, que la misma sea simple de usar e intuitiva, la aplicación SAPHCA ya cuenta con un estilo o tema predefinido, por lo que el módulo de extensión de inventario debe adecuarse a la misma tipografía, tamaño de letra, colores de vistas, entre otros, para que no exista una discrepancia al momento de usar dicho módulo.

En la figura 4.2 se presenta el diseño de la pantalla inicial del módulo de inventario, este contiene el inventario actual, así como las unidades disponibles y las fechas de vencimiento del mismo.

1. **Ambulancia**: Indica a qué ambulancia (por medio del código) pertenece el inventario visualizado, hay que recordar que cada ambulancia solo puede visualizar el contenido de su propio inventario.
2. **Tabla de inventario**: Contiene información acerca del inventario real de la ambulancia, la información se desglosa de la siguiente manera:
   1. Descripción: Contiene la descripción preliminar del medicamento.
   2. Unidad: Contiene las unidades disponibles de medicamento; en sus respectivas unidades de medida.
   3. Vencimiento: Contiene la fecha de vencimiento del medicamento; los medicamentos se agrupan por fechas de vencimiento, si se posee 2 o más versiones de un medicamento con diferentes fechas de vencimiento, se agruparán individualmente.
3. **Botón de ajustes de inventario**: Este botón permite dirigirse a la pantalla de ajuste de inventario, para realizar ajustes por alguna incidencia en el uso del medicamento, indicando el medicamento, cuántas unidades se descontarán de inventario y el motivo de la incidencia.
4. **Ingresar consumo:** Este botón permite dirigirse a la pantalla de ingreso de consumo de medicamentos; donde se seleccionará el medicamento y la cantidad que se ha utilizado en alguna emergencia, para así ser descontado del inventario de la ambulancia.

Como se puede apreciar se realizó un cambio de orientación de la pantalla; es decir de vertical (portrait) como se observa en la figura 4.2 a horizontal (landscape) como se observa en la figura 4.3, este cambio se debe a que en la aplicación Android existente SAPCHA, la vista por defecto para la aplicación y con la cual se trabaja con los dispositivos es en horizontal, por lo que las pantallas desarrolladas para la extensión de la aplicación, deben seguir el mismo patrón. Dicho cambio también fue realizado en las figuras 4.4, 4.5, 4.6 y 4.7.



Figura 4.2. Diseño de la pantalla inicial del módulo de inventario.

Figura 4.3. Pantalla inicial del módulo de inventario.

En la figura 4.4 se presenta el diseño de la pantalla de ajuste de inventario por incidencia, esta pantalla permite realizar ajustes por algún accidente en el cual se vio involucrado el medicamento; indicando el mismo, cuántas unidades se descontarán de inventario y el motivo de la incidencia.

1. **Selección de medicamento:** Permite seleccionar mediante una lista de medicamentos disponibles, el medicamento al cual se le realizará el ajuste (solo se muestran los medicamentos disponibles o actuales en el inventario dentro de la ambulancia).
2. **Cantidad:** Este campo de texto permite que el usuario introduzca la cantidad que se ajustará de acuerdo al medicamento seleccionado.
3. **Unidad de medida:** En el campo de texto de Unidad de medida se muestra al usuario la unidad de medida predefinida para ese medicamento, pudiendo ser el mL, L, unidad, paquete, entre otros, según este predefinido cuando se agregó el medicamento a la base de datos. El campo de texto de unidad de medida no puede ser seleccionado por el usuario, es rellenado automáticamente al seleccionar el medicamento, la unidad de medida sirve para que el usuario tenga una referencia al momento de ingresar la cantidad a ajustar.
4. **Justificación de incidencia:** Permite al usuario ingresar el detalle sobre la justificación del ajuste de inventario, esta justificación es requerida u obligatoria.
5. **Botón aceptar:** Al hacer clic en el botón aceptar se realizará el ajuste ingresado y procederá a descontar del inventario. Todos los campos del formulario mostrado en pantalla son obligatorios, por lo que se realizará una validación previa al hacer clic en el botón.
6. **Botón regresar:** Permite regresar a la pantalla de inventario de ambulancias.

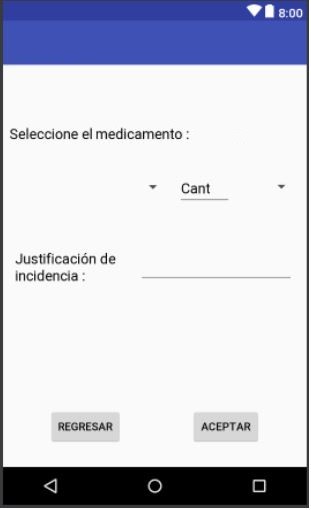


Figura 4.4. Diseño de la pantalla de ajuste de inventario por incidencia.

****

Figura 4.5. Pantalla de ajuste de inventario por incidencia.

En la figura 4.6 se presenta el diseño de la pantalla de ingreso de consumo de medicamentos, en dicha pantalla se descontará del inventario el medicamento y la cantidad que ha sido utilizada en alguna emergencia, además del detalle. Los detalles que son muy parecidos a los vistos en la pantalla de ajuste de inventario, con excepción de la justificación de incidencia que no se presenta en este caso. En caso que, al agregar un medicamento para darlo de baja, se puede ver el detalle del mismo y en algún caso hubo un error, el mismo se puede eliminar antes de sé que confirme el consumo del mismo.

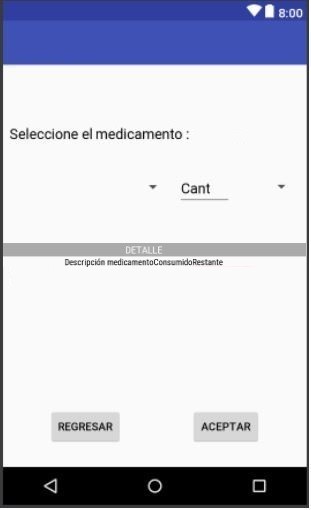


Figura 4.6. Diseño de la pantalla del ingreso de consumo.



Figura 4.7. Pantalla de ingreso de consumo de medicamentos.

**4.3.2 Diseño de interfaces sistema web SICIA**

En esta parte se presentará el diseño del sistema web SICIA, dicho diseño tiene como objetivo que el sistema sea atractivo visualmente; además de no ser nada complicado el manejo de este para los distintos usuarios del sistema, para esto se manejara gráficas y colores agradables para el usuario; acompañado con iconos referentes a las opciones que este manejara.

En la figura 4.9 se presenta el diseño de la pantalla de inicio de sesión del sistema web SICIA*.* Esta pantalla es muy simple y está compuesta por los siguientes elementos:

1. **Recuadro de inicio de sesión**: es el área en el que el usuario ingresará sus respectivas credenciales para ingresar al sistema web SICIA*.*
2. **Cuadro de usuario**: en este recuadro se ingresa el nombre de usuario mediante el teclado.
3. **Cuadro de contraseña**: en este recuadro el usuario ingresará su respectiva contraseña mediante el teclado.
4. **Botón ingresar**: una vez el usuario haya escrito sus credenciales este presionará el botón ingresar; y si estos son correctos tendrá acceso al sistema; caso contrario se le notificará que sus credenciales no son válidas.

****

Figura 4.8 Diseño de pantalla de inicio de sesión del sistema web SICIA.

****

Figura 4.9. Pantalla de inicio de sesión del sistema web SICIA*.*

Una vez se haya iniciado sesión, el usuario ingresará a la siguiente pantalla donde se presentarán los diferentes módulos disponibles. En la figura 4.11 se presenta el diseño de la pantalla principal del sistema web SICIA,que se compone de los siguientes elementos:

1. **Módulo de asignación de inventario**: En esta opción se mostrará la asignación de inventario para cada ambulancia que esté disponible.
2. **Módulo de consulta de inventarios**: Esta opción es visible sin importar el rol que esté asignado al usuario que ingreso al sistema; dicha opción permitirá consultar el inventario asignado a las diferentes ambulancias del SEM.
3. **Módulo consultar operaciones de inventario**: Esta opción permite consultar operaciones que se han realizado con el inventario.
4. **Cerrar sesión:** Esta opción finaliza la sesión del usuario.

****

Figura 4.10. Diseño de la pantalla principal del sistema web SICIA.



Figura 4.11. Pantalla principal del sistema web SICIA.

En la figura 4.13 se tiene a detalle la pantalla asignar inventario en la que es posible desplegar las diferentes ambulancias su respectiva base e institución y otras opciones, esta pantalla cuenta con los siguientes elementos:

1. **Seleccionador de registros:** Este permite al usuario seleccionar la cantidad de registros que desea ver en pantalla, dependiendo también de la cantidad de registros disponibles.
2. **Buscador:** Este buscador permite ingresar mediante teclado múltiples opciones y al ingresarse la búsqueda se realiza de manera automática.
3. **Resultados de ambulancias sin inventario asignado:** se presentan en forma de tabla que permite visualizar de una manera ordenada las ambulancias pendientes de asignación de inventario, cuenta con 4 columnas encabezadas por: identificador de ambulancia, base, institución y acción para realizar.
4. **Botón asignar inventario:** Al presionar dicho botón permite asignar inventario a cierta ambulancia.
5. **Botón siguiente/anterior:** Este botón permite avanzar a una siguiente página en caso que la información mostrada por la tabla sea demasiada; o regresar a la página anterior.

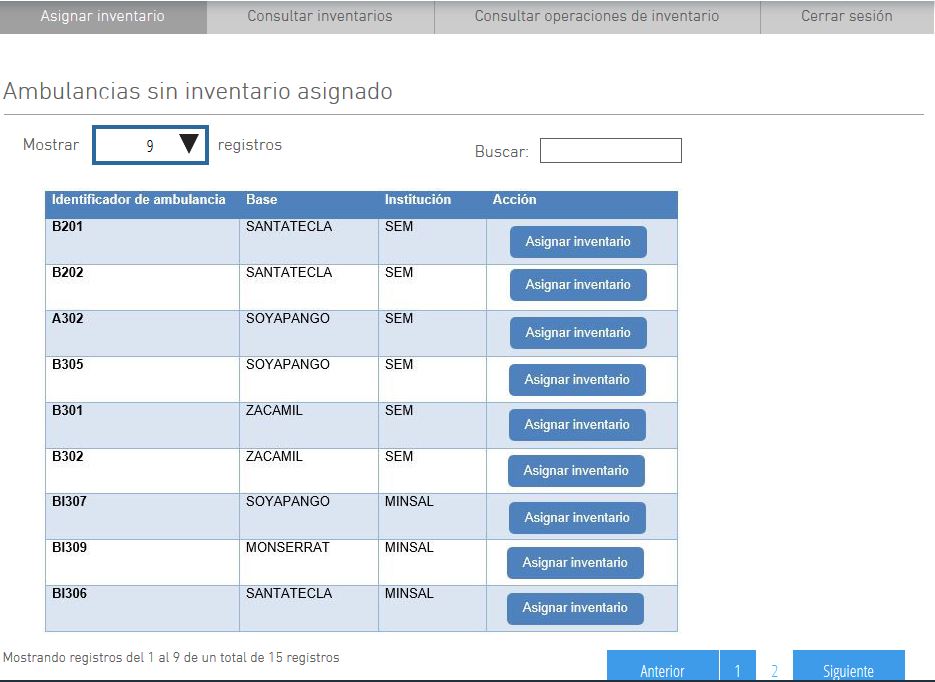


Figura 4.12. Diseño de la pantalla asignar inventario.

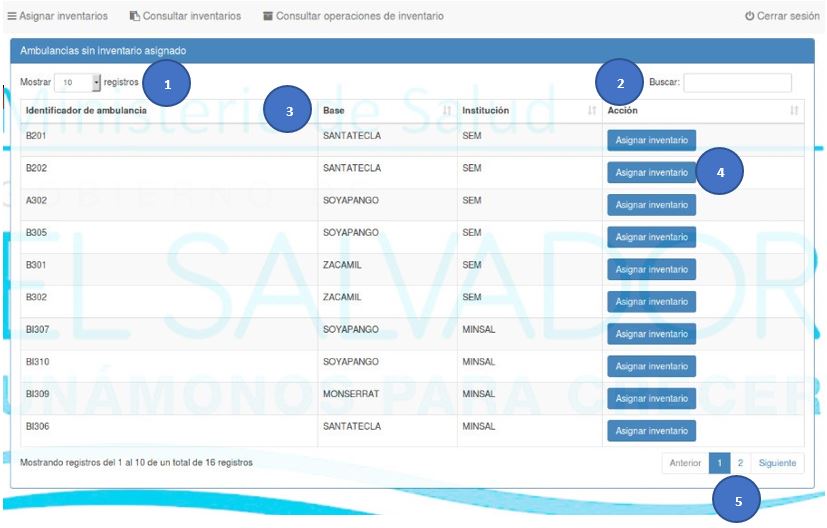
****

Figura 4.13. Pantalla de asignación de inventarios.

La pantalla consultar inventarios descrito en la figura 4.15 dispone de múltiples opciones y cuenta con distintos elementos que se componen de la siguiente forma:

1. **Selector de ambulancia:** Esta opción permite seleccionar cierta ambulancia para hacer la consulta de su respectivo inventario.
2. **Botón ver inventario:** Este botón permite realizar la búsqueda de inventario de la ambulancia que se ha seleccionado previamente.
3. **Resultados de inventario:** se presentan en forma de tabla que permite visualizar de manera ordenada el inventario, cuenta con 7 columnas encabezadas por: medicamento, forma farmacéutica, concentración, número de lote, cantidad, fecha de vencimiento y recibido por.
4. **Buscador:** este buscador permite ingresar mediante teclado múltiples opciones y al ingresarse la búsqueda se realiza de manera automática.

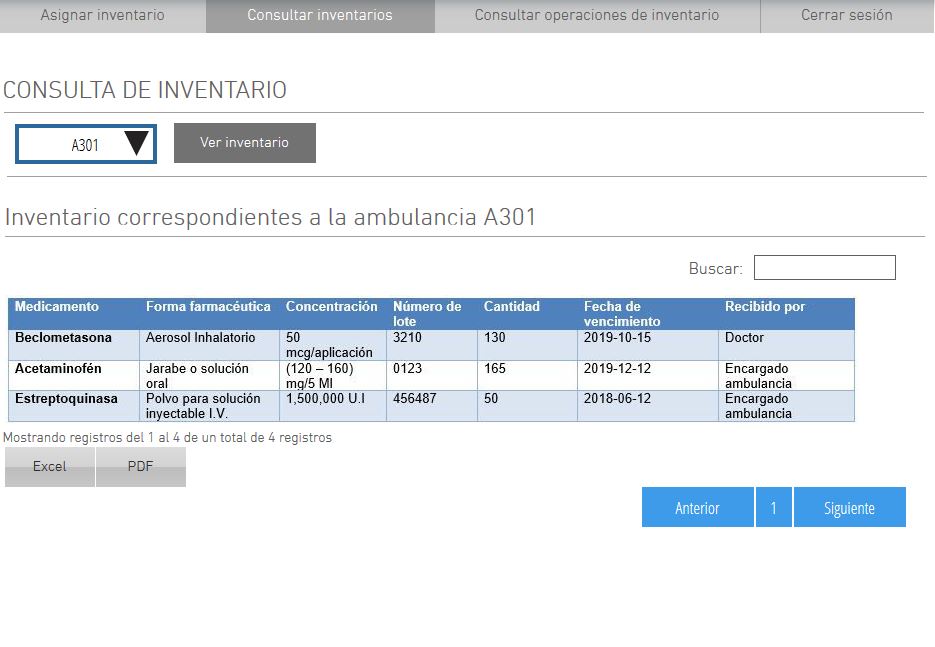
****

Figura 4.14. Diseño de pantalla del módulo Consultar inventarios del sistema web SICIA.



Figura 4.15. Pantalla del módulo Consultar inventarios del sistema web SICIA.

El módulo “Consultar operaciones de inventario” descrito en la figura 4.17, despliega una nueva página en la que se cuenta con distintos elementos que se componen de la siguiente forma:

1. **Calendario desplegable de fecha inicial y final:** Esta opción permite desplegar un calendario en cuál el usuario seleccionará la fecha inicial y final de los resultados que desea buscar.
2. **Menú de tipo de operación:** en esta opción el usuario seleccionará el tipo de operación mediante las opciones que se desplegarán.
3. **Selector de ambulancia:** Esta opción permite seleccionar cierta ambulancia para hacer la consulta de las operaciones que se desee.
4. **Botón ver operaciones:** esta opción mostrará los distintos tipos de operaciones que maneja el inventario de las ambulancias.
5. **Resultados:** se presentan en forma de tabla que permite visualizar de una manera ordenada el inventario, cuenta con 6 columnas encabezadas por: nombre, descripción, tipo, cantidad, tipo de operación y la fecha de vencimiento del medicamento.
6. **Botón Excel:** Este botón permite exportar la información que muestra la tabla en un documento de tipo hoja de cálculo.
7. **Botón PDF:** Este botón permite exportar la información que muestra la tabla en un documento de tipo PDF.
8. **Buscador:** este buscador permite ingresar mediante teclado múltiples opciones y al ingresarse la búsqueda se realiza de manera automática.

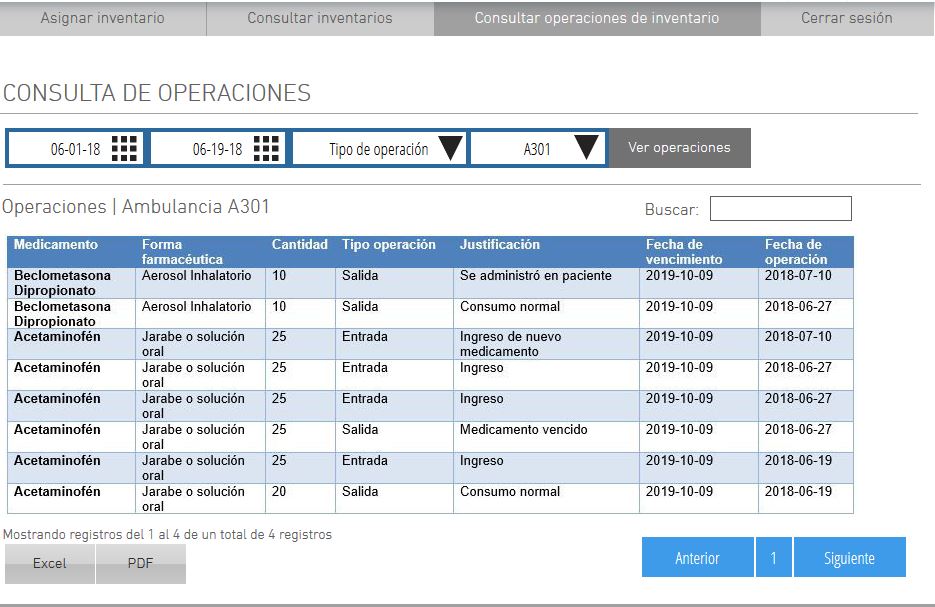


Figura 4.16. Diseño de pantalla del módulo Consultar operaciones de inventario del sistema web SICIA.

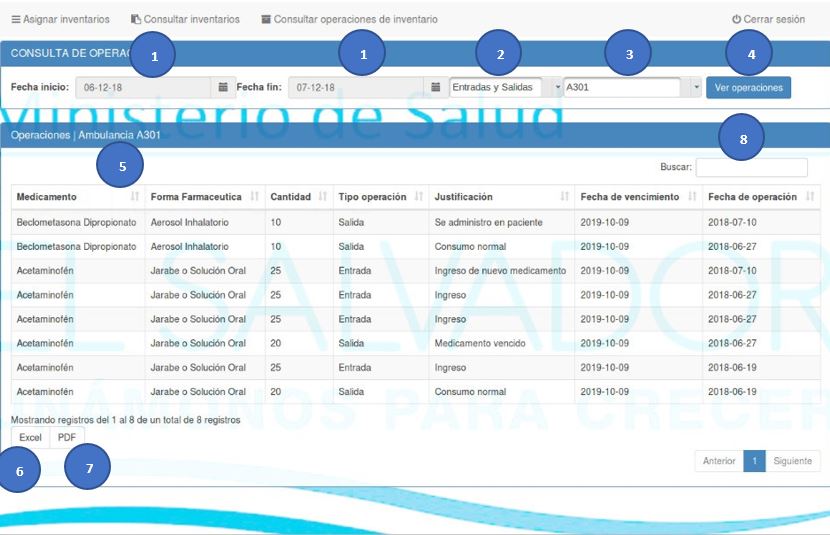
****

Figura 4.17. Pantalla del módulo Consultar operaciones de inventario del sistema web SICIA.

**CAPÍTULO 5. DESARROLLO Y PRUEBAS DEL SISTEMA**

Como en cualquier proyecto de creación de software se es necesario aplicar ciertas etapas o pasos para que el desarrollo del mismo sea óptimo; estas etapas incluyen el análisis de los requerimientos, diseño, programación, verificación, corrección y pruebas para depurar la aplicación. En el capítulo 4 se presentó el diseño del módulo de inventario de ambulancia y el sistema SICIA, dicho diseño es el que en el desarrollo del módulo y sistema siguieron para que la programación de ambos fuese óptima, además se realizaron pruebas unitarias para verificar la calidad y el óptimo desempeño del módulo y el sistema.

**5.1 METODOLOGÍA DE DESARROLLO**

Se utilizó como base de desarrollo el modelo “cascada”, que es uno de los más utilizados en desarrollo de software; algunas etapas ya fueron mencionadas anteriormente en el capítulo, pero cabe recalcar que el proceso ha tenido una duración aproximada de 4 meses, en los que se hizo uso de múltiples herramientas con el objetivo de que la programación y diseño del software fuese óptima.

**5.1.1. Distribución de labores de programación**

Con el propósito de que el desarrollo del módulo de control de inventario y del sistema de control de inventario de ambulancias, tuviera un desarrollo paralelo, se hicieron divisiones en ambos proyectos, con el fin de aligerar un poco la carga de trabajo. El primer paso fue el de la creación de tablas extras que se necesitarían para el manejo del inventario, ya que no se contaba con las tablas necesarias para la generación de este. Una vez creada las nuevas tablas que se ocuparían se procedió al desarrollo del módulo y el sistema.

Para el módulo de control de inventarios de ambulancias se hizo una división de 3 sub módulos: el primero fue el de mostrar el inventario actual, el segundo fue el de realizar el ajuste de inventario debido a incidencia y por último el de realizar un ingrese de consumo de medicamentos debido a la atención de alguna emergencia.

Para el sistema de control de inventario de ambulancias también se realizó una división de sub módulos, estos son: asignar inventario a una ambulancia que tenga inventario pendiente de entrega, consultar el inventario de cada ambulancia disponible y, por último, el de consultar las operaciones de inventario de cada ambulancia, es decir entradas y salidas de inventario.

**5.1.2. Herramientas de desarrollo**

Se realizó una investigación previa en el capítulo 2 acerca de los sistemas operativos, dispositivos móviles, herramientas utilizadas y demás para las aplicaciones que son desarrolladas por los dispositivos móviles en la actualidad, gracias a esta investigación se logró identificar las mejores herramientas para el desarrollo del módulo, estas son listadas a continuación:

* Sistema operativo móvil: Android 4.4 (API 19) KitKat
* Motor de base de datos: SQLite
* Lenguaje de programación JAVA orientado a Android para el funcionamiento y XML para estructura y presentación de las vistas.
* Otras tecnologías: REST API, GIT.

Con respecto a las herramientas utilizadas para el desarrollo del sistema de control de inventario de ambulancias, son las siguientes:

* Motor de base de datos: PostgreSQL
* Lenguaje de programación JAVA para el desarrollo del back end y JSF para el desarrollo del front end.
* Otras tecnologías: REST API, Apache Shiro, GIT.

Para la implementación y desarrollo del módulo y del sistema se utilizaron otras herramientas que permitieron que la elaboración de ambos proyectos se volviera más sencilla, dichas herramientas se presentan a continuación:

* Android Studio: Este se puede decir que es por excelencia el IDE más óptimo para el desarrollo de aplicaciones para Android ya que ha sido desarrollado por el propio Android, este está basado en Android IntelliJ IDEA que analiza el código, buscando conexiones en todos los archivos del proyecto y lenguajes, con esta información, proporciona asistencia de codificación y profundidad, navegación rápida, análisis inteligente de errores.
* Eclipse IDE for Java EE Developers: Es uno de los IDE más recomendable y más estable para la creación de aplicaciones Java EE y Web, este incluye un IDE de java que cuenta con muchas herramientas para la integración de diferentes herramientas para Java EE, JSF, JPA, entre otros, además de incluir muchos paquetes como herramientas de desarrollo de Java, Java EE, Javascript. Además, cuenta con la integración de GIT, Maven, editores XML, y muchas funcionalidades más, que lo hace un IDE ideal para el desarrollo web y la utilización de frameworks.
* GIT: Es un software diseñado para el control de versiones, es decir versiones de un proyecto o archivos, esto permite que el desarrollo de un proyecto se pueda hacer desde varias áreas de trabajo y siempre manteniendo actualizada la versión del proyecto, esto también permite manejar diferentes versiones de un mismo proyecto o archivo, ya que en el mundo de la programación, un error en un línea de cientos y cientos de líneas de código pueden arruinar un proyecto al completo, con GIT las personas pueden estar un poco más seguras, ya que gracias a las diferentes versiones que se generan del proyecto o archivo, es muy sencillo volver a la versión anterior.
* PgAdmin: Es un gestor de base de datos de PostgreSQL, este es un software de código abierto, es decir no tiene costo alguno su uso, y este es mejorado por miles de colaboradores alrededor del mundo, este gestor tiene como propósito satisfacer las necesidades de los diferentes usuarios que buscan desde lo más básico como realizar consultas SQL sencillas, hasta la creación de bases de datos con una cantidad inmensa de tablas.

**5.2. ETAPAS DE PRUEBAS DEL SISTEMA**

La realización de las pruebas de software comprende una serie de actividades dentro de la etapa de desarrollo de las aplicaciones y es de mucha importancia en todo buen proyecto, estas presentan información respecto a la calidad del producto que se entregado, garantizando una óptima solución, y además ayudan a la depuración de procesos y funcionalidades; comprende una retrospectiva de las especificaciones iniciales del diseño y codificación.

**5.2.1. METODOLOGÍA DE LA PRUEBA**

La metodología que se realizó son las pruebas unitarias, las cuales tienen como objetivo probar cada una de las partes del funcionamiento del Módulo de inventario y del SICIA para evidenciar que este trabaje de forma correcta, asegurando que tenga el resultado esperado en cada una de las acciones, es por ello que estas pruebas fueron efectuadas paralelamente con el desarrollo de las aplicaciones, tras la implementación de cada nueva función, flujos inesperados, entre otros. En la Tabla 5.1 se presenta el formato de las pruebas unitarias utilizado.

Tabla 5.1. Formato de pruebas unitarias.

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | Número de prueba unitaria. |
| **Descripción** | Detalle de la prueba. |
| **Objetivos** | Lo que se desea lograr con la prueba. |
| **Condiciones** | Lo necesario para realizar la prueba. |
| **Resultado Esperado** | El resultado que se espera obtener con la prueba. |
| **Resultado Obtenido** | El resultado que se obtuvo luego de la realización de la prueba. |

En la Tabla 5.2 se presenta una de las pruebas unitarias llevadas a cabo en el módulo de georreferencia. En el ANEXO E se encuentran el total de pruebas unitarias realizadas.

Tabla 5.2. Prueba unitaria “Acceso al módulo de control de inventario (SAPHCA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Modulo Inventario** |
| **Prueba No.** | 1 |
| **Descripción** | Acceso al módulo de control de inventario (SAPHCA) |
| **Objetivos** | * Mostrar el inventario activo de la ambulancia |
| **Condiciones** | * Conexión estable y rápida a internet |
| **Resultado Esperado** | Mostrar el inventario actual de la ambulancia |
| **Resultado Obtenido** | Se mostró todo el inventario activo que posee la ambulancia |

**CAPÍTULO 6. CONCLUSIONES Y RECOMENDACIONES**

**6.1 Conclusiones**

Se desarrolló un módulo para el sistema SAPHCA del MINSAL, llamado módulo de control de inventario para ambulancias, la implementación de este módulo hace hincapié en el buen uso que se le puede dar a la tecnología, ya que el proceso pasó de ser manual a digital con lo se podrá llevar a cabo un mejor control de los insumos y medicamentos que utiliza cada ambulancia del SEM; esto da paso a que se pueda llevar un registro en ciertos períodos de tiempo y hacer proyecciones de la cantidad de medicamentos e insumos que se requieren en cada ambulancia.

Se adquirió el conocimiento necesario acerca de las herramientas que se utilizaron para la realización del proyecto y se realizó un estudió a profundidad para el entendimiento del código fuente de la aplicación, con el fin de que el módulo de control de inventario de ambulancias del SEM siguiera los estándares de programación que posee el código ya escrito de la aplicación.

El desarrollo del módulo de control de inventario de ambulancias del SEM se realizó con éxito; además de esto, se desarrolló un sistema web que permite que permite el control de inventario de las ambulancias; dicho sistema permite el enlace con el módulo también desarrollado en esta etapa y cuenta con opciones muy útiles y muy sencillas de manejar; lo que hace que se dé un salto de calidad y una mejora en vista a futuros cambios en el sistema web y el módulo, ya que el desarrollo de código está bien estructurado y codificado.

**6.2 Recomendaciones**

Debido a situaciones experimentadas al inicio del proceso de graduación, se recomienda definir de manera más formal el inicio del proceso, ya que hubo dificultades para establecer los primeros análisis y requerimientos del sistema, además de establecer reuniones con las partes involucradas.

Capacitar al personal encargado para el correcto manejo de módulo de control de inventario de ambulancias por medio de video tutoriales cortos y concisos, donde se muestre el correcto uso del módulo; todo esto con el fin de optimizar al máximo el uso que se dé al mismo.

En este proyecto se vio involucrada más de una entidad; por lo que se dificulto coordinar reuniones con personal de ambas entidades, debido al tiempo disponible por el personal; ya que este no coincidía todo el tiempo; pero sí hubo mucha voluntad de parte del personal de ambas entidades por establecer reuniones para recopilar los requerimientos del sistema; por lo que se sugiere que los proyectos involucran una entidad a lo mucho para evitar cualquier posible problema.

**REFERENCIAS**

**Fuentes electrónicas**

https://www.mindmeister.com/es/644743388/software-conceptos-b-sicos (Consulta: 20/05/2018).

https://www.digitallearning.es/curso-java-para-android.html (Consulta: 21/05/2018).

https://obux.wordpress.com/2017/03/03/que-es-y-para-que-sirve-android-sdk/ (Consulta: 21/05/2018).

https://www.xatakandroid.com/programacion-android/kotlin-ya-es-un-lenguaje-oficial-en-android-que-implicaciones-tiene-y-por-que-es-tan-importante (Consulta: 21/05/2018).

https://fergarciac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/ (20/05/2018).

https://www.softwaredoit.es/definicion/definicion-depuracion.html (Consulta: 20/05/2018).

https://developer.android.com/studio/intro/?hl=es (Consulta: 22/05/2018).

https://developer.android.com/studio/projects/create-project?hl=es (Consulta: 22/05/2018)

http://www.informajoven.org/info/informacion/I\_12\_4.asp (Consulta: 20/05/2018).

https://sites.google.com/site/dispositivosmovilesyulianah/concepto-general-dispositvo-movil/categoria-de-dispositivos-moviles?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1 (Consulta: 20/05/2018)

https://www.consumoteca.com/electronica/telefonos/telefono-movil/ (Consulta: 20/05/2018)

https://www.ecured.cu/Sistema\_de\_navegaci%C3%B3n\_para\_autom%C3%B3viles (Consulta: 20/05/2018)

https://es.ccm.net/contents/394-pda-organizador (Consulta: 20/05/2018)

https://www.ibm.com/support/knowledgecenter/es/SSMKHH\_9.0.0/com.ibm.etools.mft.doc/ac55710\_.htm (Consulta: 28/05/2018)

http://www.palentino.es/blog/servicios-web-utilidades-estandares-y-beneficios/ (Consulta: 29/05/2018)

https://www.ecured.cu/Servicios\_Web (Consulta: 28/05/2018)

https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos (Consulta: 28/05/2018)

https://www.json.org/json-es.html (Consulta: 05/06/2018)

https://www.ibm.com/support/knowledgecenter/es/SSAW57\_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/cweb\_javaserver\_faces.html (Consulta: 05/06/2018)

http://wikis.uca.es/wikiCE/index.php/JSF (Consulta: 05/06/2018)

https://developer.android.com/training/data-storage/sqlite (Consulta: 22/05/2018)

https://techterms.com/definition/dbms (Consulta: 22/05/2018)

https://searchdatamanagement.techtarget.com/definition/RDBMS-relational-database-management-system (Consulta: 22/05/2018)

https://www.google.com.sv/amp/s/blog.panoply.io/mysql-vs-oracle%3fhs\_amp=true (Consulta: 22/05/2018)

http://www.dataversity.net/brief-history-database-management/ (Consulta: 22/05/2018)

https://www.pluralsight.com/blog/software-development/relational-non-relational-databases (Consulta: 21/05/2018)

https://www.informatica-hoy.com.ar/soluciones-moviles/Cual-es-el-mejor-sistema-operativo-para-un-smartphone.php (Consulta: 21/05/2018)

http://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html (Consulta: 20/05/2018)

https://darioalberto.weebly.com/so-moacutevildefinicioacutencaracteriacutesticas-y-ejemplos.html (Consulta: 20/05/2018)

https://edu.gcfglobal.org/es/informatica-basica/sistemas-operativos-para-dispositivos-moviles/1/ (Consulta: 20/05/2018)

https://blogthinkbig.com/los-sistemas-operativos-moviles-que-mas-utilizamos (Consulta: 21/05/2018)

https://searchdatacenter.techtarget.com/es/definicion/Sistema-operativo (Consulta: 21/05/2018)

http://www.pcactual.com/noticias/actualidad/ponemos-nota-sistemas-operativos-moviles-populares-2\_12412 (Consulta: 21/05/2018)

https://andro4all.com/2018/08/versiones-android-historia (Consulta: 20/05/2018)

https://www.android.com/intl/es\_es/history/ (Consulta: 21/05/2018)

https://norfipc.com/celulares/todas-versiones-sistema-operativo-android.php (Consulta: 20/05/2018)

https://www.adslzone.net/tag/android/ (Consulta: 23/05/2018)

https://norfipc.com/celulares/todas-versiones-sistema-operativo-ios-apple.php (Consulta: 21/05/2018)

https://www.applesfera.com/ios/la-evolucion-de-ios-desde-sus-origenes-una-carrera-para-ser-el-mejor-sistema-operativo-movil-de-la-historia (Consulta: 01/06/2018)

https://es.support.t-mobile.com/docs/DOC-36411 (Consulta: 01/06/2018)

https://reparacionescelulares.com/blog/versiones-de-ios-cual-es-la-mejor/ (Consulta: 01/06/2018)

https://www.movilcelular.es/windows-phone/versiones/5 (Consulta: 01/06/2018)

https://copro.com.ar/Historia\_de\_las\_versiones\_de\_Windows\_Phone.html (Consulta: 01/06/2018)

https://www.xatakamovil.com/sistemas-operativos/windows-phone-8-1-toda-la-informacion (Consulta: 01/06/2018)

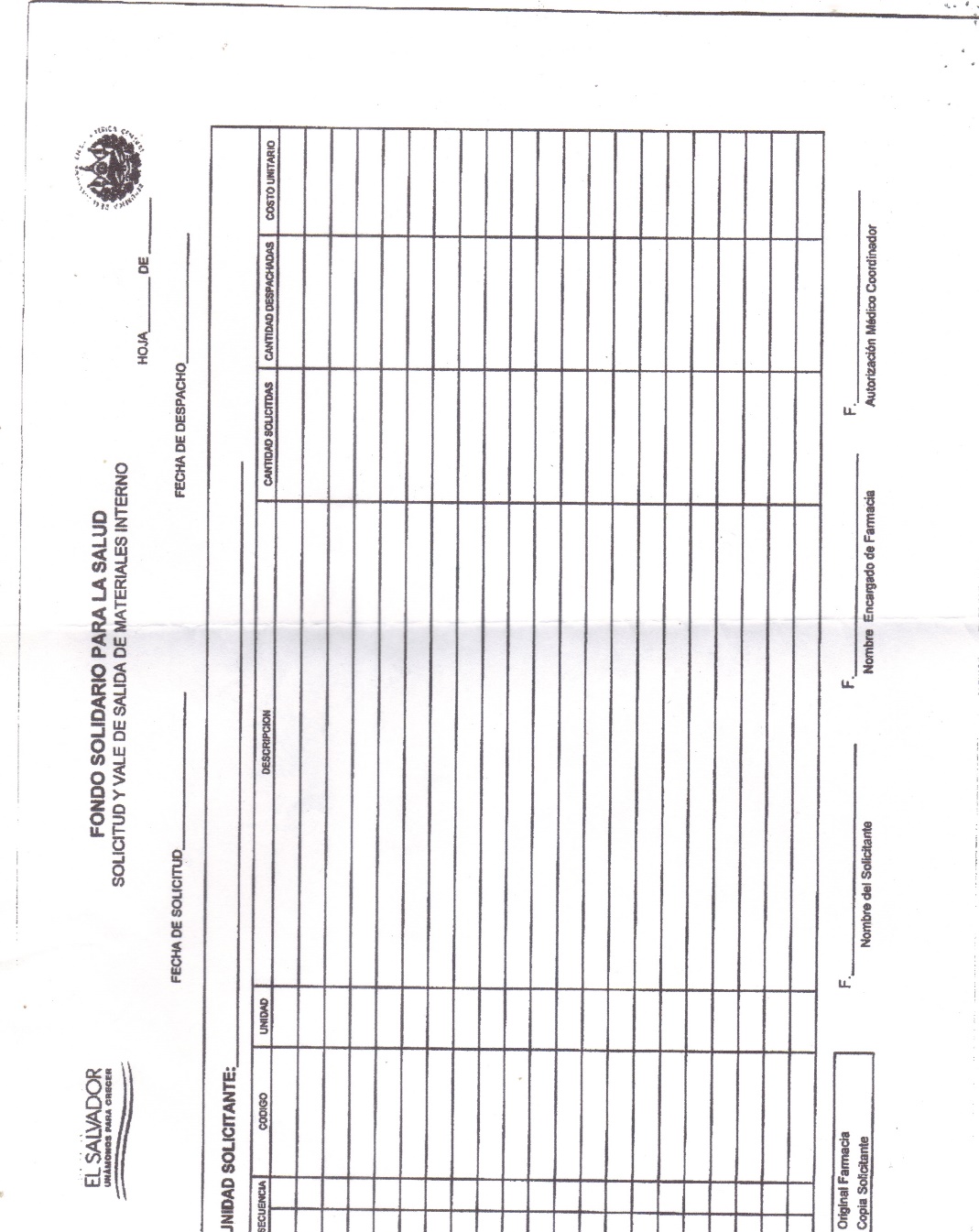
https://www.definicionabc.com/economia/inventario.php (Consulta: 09/09/2018)

**ANEXO A**

SOLICITUD Y VALE DE SALIDA DE MATERIAL INTERNO

Y

HOJA DE TRANSFERENCIA DE INVENTARIO

Figura A-1. Hoja de solicitud y vale de salida de material interno.

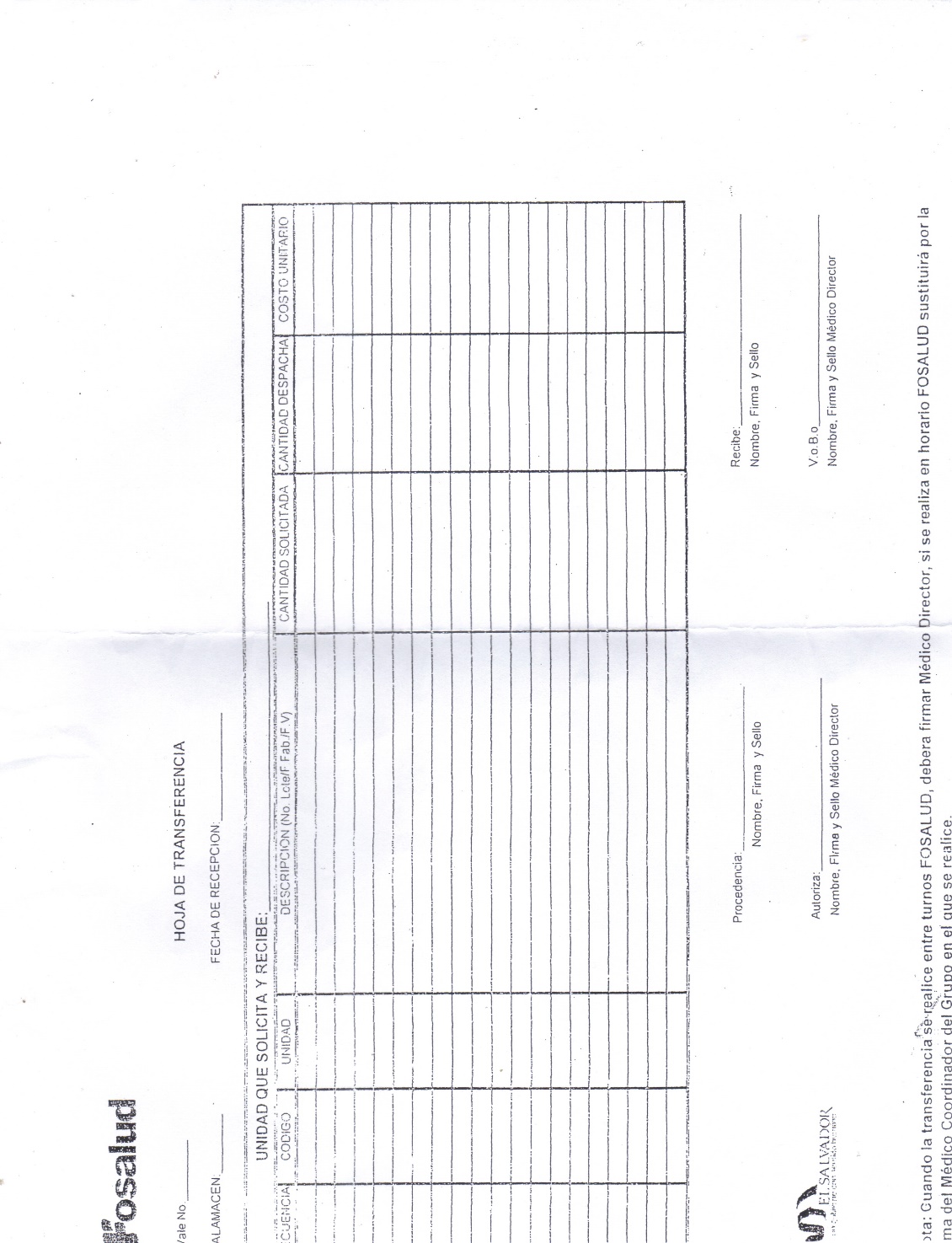


Figura A-2. Hoja de transferencia de inventario

**ANEXO B**

CASOS DE USO

Tabla B-1. Caso de uso “Registrar consumo de medicamentos (SAPHCA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador** | 2 | | |
| **Nombre del caso de uso:** | Registrar el consumo de medicamentos (SAPHCA) | | |
| **Creado por:** | René Masin | **Fecha de creación:** | 26/05/2018 |
| **Participantes:** | Persona asignada como responsable de ambulancia. | | |
| **Descripción:** | Corresponde al registro de la información referente al consumo de un determinado medicamento, a través del llenado de un formulario que se mostrará en pantalla, con las opciones de aceptar o cancelar el ingreso. | | |
| **Pre-condiciones:** | Haber ingresado al módulo de control de inventario de ambulancias de la aplicación SAPHCA y haber seleccionado la opción Ingresar consumo, además que la ambulancia debe tener un inventario asignado. | | |
| **Post-condiciones:** | Almacenar la información ingresada en el formulario que se presenta en la pantalla de registro de consumo de medicamento en una base de datos y se mostrará un mensaje que confirme el éxito o de almacenamiento de la información. | | |
| **Flujo Principal:** | 1 Seleccionar en el menú la opción “Ingresar consumo”.  2. El usuario deberá llenar el formulario con la información que se necesita para registrar un consumo de medicamento. En campos donde deba ingresar texto se vuelve necesario que el usuario seleccione el campo y luego ingrese la información, en otros casos la información se solicitará mediante la selección de una opción presentadas en el formulario.  3. El usuario deberá presionar la opción de Agregar, dependiendo de cuantos medicamentos desee ingresar al consumo.  4. Finalmente el usuario confirma el registro presionando el boton Aceptar, un dialogo aparecera en pantalla.  5. El usuario hace click en el botón Aceptar del dialogo de confirmacion. | | |
| **Flujo Alterno:** | 4A. El usuario decide cancelar el registro en el diálogo de confirmación y se muestra la pantalla consumo de medicamento. | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | Para el flujo principal se asume que el usuario ya inició sesión en la aplicación . | | |
| **Notas:** | Registrado el consumo de un medicamento no será posible eliminarlo de la base de datos. | | |

Tabla B-2. Caso de uso “Realizar ajuste de inventario de ambulancia (SAPHCA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 3 | | |
| **Nombre del caso de uso:** | Realizar ajuste de inventario de ambulancia (SAPHCA) | | |
| **Creado por:** | José Polanco | **Fecha de creación:** | 28/05/2018 |
| **Participantes:** | Persona asignada como responsable de ambulancia. | | |
| **Descripción:** | Esta opción permitirá al encargado de la ambulancia de la SEM remover manualmente en la Tablet medicamentos de la ambulancia que hayan expirado o que se haya dañado por una mala manipulación. | | |
| **Pre-condiciones:** | Haber accedido previamente al módulo de control de inventario de ambulancias y que la ambulancia posea un inventario asignado. | | |
| **Post-condiciones:** | Remover de la base de datos la cantidad ingresada de medicamentos vencidos.  El usuario recibirá una notificación de eliminación exitosa. | | |
| **Flujo Principal:** | 1 Seleccionar en el menú la opción “Ajuste de inventario”.  2. El usuario deberá llenar el formulario con la información que se necesita para registrar un consumo de medicamento. En campos donde deba ingresar texto se vuelve necesario que el usuario seleccione el campo y luego ingrese la información, en otros casos la información se solicitará mediante la selección de una opción presentadas en el formulario.  3. El usuario deberá presionar la opción de Agregar, dependiendo de cuantos medicamentos desee registrar un ajuste.  4. Finalmente el usuario confirma el registro presionando el boton Aceptar, un dialogo aparecera en pantalla.  5. El usuario hace click en el botón Aceptar del dialogo de confirmacion. | | |
| **Flujo Alterno:** | 4A. El usuario decide cancelar el registro en el diálogo de confirmación y se muestra la pantalla de Ajuste de inventario. | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | No aplica | | |
| **Notas:** | Al remover ciertos medicamentos ya caducados, existe la posibilidad que el stock de dichos medicamentos que se han cargado en la ambulancia se agote. | | |

Tabla B-3. Caso de uso “Acceder al sistema web (SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 4 | | |
| **Nombre del caso de uso:** | Acceder al sistema web (SICIA) | | |
| **Creado por:** | César Mejía | **Fecha de creación:** | 8/06/2018 |
| **Participantes:** | Persona registrada como usuario del sistema SICIA. | | |
| **Descripción:** | Inicio de sesión del sistema web, en la que el usuario iniciará sesión con sus respectivas credenciales. | | |
| **Pre-condiciones:** | Para poder ingresar al sistema web, la persona debe estar registrada como usuario de este. | | |
| **Post-condiciones:** | Ingresar al sistema web. | | |
| **Flujo Principal:** | 1. Digitar sus respectivas credenciales en los cuadros de inicio de sesión.  2. Hacer click en el botón de inicio de sesión. | | |
| **Flujo Alterno:** | No aplica | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | El usuario tiene conexión a internet y posee credenciales válidas. | | |
| **Notas:** | No aplica. | | |

Tabla B-4. Caso de uso “Realizar asignación de inventario (SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 5 | | |
| **Nombre del caso de uso:** | Realizar asignación de inventario (SICIA) | | |
| **Creado por:** | René Masin | **Fecha de creación:** | 8/06/2018 |
| **Participantes:** | Persona registrada como usuario del sistema web. | | |
| **Descripción:** | Permite al usuario asignar un inventario a aquellas ambulancias que no cuenten con este. | | |
| **Pre-condiciones:** | El usuario tuvo que haber iniciado sesión de forma exitosa.  Ambulancias recientemente registradas y que no cuenten con un inventario asignado. | | |
| **Post-condiciones:** | Se crea un inventario en la base de datos para la ambulancia seleccionada por el usuario. | | |
| **Flujo Principal:** | 1. Acceder a la opción asignar inventario.  2. Hacer clic sobre el botón Asignar inventario, que corresponde a la fila de la ambulancia seleccionada. | | |
| **Flujo Alterno:** | No aplica | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | Hay al menos una ambulancia que no tiene asignado un inventario. | | |
| **Notas:** | No aplica. | | |

Tabla B-5. Caso de uso “Consultar inventario de ambulancias (SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador** | 6 | | |
| **Nombre del caso de uso:** | Consultar inventario de ambulancias (SICIA) | | |
| **Creado por:** | René Masin | **Fecha de creación:** | 26/05/2017 |
| **Participantes:** | Persona registrada como usuario del sistema web. | | |
| **Descripción:** | El sistema Web contará con una pantalla donde se generarán tablas con información de inventario de una ambulancia determinada, es decir, el usuario seleccionará la ambulancia de la cual quiere ver la información. | | |
| **Pre-condiciones:** | Ingresar al módulo de consultas de la aplicación Sistema de consulta de inventario de ambulancias y haber seleccionado la opción Consultar inventario de ambulancias. | | |
| **Post-condiciones:** | Mostrar en pantalla una tabla que detalle la información del inventario que corresponde a la ambulancia seleccionada . | | |
| **Flujo Principal:** | 1. El usuario deberá seleccionar la ambulancia de la cual quiere obtener información.  2. El usuario deberá presionar el botón con la opción de Ver inventario.  3. Se mostrará la tabla que muestra la información del inventario de la ambulancia seleccionada. | | |
| **Flujo alterno:** | No aplica. | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | No aplica | | |
| **Notas:** | En el caso en que no se cuente con información para cierta ambulancia se mostrará mensaje que informe al usuario. | | |

Tabla B-6. Caso de uso “Consultar operaciones de inventario(SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 7 | | |
| **Nombre del caso de uso:** | Consultar operaciones de inventario (SICIA) | | |
| **Creado por:** | René Masin | **Fecha de creación:** | 09/06/2018 |
| **Participantes:** | Persona registrada como usuario del sistema web. | | |
| **Descripción:** | Permite al usuario obtener información sobre movimientos del inventario de una ambulancia que este debe seleccionar. La información podrá ser filtrada por un periodo de fecha seleccionado por el usuario, este periodo no puede ser mayor a tres meses atrás desde la fecha en que se realiza la consulta. Además se usa como filtro la ambulancia, es decir se obtiene información únicamente de la ambulancia seleccionada. | | |
| **Pre-condiciones:** | Haber ingresado a la opción Consultar operaciones de inventario. | | |
| **Post-condiciones:** | El usuario podrá visualizar una tabla que detalla la información del inventario. | | |
| **Flujo Principal:** | 1. Seleccionar la opción Consultar operaciones de inventario.  2. Ingresar un periodo de tiempo y ambulancia como filtros.  3. Hacer clic sobre la opción Ver operaciones | | |
| **Flujo Alterno:** | No aplica | | |
| **Excepciones:** | No aplica. | | |
| **Supuestos:** | No aplica. | | |
| **Notas:** | No aplica. | | |

Tabla B-7. Caso de uso “Generar reporte (SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 8 | | |
| **Nombre del caso de uso:** | Generar reporte (SICIA) | | |
| **Creado por:** | Jesús Rodríguez | **Fecha de creación:** | 29/05/2018 |
| **Participantes:** | Persona registrada como usuario del sistema web. | | |
| **Descripción:** | Esta opción se presenta en las pantallas de las opciones Consultar inventario y Consultar operaciones de inventario, y se muestran específicamente al pie de la tabla que contiene la información que se genera en estas pantallas, se cuenta con dos opciones que son, PDF y Excel, que básicamente, generan un archivo con el formato que el usuario selecciona, que contiene la información que se detalla en la tabla que se presenta en pantalla. | | |
| **Pre-condiciones:** | Ingresar a una de las siguiente opciones, Consultar inventarios o Consultar operaciones de inventario, haber realizado las consultas que generan la tabla con la información de acuerdo a las opciones seleccionadas por el usuario. | | |
| **Post-condiciones:** | Se almacena localmente el archivo generado. | | |
| **Flujo Principal:** | 1. El usuario debe hacer clic sobre una de las opciones, PDF o Excel.  2. Se muestra un cuadro de diálogo informando al usuario que el archivo fue generado. | | |
| **Flujo alterno** | No aplica | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | No aplica | | |
| **Notas:** | No aplica | | |

Tabla B-8. Caso de uso “Cerrar sesión (SICIA)”

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | 9 | | |
| **Nombre del caso de uso:** | Cerrar sesión (SICIA) | | |
| **Creado por:** | Jesús Rodríguez | **Fecha de creación:** | 29/05/2018 |
| **Participantes:** | Persona registrada como usuario del sistema web. | | |
| **Descripción:** | Esta opción permite al usuario dar por terminada su sesión, de manera que al ser seleccionada el sistema re direcciona hacia la pantalla de inicio de sesión. | | |
| **Pre-condiciones:** | Contar con una sesión activa. | | |
| **Post-condiciones:** | Se elimina la sesión del servidor y se re direcciona hacia la pantalla de inicio de sesión. | | |
| **Flujo Principal:** | 1. El usuario debe hacer clic sobre la opción Cerrar sesión.  2. Se muestra pantalla de inicio de sesión. | | |
| **Flujo alterno** | No aplica | | |
| **Excepciones:** | No aplica | | |
| **Supuestos:** | No aplica | | |
| **Notas:** | No aplica | | |

**ANEXO C**

DICCIONARIO DE DATOS

**Tabla C-1.** Diccionario de datos de la tabla ctl\_ambulancia

|  |  |  |  |
| --- | --- | --- | --- |
| **ctl\_ambulancia** | | | |
| Columna | Tipo de dato | Datos nulos | Descripción |
| id\_ambulancia | SERIAL | NO | Identificador que posee las diferentes ambulancias con las que contará SEM para asignarlas a las demandas |
| id\_provincia | VARCHAR | NO | Identificador de la provincia |
| id\_unidad | VARCHAR | NO | Identificador de ambulancia. |
| id\_institucion | VARCHAR | NO | Identificador de la institución a la que pertenece. |
| id\_base | VARCHAR | NO | Identificador de la base donde está destacada la ambulancia. |
| ds\_num\_ambulancia | TEXT | NO | Número de ambulancia, identificador SISEM (Tipo Ambulancia-Zona-Correlativo) |
| ds\_ambulancia | TEXT | NO | Descripción de la ambulancia. |
| ds\_tipo\_ambulancia | TEXT | NO | Descripción del tipo de ambulancia (Tipo A, Tipo B o Tipo C) |
| cd\_estado | CHAR | NO | Estado del campo (Activo o Inactivo) |
| usua\_crea | TEXT | SI | Datos de usuario que creó el campo |
| fech\_crea | TIMESTAMP | NO | Fecha y hora de la creación del campo |
| usua\_actu | TEXT | SI | Datos de usuario que modifico el campo |
| fech\_actu | TIMESTAMP | NO | Fecha y hora de la modificación del campo |

**Tabla C-2.** Diccionario de datos de la tabla ctl\_medicamento

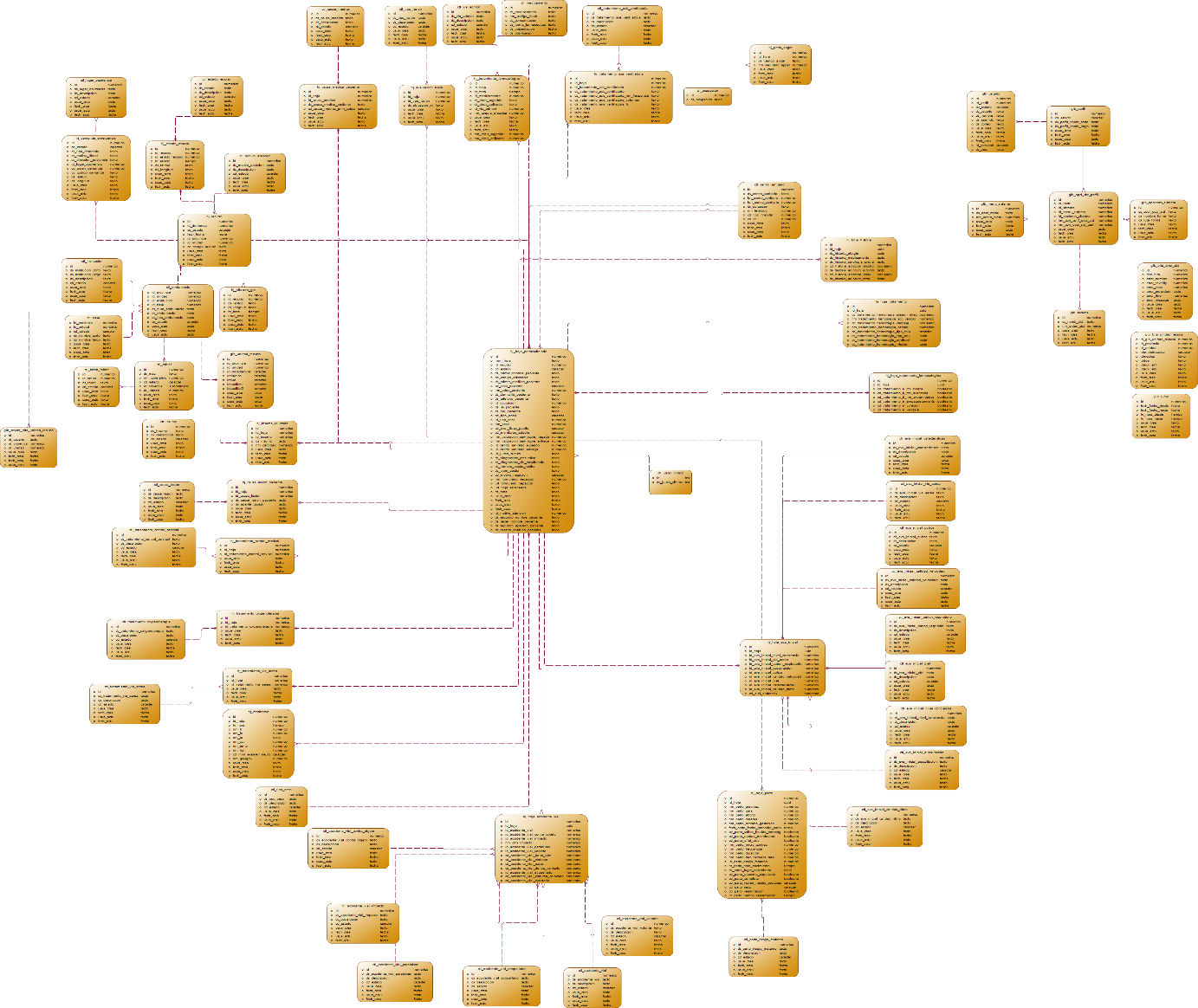
|  |  |  |  |
| --- | --- | --- | --- |
| **ctl\_medicamento** | | | |
| Columna | Tipo de dato | Datos nulos | Descripción |
| id | SERIAL | NO | Identificador del medicamento |
| ds\_medicamento | TEXT | NO | Nombre del medicamento |
| nm\_codigo\_sinab | VARCHAR | NO | Código del medicamento en el SINAB |
| ds\_concentracion | VARCHAR | SI | Concentración del medicamento |
| ds\_forma\_farmaceutica | VARCHAR | SI | Forma farmacéutica del medicamento |
| ds\_presentacion | CHAR | SI | Presentación del medicamento |
| ds\_dosificacion | TEXT | SI | Campo para la unidad de medida |

**Tabla C-3.** Diccionario de datos de la tabla detalle\_inventario

|  |  |  |  |
| --- | --- | --- | --- |
| **detalle\_inventario** | | | |
| Columna | Tipo de dato | Datos nulos | Descripción |
| id\_detalle\_inventario | SERIAL | NO | Identificador del detalle del inventario |
| id\_inventario | INTEGER | NO | Identificador de cada inventario |
| corr\_producto | VARCHAR | NO | Correlativo del producto |
| numero\_vale | VARCHAR | NO | Número de vale |
| nombre\_almacen | VARCHAR | NO | Nombre del almacén |
| destino | VARCHAR | NO |  |
| cantidad | NUMERIC | NO |  |
| precio\_unitario | NUMERIC | NO | Precio unitario del inventario |
| numero\_lote | VARCHAR | NO |  |
| fecha\_vencimiento | DATE | NO |  |
| id\_fuente\_financiamiento | INTEGER | NO | Identificador de la fuente de financiamiento |
| persona\_recibe | VARCHAR | NO | Usuario que recibe el inventario |
| numero\_identificacion | VARCHAR | NO | Número de identificación del usuario |
| observacion | VARCHAR | NO | Campo para registrar algún tipo de observación |
| id\_movimiento | INTEGER | NO | Identificador del movimiento del inventario |
| id\_maestro | INTEGER | NO |  |
| id\_almacen | INTEGER | NO | Identificador del almacén |
| id\_establecimiento | INTEGER | NO | Identificador del establecimiento |

**ANEXO D**

DIAGRAMA DE LA BASE DE DATOS DE LA APLICACIÓN SAPHCA

Figura D-1. Diagrama de la base de datos de la aplicación SAPCHA

**ANEXO E**

PRUEBAS UNITARIAS

Tabla E-1. Prueba unitaria “Registrar consumo de medicamentos (SAPHCA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 2 |
| **Descripción** | Registrar el consumo de medicamentos (SAPHCA) |
| **Objetivos** | * Registrar en la base de datos la información referente al consumo de un determinado medicamento. * Actualizar el inventario de la ambulancia. |
| **Condiciones** | * Tener medicamentos cargados y listos para ser ingresados. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Confirmación de actualización del inventario. |
| **Resultado Obtenido** | Se confirmó que el inventario había sido actualizado. |

Tabla E-2. Prueba unitaria “Realizar ajuste de inventario de ambulancia (SAPHCA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 3 |
| **Descripción** | Realizar ajuste de inventario de ambulancia (SAPHCA) |
| **Objetivos** | * Registrar en la base de datos la información referente al ajuste de un determinado medicamento. * Actualizar el inventario de la ambulancia. |
| **Condiciones** | * Tener medicamentos cargados y listos para ser ingresados. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Confirmación de actualización del inventario. |
| **Resultado Obtenido** | Se confirmó que el inventario había sido actualizado. |

Tabla E-3. Prueba unitaria “Acceder al sistema web (SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 4 |
| **Descripción** | Acceso al sistema web (SICIA) |
| **Objetivos** | * Iniciar sesión en el sistema SICIA. |
| **Condiciones** | * Poseer un usuario y contraseñas válidas. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Re direccionar al menú principal del sistema web SICIA. |
| **Resultado Obtenido** | Se re direccionó a la pantalla principal del sistema web SICIA. |

Tabla E-4. Prueba unitaria “Realizar asignación de inventario (SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 5 |
| **Descripción** | Realizar asignación de inventario (SICIA) |
| **Objetivos** | * Permitir al usuario asignar un inventario a una ambulancia que no posea uno. |
| **Condiciones** | * Debe existir una ambulancia registrada y esta no debe contar con un inventario. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Confirmar que el inventario se le asignó a la ambulancia. |
| **Resultado Obtenido** | Se confirmó que a la ambulancia seleccionada se le asignó un inventario. |

Tabla E-5. Prueba unitaria “Consultar inventario de ambulancias (SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 6 |
| **Descripción** | Consultar inventario de ambulancias (SICIA) |
| **Objetivos** | * Mostrar al usuario el inventario activo de una ambulancia seleccionada. * Una conexión estable y rápida a internet. |
| **Condiciones** | * Haber seleccionado una ambulancia. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Mostrar el inventario actual de la ambulancia seleccionada. |
| **Resultado Obtenido** | Se mostró en una tabla todo el inventario activo que posee la ambulancia seleccionada. |

Tabla E-6. Prueba unitaria “Consultar operaciones de inventario(SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 7 |
| **Descripción** | Consultar operaciones de inventario (SICIA) |
| **Objetivos** | * Permitir al usuario obtener la información de los movimientos realizados en una ambulancia que este seleccione. |
| **Condiciones** | * Haber seleccionado una ambulancia. * Una conexión estable y rápida a internet. |
| **Resultado Esperado** | Mostrar los movimientos que se realizaron en el inventario de la ambulancia seleccionada. |
| **Resultado Obtenido** | Se mostró en una tabla todos los movimientos que se han realizados en el inventario de la ambulancia seleccionada. |

Tabla E-7. Prueba unitaria “Generar reporte (SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 8 |
| **Descripción** | Generar reporte (SICIA) |
| **Objetivos** | * Generar documento PDF o Excel, según el usuario seleccione, que debe contener la información exacta que se muestra en las tablas de “Consultar inventarios” o “Consultar operaciones de inventario”. |
| **Condiciones** | * Haber seleccionado en una de las siguientes opciones, “Consultar inventarios” o “Consultar operaciones de inventario” y haber seleccionado una ambulancia para muestre la información correspondiente. |
| **Resultado Esperado** | Descargar el archivo de manera local. |
| **Resultado Obtenido** | Se descargó el archivo localmente con el formato indicado. |

Tabla E-8. Prueba unitaria “Cerrar sesión (SICIA)”

|  |  |
| --- | --- |
| **Pruebas Unitarias** | **Nombre** |
| **Prueba No.** | 9 |
| **Descripción** | Cerrar sesión (SICIA) |
| **Objetivos** | * Dar por terminada la sesión del usuario |
| **Condiciones** | * Contar con una sesión activa. |
| **Resultado Esperado** | Re direcciona a la pantalla de inicio de sesión. |
| **Resultado Obtenido** | Se re direccionó hacia la pantalla donde se inicia sesión. |

**ANEXO F**

MANUAL DE USUARIO

1. **Módulo de control de inventario de ambulancias**
   1. **Acceso al módulo**

Este manual esta creado con el fin de que los usuarios de la aplicación SAPHCA aprendan el uso de las funcionalidades del módulo de control de inventario.

El acceso al módulo de control de inventario de ambulancias es muy sencillo, ya que en primer lugar se debe ingresar a la aplicación SAPHCA, para ello el usuario deberá ingresar al menú de aplicaciones instaladas en el dispositivo móvil, estando en dicho menú, deberá buscar la aplicación SAPHCA e ingresar a la misma.

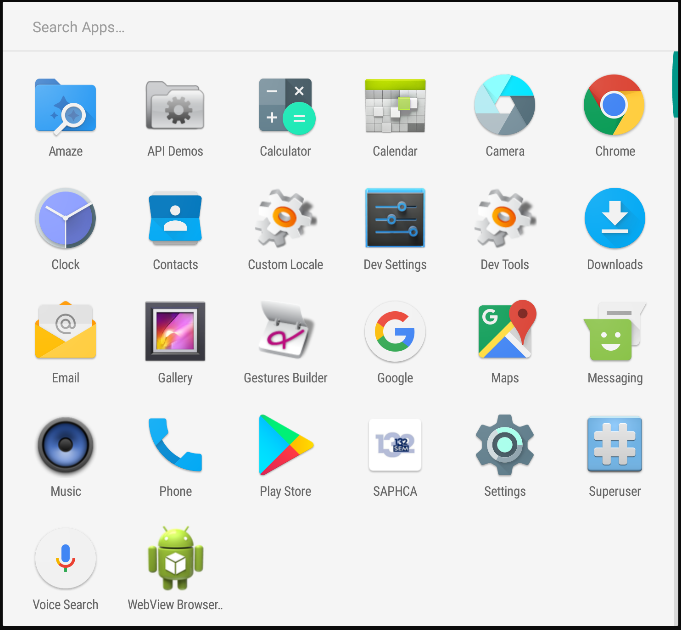
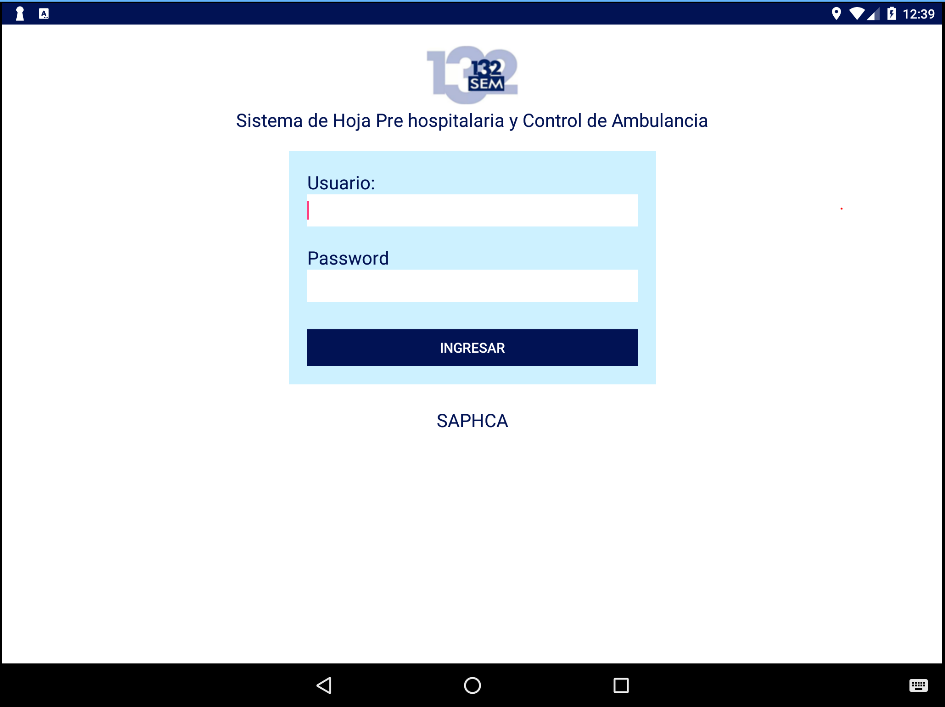


Figura F-1. Menú del dispositivo móvil.

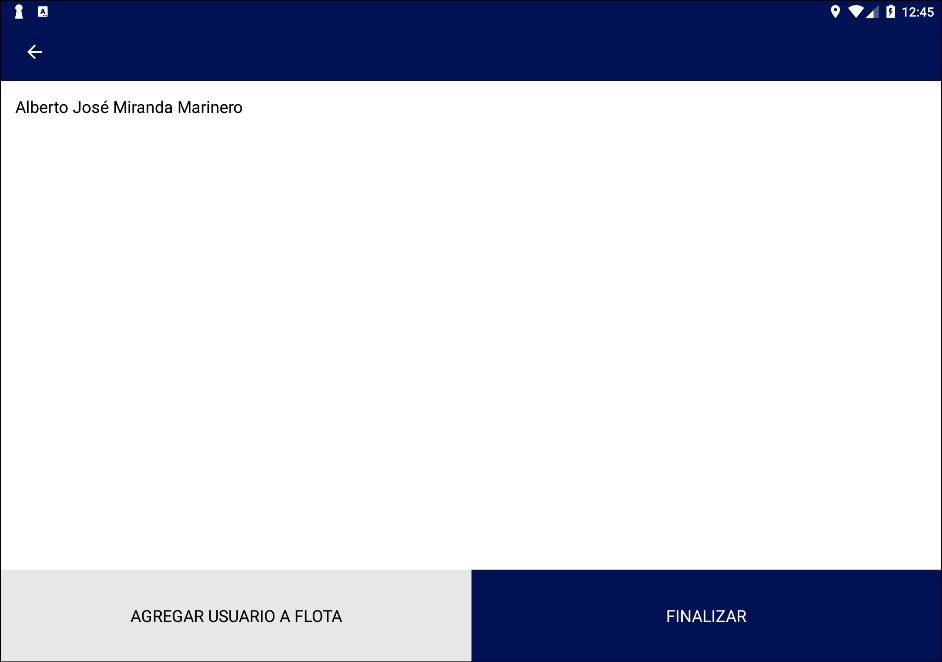
El usuario deberá seleccionar el icono de la aplicación SAPHCA para ingresar, tal y como se marca en rojo en la figura F-1.

* 1. **Pantalla de inicio de sesión**

Al hacer ingreso a la aplicación, se mostrará la pantalla de inicio de sesión como se muestra en la figura F-2, en la que el usuario deberá ingresar sus credenciales para poder tener acceso a las diferentes funcionalidades de la aplicación, después deberá seleccionar el botón “ingresar” y listo. En caso el usuario se equivoque al ingresar sus datos, entonces él los debe volver ingresar de nuevo.

Figura F-2. Pantalla de inicio de sesión de la app SAPHCA.

Una vez se inicia la sesión correctamente, se muestra la pantalla de la figura F-3, en la que se debe seleccionar la opción señalada en rojo para poder tener acceso al menú principal de la aplicación.

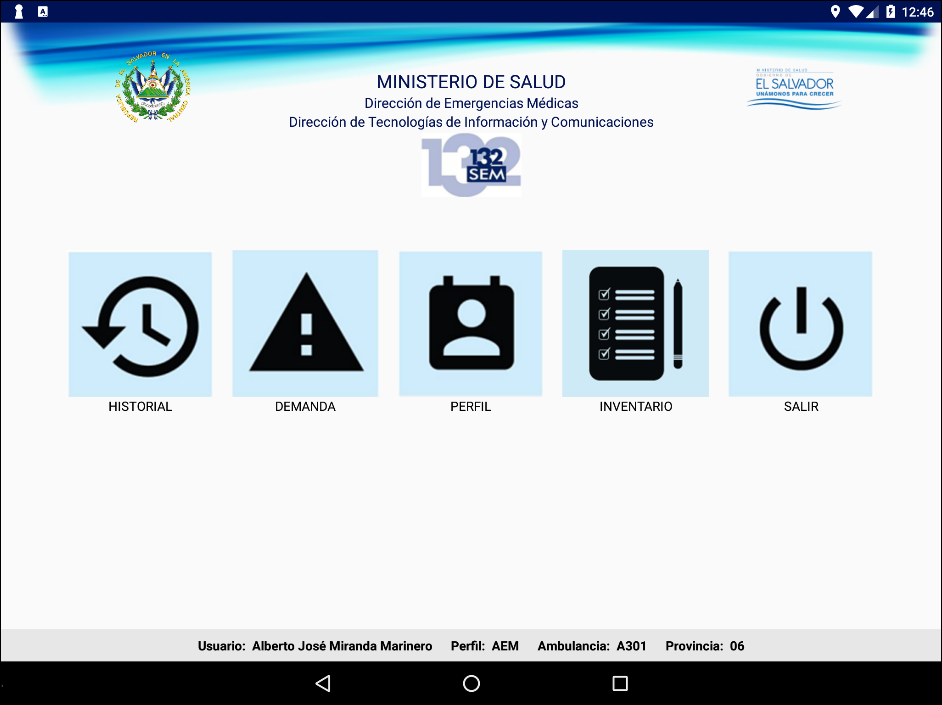
Figura F-3. Pantalla previa al menú principal de la aplicación SAPHCA.

* 1. **Menú principal**

Ahora el usuario tendrá acceso a las distintas opciones del menú, estas opciones son las que se muestran en la figura F-4. Las opciones disponibles son las siguientes:

* Historial
* Demanda
* Perfil
* Inventario
* Salir

El módulo desarrollado para la aplicación es el que está señalado en la figura F-4, para ingresar a dicho módulo solo se debe seleccionar el icono del inventario.

Figura F-4. Menú principal de la aplicación SAPHCA.

* 1. **Pantalla del módulo de control de inventario de ambulancias**

Al ingresar en el módulo de control de inventario de ambulancias, se muestra directamente la pantalla con el inventario actual que se maneja en la ambulancia con la descripción del medicamento, la fecha de vencimiento y la cantidad disponible; esto se ve reflejado en la figura F-5.

Además, esta pantalla permite visualizar la unidad a la que pertenece el inventario. El botón “refresh” permite mantener actualizado el inventario. Dicha pantalla también muestra dos botones en la parte inferior de la pantalla, estos son: ajuste de inventario y el ingreso de consumo.



Figura F-5. Pantalla principal del módulo de control de inventario de ambulancias.

* 1. **Pantalla de ajuste de inventario**

Cuando el usuario ingrese a la opción de ajuste de inventario (figura F-6) tendrá disponible el poder realizar un ajuste al inventario actual que posee la ambulancia, este ajuste se realizará en algún caso que el medicamento sufra alguna incidencia. El usuario podrá seleccionar el medicamento, al haber escogido el medicamento podrá ver la cantidad disponible que posee, la concentración y su fecha de vencimiento. El usuario tendrá dos campos:

* El primero es para ingresar la cantidad de medicamento que desea retirar del inventario, dicha cantidad debe ser reflejada en base a la cantidad de medicamento disponible, además debe ser un número entero.
* El segundo campo es para la justificación de la incidencia, donde el usuario podrá ingresar los motivos por los cuales desechará cierta cantidad de medicamento.

Cuando el usuario haya completado todo lo anterior entonces debe seleccionar el botón “agregar” para que aparezca en la pantalla el detalle del ajuste que se realizará al inventario; todo esto con el fin de que pueda revisar si el ajuste que desea realizar es correcto.

Si el usuario quiere finalizar el ajuste, solo deberá seleccionar el botón “Aceptar” y confirmar la operación, pero si el usuario no desea realizar un ajuste entonces solo de presionar el botón “Regresar”.

****

Figura F-6. Pantalla de ajuste de inventario por incidencia.

* 1. **Pantalla de ingreso de consumo**

La segunda opción que tiene disponible el usuario es la de ingreso de consumo (figura F-5), esta pantalla se muestra en la figura F-7; el usuario podrá seleccionar el medicamento, una vez escogido el medicamento podrá ver la cantidad disponible que posee, la concentración y su fecha de vencimiento.

El usuario tendrá un campo para ingresar la cantidad de medicamento que desea retirar del inventario, dicha cantidad debe ser reflejada en base a la cantidad de medicamento disponible, además debe ser un número entero.

Cuando el usuario haya completado lo anterior entonces debe seleccionar el botón “agregar” para que aparezca en la pantalla el detalle del consumo que se realizará; todo esto con el fin de que pueda revisar si el ajuste que desea realizar es correcto.

Si el usuario quiere finalizar el consumo de medicamentos, solo deberá seleccionar el botón “Aceptar” y confirmar la operación, pero si el usuario no desea completar el consumo entonces solo de presionar el botón “Regresar”.

Figura F-7. Pantalla de consumo de medicamentos.

1. **Sistema SICIA**
   1. **Pantalla de inicio de sesión**

Para poder ingresar al Sistema Web de Control de Inventario de Ambulancia, el usuario deberá iniciar sesión ingresando sus datos en los campos mostrados en pantalla; siempre y cuando este usuario este registrado en el sistema. Cuando haya completado los datos en los campos el usuario deberá seleccionar el botón “Ingresar” y entrará al sistema, dicha pantalla se muestra en la figura F-8, en caso que el usuario se haya equivocado ingresando sus datos, este deberá de reingresar sus datos de nuevo.

Figura F-8. Pantalla de inicio de sesión de sistema SICIA

* 1. **Pantalla principal de sistema SICIA**

Una vez se ingresó al sistema correctamente, se mostrará una pantalla (figura F-9) con 4 opciones diferentes en la barra principal del menú: asignar inventarios, consultar inventario, consultar operaciones de inventario y cerrar sesión.

Figura F-9. Pantalla principal del sistema SICIA

* 1. **Pantalla asignar inventarios**

Para acceder a esta pantalla basta con seleccionar en la barra principal la opción “Asignar inventarios”, una vez esta es seleccionada se mostrará en pantalla. Esta opción tiene distintas funcionalidades (ver figura F-10) como, por ejemplo:

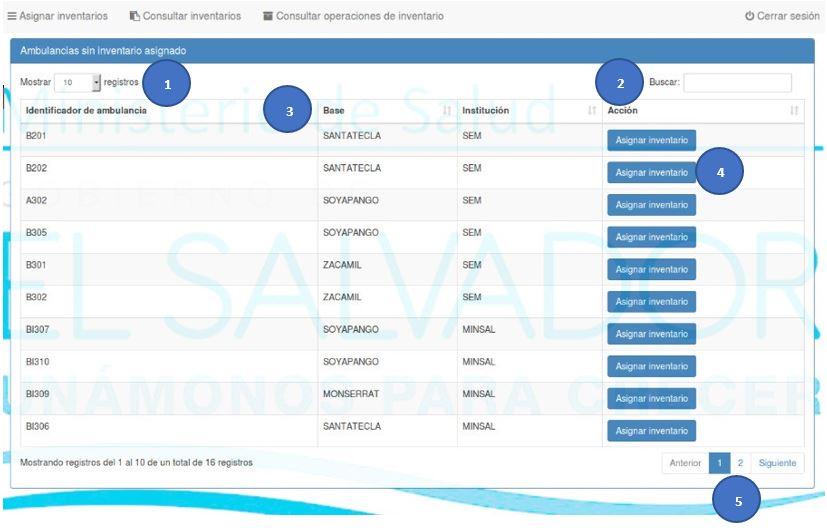
1. Barra desplegable de registros: en esta barra el usuario podrá desplegar la cantidad de registros que desee se muestre en pantalla; siempre y cuando haya alguna ambulancia pendiente de asignación de inventario.
2. Barra de búsqueda: mediante el uso de esta, se puede ingresar en el campo alguna palabra clave que el usuario desee buscar y automáticamente los registros que coincidan con la palabra se mostrarán.
3. Resultados: estos se muestran en forma de tabla para que se pueda tener una mejor visualización de los mismos, los encabezados que se muestran son: identificador de ambulancia, base, institución y acción a realizar.
4. Botón de asignación de inventario: al seleccionar este botón se le asignará el inventario que corresponde a la ambulancia que se encuentra en la misma fila que el mencionado botón.
5. Botones de siguiente/anterior: si los resultados son mayores a diez registros, automáticamente se creará una segunda o más paginas para mostrar los resultados faltantes, y se puede acceder a ellos mediante estos botones.

Figura F-10. Pantalla de asignar inventarios.

* 1. **Pantalla consultar inventarios**

Para acceder a esta pantalla basta con seleccionar la opción “Consultar inventarios” que se encuentra en la barra principal, al seleccionar la opción mencionada muestra la pantalla de la figura F-11. Esta pantalla permite la consulta de inventario de las ambulancias disponibles en el sistema. Las opciones con las que cuenta son varias, entre ellas:

1. Selector de ambulancia: Mediante esta opción el usuario, puede seleccionar una de las ambulancias.
2. Botón de ver inventario: Para que este botón tenga funcionalidad basta con haber seleccionado previamente una ambulancia, al seleccionar este botón se mostrará el inventario correspondiente a la ambulancia.
3. Resultados: estos se muestran en forma de tabla para que se pueda tener una mejor visualización de los mismos, los encabezados que se muestran son: Medicamento, forma farmacéutica, concentración, número de lote, cantidad, fecha de vencimiento y recibido por.
4. Barra de búsqueda: mediante el uso de esta, se puede ingresar en el campo alguna palabra clave que el usuario desee buscar y automáticamente los registros que coincidan con la palabra se mostrarán.

Cabe recalcar que también se posee el botón de siguiente y anterior que funciona de manera similar a como se explicó en el apartado anterior; adicional a esto se pueden exportar los resultados en un documento en formato PDF o formato de hoja de cálculo Excel; para ello solamente se debe seleccionar el botón de Excel o PDF que se muestran en la figura F-11.

Figura F-11. Pantalla de consultar inventarios.

* 1. **Pantalla consultar operaciones de inventario**

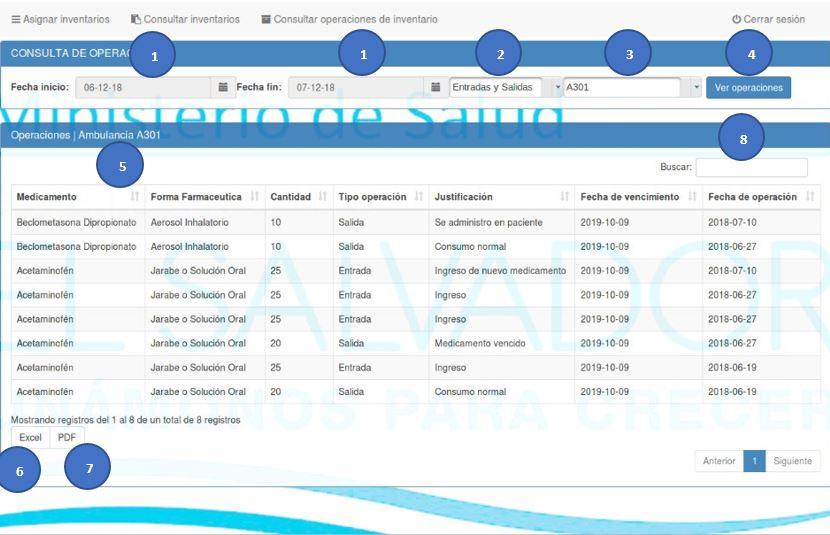
Para acceder a esta opción el usuario debe seleccionar la opción “Consultar operaciones de inventario”, esta pantalla (figura F-12) cuenta con múltiples opciones muy útiles y algunas de ellas son las siguientes:

1. Fecha de inicio y fin: mediante esta opción el usuario puede seleccionar un rango de fechas para filtrar la búsqueda de operaciones de inventario en las fechas que el usuario desee.
2. Selector de operaciones: su función principal es seleccionar el tipo de operaciones que el usuario desea buscar: entradas, salidas o entradas y salidas de inventario.
3. Seleccionador de ambulancia: la función de este, es hacer un filtro en la búsqueda de las operaciones al seleccionar una ambulancia.
4. Botón de ver operaciones: una vez se han seleccionado todos los filtros correspondientes para los resultados, el usuario al seleccionar este botón verá reflejado los filtros aplicados en la correspondiente búsqueda.
5. Resultados: estos se muestran en forma de tabla para que se pueda tener una mejor visualización de los mismos, los encabezados que se muestran son: Medicamento, forma farmacéutica, cantidad, tipo de operación, justificación, fecha de vencimiento y la fecha de la operación.
6. Botón para exportar los resultados en formato de hoja de cálculo de Excel.
7. Botón para exportar los resultados en formato PDF.
8. Barra de búsqueda: mediante el uso de esta, se puede ingresar en el campo alguna palabra clave que el usuario desee buscar y automáticamente los registros que coincidan con la palabra se mostrarán.

Cabe recalcar que en caso que los resultados de la consulta de operaciones sean muchos, estos pasan automáticamente a una nueva página, y para hacer consulta de ellos se deberá hacer uso del botón “siguiente” que fue mencionado con anterioridad y para regresar en los datos se usará el botón “anterior”.

Por último, el usuario podrá navegar y dirigirse a cualquier opción del menú principal en cualquier momento, sin importar la página en la que se encuentre, es decir: asignar inventarios, consultar inventarios y consultar operaciones de inventario.

Figura F-12. Pantalla de consultar operaciones de inventario.



**ANEXO G**

MANUAL TÉCNICO

1. **INTRODUCCIÓN**

El módulo de control de inventario de ambulancia ha sido implementado para facilitar la manipulación del inventario con el que dispone cada ambulancia abastecida por SEM, y para que se pueda tener un mejor control del medicamento utilizado en las emergencias que son atendidas por estas ambulancias, paralelamente se creó el Sistema de Control de Inventario de Ambulancias, para permitir generar reportes de las operaciones de inventario; es decir tanto entradas como salidas de este, de consultar el inventario de todas las ambulancias, y de asignar inventario a las ambulancias que no se les haya asignado.

Este manual está elaborado con el propósito de que se le pueda seguir dando mantenimiento a módulo y al SICIA, y de facilitar dicha tarea de mantenimiento de ambos, cabe recordar que para realizar dicho mantenimiento se debe poseer ciertos conocimientos en programación, sobre todo en el lenguaje Java, tanto como para Android como para web, además de conocimientos en SQLite y PostgreSQL.

1. **HERRAMIENTAS DE DESARROLLO**
   1. **Sistema de control de inventario de ambulancias**

Para el desarrollo del sistema de control de inventario de ambulancia se utilizaron distintas herramientas con el fin de que el producto final sea lo más óptimo posible.

|  |  |
| --- | --- |
| FUNCIÓN | HERRAMIENTAS UTILIZADAS |
| Diseño del front-end | JSF |
| Diseño del back.end | Java |
| Motor de base de datos | PostgreSQL |
| Gestor de base de datos | PgAdmin |
| Programa servidor repositorio para versiones del producto | GitHub |
| Programa cliente repositorio | GitHub |

* 1. **Módulo de inventario de ambulancias SAPHCA**

Para el desarrollo del módulo de control de inventario de ambulancia se utilizaron distintas herramientas con el fin de que el producto final tenga los mejores resultados.

|  |  |
| --- | --- |
| FUNCIÓN | HERRAMIENTAS UTILIZADAS |
| Diseño del front-end | MVC |
| Diseño del back.end | Java |
| Vistas | XML |
| Motor de base de datos | SQLite |
| Gestor de base de datos | SQLiteOpenHelper |
| Programa servidor repositorio para versiones del producto | GitHub |
| Programa cliente repositorio | GitHub |

1. **FUNCIONALIDADES DEL SISTEMA**
   1. **Módulo de inventario de ambulancias SAPHCA**

Para el desarrollo del módulo de control de inventario de ambulancias, se crearon paquetes con el fin de agrupar los archivos según su utilidad y que a su vez permita una organización eficaz de los mismos, lo que permite al módulo estar organizado de una manera jerárquica como se puede apreciar en la figura G-1.

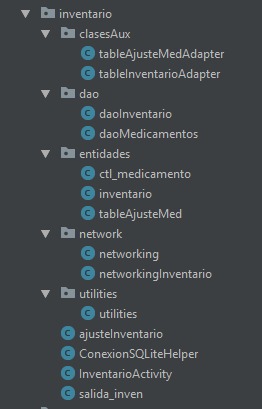


Figura G.1. Estructura del módulo en Android Studio.

Las clases del módulo se agrupan en conjuntos según su funcionalidad, estas agrupaciones están creadas en paquetes, estos son los siguientes: clasesAux, dao, entidades, network, y utilities, a su vez incluye 4 clases que llamadas: ajusteInventario, ConexionSQLiteHelper, MainActivity y salida\_inven.

El paquete clasesAux agrupa los adaptadores que permiten el llenado de las tablas de ajuste de medicamentos, consumo de medicamentos e inventario para que se muestren en la vista, las clases que contiene este paquete son las siguientes:

* tableAjusteMedAdapter: esta clase se encarga de mostrar los datos que se requieren para la vista previa de operación de ajuste de inventario, el método public View getView es el encargado de llenar la tabla adapter con los datos para que sean mostrados en la vista.
* tableInventarioAdapter: esta clase sirve para con la información que corresponde del inventario de la ambulancia, es decir muestra en la vista la descripción del medicamento acompañado de la fecha de vencimiento y la cantidad que se dispone del mismo en la ambulancia, el método encargado de realizar dicha función es public View getView, que se encarga de recopilar la información de la base de datos y a su vez llena la tabla adapter para mostrar los resultados en la vista.

El paquete “dao” contiene las clases que se encargan de las funciones que realizará el sistema respecto a la base datos, este paquete contiene las siguientes clases:

* daoInventario: Esta clase contiene métodos que permiten realizar diferentes funciones con respecto al inventario de la ambulancia en la base de datos, está cuenta con métodos como:

1. public ArrayList<inventario> getInventario(): este método realiza la lectura y obtención de datos del inventario de la ambulancia; esta información es retornada en un ArrayList de tipo inventario, los campos que obtiene de la tabla inventario son: el id del medicamento, la fecha de vencimiento y la cantidad .
2. public ArrayList<inventario> getInventarioFilter(ArrayList<String> ids): este método recibe como parámetro una lista de id de medicamentos que desea leer y obtener de la base de datos, los datos que obtendrá de cada medicamento son: el id, la fecha de vencimiento y la cantidad, cada medicamento será almacenado en otra la lista que será la que devuelva este método.
3. public void insertInventarioFromArray(ArrayList<inventario> inventarios): este método ingresa nuevo inventario en la tabla inventario de la base de datos por medio de la lista de tipo inventarios que recibe.

* daoMedicamentos: Esta clase contiene métodos que permite manipular el medicamento de la base de datos, algunos de estos métodos son los siguientes:

1. public ctl\_medicamento getMedicamentoById(String filter): este método obtiene un objeto de tipo medicamento de la base de datos, esto se logra gracias a que recibe el id del medicamento por parámetro.
2. public void JSONFromFiles(Context m): este método lee un archivo TXT tipo JSON y lo convierte en un objeto JSON.
3. public void insertMedicamentosJSONFILE(): este método ingresa los medicamentos en la tabla medicamentos de la base de datos, estos son obtenidos previamente según el archivo RAW de MINSAL, que previamente fue convertido en JSON en el método anterior.

En el paquete entidades contiene clases que emulan las entidades de la base de datos, estas clases son descritas como se puede ver a continuación:

* ctl\_medicamento: Esta clase describe y contiene los campos pertenecientes a un medicamento, es decir a la entidad “medicamentos” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el módulo.
* inventario: Esta clase describe y contiene los campos pertenecientes al inventario, es decir a la entidad “inventario” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el módulo.
* tableAjusteMed: Esta clase permite la manipulación de los datos de esta clase, esto ayuda a que el manejo de estos datos en el módulo sea de manera más sencilla.

El paquete network contiene dos clases que permiten el manejo de envío de datos al SICIA, y a su vez obtener información. Esas clases son las siguientes:

* networking: esta clase se encarga del envío de datos al Web Service de ajuste de inventario y salida de inventario, algunos métodos vitales para su funcionamiento son los siguientes:

1. public void alertError(String titulo, String mensaje): este método muestra un mensaje en pantalla de carácter exitoso.
2. protected void onPreExecute(): este método crea una barra de progreso para verificar el estado de la operación enviada al Web Service.
3. protected void onPostExecute(String s): Si la operación se realizó con éxito entonces la vista se actualizara, en caso suceda algún error de conexión, la acción se cancelará de inmediato y se mostrará un mensaje de error en la pantalla.
4. protected String doInBackground(String... urls): este método envía los datos desde la vista donde se mandó a llamar el Web Service hacia el Web Service.

* networkingInventario: Esta clase se utiliza para obtener el inventario actual desde la base de datos de postgreSQL, los métodos más importantes de esta clase son los siguientes:

1. public void alertError(String titulo, String mensaje): este método muestra un mensaje en pantalla de carácter exitoso.
2. public void alertErrorConexion(String titulo, String mensaje): este método se utiliza para para mostrar mensajes de errores de conexión.
3. protected void onPreExecute(): este método crea una barra de progreso para verificar el estado de la operación.
4. protected void onPostExecute(String s): Si la operación se realizó con éxito entonces la vista se actualizara, en caso suceda algún error de conexión, la acción se cancelará de inmediato y se mostrará un mensaje de error en la pantalla.

* ajusteInventario: esta clase esta la carpeta principal del módulo, esta clase muestra la información necesaria en pantalla para realizar un ajuste del inventario, sus principales métodos son los siguientes:

1. public void addMedicamentoToTable(View view): este método permite agregar un medicamento en el detalle del ajuste, para que el usuario pueda verificar la descripción del medicamento, la cantidad que se descontará del medicamento, la nueva cantidad con la que contará el medicamento en el inventario.
2. public static boolean isParsable(String input): este método verificará que haya ingresado algún dato en el campo correspondiente a la cantidad de medicamento a desechar.
3. private ArrayList<String> getMedicamentosSpinnerFromInv(ArrayList<inventario> inventarios): Este método permite llenar el combo box donde se mostrarán los medicamentos que se podrán ajustar, estos datos los obtendrá desde la base de datos.
4. private ArrayList<String> getMedicamentosSpinner(ArrayList<ctl\_medicamento> medsList): este método permite obtener el medicamento que ha sido seleccionado en el combo box, y a devuelve la información del medicamento en pantalla.
5. public static void hideSoftKeyboard(Activity activity): este metodo permite que el teclado se vuelva solo númerico cuando se ingresará la cantidad de medicamento que se desea desechar.
6. public void AceptarOperation(View view): este método ejecuta el ajuste que se realizará al inventario.

* ConexionSQLiteHelper: esta clase se extiende de la clase abstracta SQLiteOpenHelper, que sirve para manipular una base de datos en Android, esta clase contiene métodos que permiten la creación de las tablas de la base de datos (onCreate) y que, al actualizar la versión de la base de datos, se eliminen las tablas anteriores y se creen nuevamente para no generar conflicto en la base de datos (onUpgrade).
* MainActivity: esta clase permite mostrar el inventario actual de la ambulancia, algunos de los métodos que utiliza para ello son los siguientes:

1. protected void onCreate(Bundle savedInstanceState): este método permite mostrar el inventario proveniente del WebService.
2. public void sendMessage(View view): este método redirige la vista hacia ajustes de inventario por incidencia.
3. public void ingConsumo(View view): este método redirige la vista hacia el ingreso de consumo de ingreso de consumo de medicamentos.
4. public void updateInventario(View view): este método permite la actualización del inventario y refresca la vista.
5. boolean chkStatus(): este método verifica que exista conexión de red wifi o de datos.

* salida\_inven: Esta clase cuenta con métodos muy similares a los de la clase ajusteInventario, con la única excepción que en este caso se ingresará un campo menos, ya que para este caso no se ingresará la justificación, ya que esta es la vista de ingreso de consumo de medicamento, porque no es necesario una justificación del ajuste.
  1. **Sistema de control de inventario de ambulancias**

Debido al desarrollo del Sistema de control de inventario de ambulancias (SICIA) se crearon diferentes paquetes para agrupar los archivos creados según su utilidad, además esto permite que la organización de una manera ordenada y fácil de entender.

El Sistema de Consulta de Inventario de Ambulancias, está organizado en 10 paquetes, que contienen nombres asociados a un nombre distintivo, tal y como se puede ver en la figura G.2.



Figura G.2. Estructura del proyecto en Eclipse IDE.

Las clases del SICIA se organizan en los siguientes paquetes: controller, converters, dao, dto, resolver, shiro.realm, shiro.faces, utils, y web que se compone del paquete service, que a su vez está compuesto por dos paquetes: dto y ws.

El primer paquete se llama “controller”, este paquete tiene las clases controladoras del sistema, las clases son las siguientes: AssignInvetoryController, CheckInventoryController, CheckInventoryOutputsController y LoginController, estas clases están descritas a continuación:

|  |  |  |
| --- | --- | --- |
| **private** **void** onload() | | |
| **Nombre del método** | onload | |
| **Descripción** | Mediante este método se recuperan todas las ambulancias disponibles en la base de datos, al momento de cargar la página. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

* AssignInventoryController: Esta clase se encarga del controlador que permite asignar un inventario a una ambulancia, siempre y cuando está no se le haya asignado previamente uno. Esta clase posee tres métodos que son los más relevantes e importantes para el funcionamiento de la mismas, estos métodos son: private void onload(), public List<CtlAmbulancia> getAmbulancesNIList() y public void CreateInventory(CtlAmbulancia ctlAmbulancia), estos métodos son descritos con más detalle a continuació

|  |  |  |
| --- | --- | --- |
| **public** List<CtlAmbulancia> getAmbulancesNIList() | | |
| **Nombre del método** | GetAmbulancesNIList | |
| **Descripción** | Este método realiza dos funciones: la primera de verificar si se cuenta con alguna ambulancia en los registros locales; en caso no haya ninguna, está va a la base de datos y a recuperar las ambulancias, la segunda función es la de filtrar las ambulancias que no posean inventario asignado. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | List<CtlAmbulancia> |
| Descripción | Retorna una lista de ambulancias que no tienen asignado un inventario. |

|  |  |  |
| --- | --- | --- |
| **public** **void** CreateInventory(CtlAmbulancia ctlAmbulancia) | | |
| **Nombre del método** | CreateInventory | |
| **Descripción** | Asigna inventario a la ambulancia que recibe como parámetro el método. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | ctlAmbulancia | CtlAmbulancia | Contiene la información obtenida desde la base de datos de cierta ambulancia. | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

* CheckInventoryController: Esta clase se encarga del controlador que permite mostrar las diferentes ambulancias disponibles para poder consultar su respectivo inventario, esta clase cuenta con un método que es muy importante para la realización de esta tarea, este método es: public void onload().

|  |  |  |
| --- | --- | --- |
| private void onload() | | |
| Nombre del método | onload | |
| Descripción | Mediante este método se recuperan todas las ambulancias disponibles en la base de datos, al momento de cargar la página, para ser mostradas en el combo box. | |
| Parámetros Recibidos | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| Valores de Retorno | Tipo | Void |
| Descripción | -- |

* CheckInventoryOutputsController: Esta clase se encarga del controlador que permite consultar las operaciones de inventario en diferentes ambulancias, para ello son requeridos algunos métodos que permite la realización de estas tareas como lo son: private void onload() y public void filtrarOperaciones(), a continuación se detallan los métodos mencionados:

|  |  |  |
| --- | --- | --- |
| public void onload() | | |
| **Nombre del método** | onload() | |
| **Descripción** | Este método se encarga de cargar los datos de las ambulancia desde la base de datos, además de cargar la información de los demás componentes de la página como por ejemplo las fechas. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public void filtrarOperaciones() | | |
| **Nombre del método** | filtrarOperaciones() | |
| **Descripción** | Este método se encarga de aplicar los parámetros necesarios para realizar la búsqueda según los filtros que han sido seleccionados por el usuario. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public EntityManager getEm() | | |
| **Nombre del método** | getEm() | |
| **Descripción** | Crea un EntityManager que permite el acceso a la información de la base de datos. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

* LoginController: Esta clase se encarga de manejar el controlador que a su vez se encarga del manejo de la sesión del usuario, es decir, validar las credenciales que ingresa en la página de inicio de sesión., algunos método importantes para la realización de esta tarea son: public EntityManager getEm(), public String loginShiro(), public String logout(), public String MD5(String md5) y public GlbUsuario getUser(String userName)

|  |  |  |
| --- | --- | --- |
| public String loginShiro() | | |
| **Nombre del método** | LoginShiro() | |
| **Descripción** | Se encarga de verificar la autenticación de los datos ingresados en el inicio de sesión. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | String |
| Descripción | Retorna un cadena, cuyo contenido depende del resultado de la autenticación, es decir la cadena contiene la redirección hasta la página principal o la página de inicio de sesión. |

|  |  |  |
| --- | --- | --- |
| public String logout() | | |
| **Nombre del método** | logout() | |
| **Descripción** | Sirve para eliminar la sesión actual y hace una redirección a la página de inicio de sesión. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | String |
| Descripción | Retorna un string con la ruta de redirección a la página de inicio de sesión. |

|  |  |  |
| --- | --- | --- |
| public String MD5(String md5) | | |
| **Nombre del método** | MD5(String md5) | |
| **Descripción** | Recibe una cadena con el fin de ser encriptada con el algoritmo de encriptación MD5. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | md5 | String | Cadena recibida con el fin de ser codificada. | | |
| **Valores de Retorno** | Tipo | String |
| Descripción | Retorna la cadena ya encriptada o caso de un fallo retorna null. |

|  |  |  |
| --- | --- | --- |
| public GlbUsuario getUser(String userName) | | |
| **Nombre del método** | getUser(String userName) | |
| **Descripción** | Este método se encarga de verificar su el nombre del usuario ingresado existe en la base datos. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | userName | String | Nombre de usuario ingresado desde teclado. | | |
| **Valores de Retorno** | Tipo | GlbUsuario |
| Descripción | Retorna los datos del usuario de la base de datos que coinciden con el nombre de usuario ingresado con el fin de realizar la autenticación |

El paquete “converters” este compuesto solamente por una clase: “AmbulanceConverter”, cuyas especificaciones se describen en el siguiente apartado:

* AmbulanceConverter: Esta clase se encarga de proveer el mecanismo para obtener una ambulancia como un objeto, con el fin de facilitar la manipulación de los mismos datos, esta contiene métodos para asegurar el éxito de su función como por ejemplo: public List<CtlAmbulancia> findAll(), public Object getAsObject(FacesContext context, UIComponent, component, String submittedValue) y public String getAsString(FacesContext context, UIComponent component, Object modelValue). Estos métodos son descritos en los siguientes apartados:

|  |  |  |
| --- | --- | --- |
| public List<CtlAmbulancia> findAll() | | |
| **Nombre del método** | findAll | |
| **Descripción** | Permite obtener el listado de las ambulancias desde la base de datos. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | List<CtlAmbulancia> |
| Descripción | Lista de todas las ambulancias disponibles de la base de datos. |

|  |  |  |
| --- | --- | --- |
| public Object getAsObject(FacesContext context, UIComponent component, String submittedValue) | | |
| **Nombre del método** | getAsObject | |
| **Descripción** | Método utilizado para definir la conversión de los datos, desde la interfaz de convertidor. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | context | FacesContext | Tiene el contexto de la aplicación para ver de qué página se hizo la selección | | component | UIComponent | Componente cuyos datos necesitan de conversión. | | submittedValue | String | Valor por el que se obtendrá el objeto, en este caso es el id de la ambulancia que se utiliza para obtener el objeto ambulancia. | | |
| **Valores de Retorno** | Tipo | Object |
| Descripción | Objeto resultante con la información necesaria de la ambulancia. |

|  |  |  |
| --- | --- | --- |
| public String getAsString(FacesContext context, UIComponent component, Object modelValue) | | |
| **Nombre del método** | getAsString | |
| **Descripción** | Define la conversión de los datos de la vista de modelo a la vista de presentación. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | context | FacesContext | Tiene el contexto de la aplicación para ver de qué página se hizo la selección | | component | UIComponent | Componente cuyos datos necesitan de conversión. | | modelValue | Object | Es el objeto ambulancia. | | |
| **Valores de Retorno** | Tipo | String |
| Descripción | Retorna una cadena vacía si el valor de entra del Objeto es nulo, si el objeto no es nulo entonces retorna el id de la ambulancia. |

El paquete “dao” contiene las clases que se encargan de las funciones que realizará el sistema respecto a la base datos, contiene 4 distintas clases que se describen de la siguiente manera:

* AbstractDao: Esta clase abstracta contiene métodos que son utilizados por otras clases, pero adaptadas a la necesidad de estas, estos métodos son: public void create(T entity), public void update(T entity), public void remove(Integer entityId), public void remove(T entity), public T find(Integer id) y public List<T> findAll(), la descripción de estos métodos se detalla a continuación:

|  |  |  |
| --- | --- | --- |
| public void create(T entity) | | |
| **Nombre del método** | create | |
| **Descripción** | Crea un registro en la base de datos de la entidad que recibe como parámetro. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | entity | T | Esta será la entidad que reciba el método, donde “T” será cargada con el tipo de dato de entity. | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public void update(T entity) | | |
| **Nombre del método** | update | |
| **Descripción** | Actualiza un registro en la base de datos de la entidad que recibe como parámetro. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | entity | T | Esta será la entidad que reciba el método, donde “T” será cargada con el tipo de dato de entity. | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public void remove(Integer entityId) | | |
| **Nombre del método** | remove | |
| **Descripción** | Elimina un registro en la base de datos de la entidad utilizando el parámetro que recibe. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | entity | Integer | Id de la entidad que se eliminará de la base de datos. | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public void remove(T entity) | | |
| **Nombre del método** | remove | |
| **Descripción** | Elimina el registro en la base de datos de la entidad que recibe como parámetro. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | entity | T | Esta será la entidad que reciba el método, donde “T” será cargada con el tipo de dato de entity. | | |
| **Valores de Retorno** | Tipo | void |
| Descripción | -- |

|  |  |  |
| --- | --- | --- |
| public T find(Integer id) | | |
| **Nombre del método** | find | |
| **Descripción** | Método que permite la búsqueda de una entidad en la base de datos, usando el id como parámetro de búsqueda. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | id | Integer | Id de la entidad que se buscará en la base de datos | | |
| **Valores de Retorno** | Tipo | T |
| Descripción | Donde “T” depende del tipo de entidad que sea encontrada en la base datos. |

|  |  |  |
| --- | --- | --- |
| public List<T> findAll() | | |
| **Nombre del método** | findAll | |
| **Descripción** | Método que permite la búsqueda todos los registros de cierto tipo en la base de datos. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | -- | -- | -- | | |
| **Valores de Retorno** | Tipo | List<T> |
| Descripción | Lista de datos encontrados en la base de datos donde T dependerá del tipo de entidad. |

* AmbulanceDao: Esta clase hereda los métodos de AmbulanceDao(), lo que le permite crear, actualizar, leer o eliminar registros del tipo CtlAmbulancia.
* GlbUsuario: Esta clase hereda los métodos de AmbulanceDao(), lo que le permite crear, actualizar, leer o eliminar registros del tipo GlbUsuario.
* InventarioDao: Esta clase hereda los métodos de AmbulanceDao(), lo que le permite crear, actualizar, leer o eliminar registros del tipo Inventario.

El paquete “dto” contiene clases que emulan las entidades de la base de datos, estas clases son descritas como se puede ver a continuación:

* CtlAmbulancia: Esta clase describe y contiene los campos pertenecientes a una ambulancia, es decir a la tabla “ctl\_ambulancia” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.
* CtlMedicamento: Esta clase describe y contiene los campos pertenecientes a un medicamento, es decir a la tabla “ctl\_medicamento” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.
* DetalleInventario: Esta clase describe y contiene los campos pertenecientes a un invenatario, es decir a la tabla “detalle\_inventario” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.
* GlbUsuario: Esta clase describe y contiene los campos pertenecientes a un usuario, es decir a la tabla “gbl\_usuario|” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.
* Inventario: Esta clase describe y contiene los campos pertenecientes a un inventario, es decir a la tabla “inventario|” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.
* Operacion: Esta clase describe y contiene los campos pertenecientes a una operación de inventario, es decir a la tabla “operacion|” de la base de datos, esto facilita el almacenamiento y entrega de los datos recopilados por la misma en el servicio web.

El paquete “resolver” contiene la siguiente clase:

* SiciaResolver: Esta clase es utilizada para proveer un EntityManager y sigue el patrón “singleton” para garantizar la existencia de una sola instancia de esta clase y que su uso sea global en todo el proyecto.

El paquete “shiro” provee el proyecto de algunas funciones de seguridad brindadas por el framework Apache Shiro, utilizado en el proyecto, la siguiente clase contiene más información:

* SiciaRealm: Esta clase es utilizada por el framework de Apache Shiro y provee los métodos para autenticación, autorización, criptografía y administración de sesión dentro del servidor, esta clase contiene un método esencial para el manejo de la sesión del usuario: protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authcToken), la descripción del método se hace a continuación:

|  |  |  |
| --- | --- | --- |
| protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authcToken) | | |
| **Nombre del método** | doGetAuthenticationInfo | |
| **Descripción** | Recibe un token de autenticación y se encarga de hacer la verificación de las credenciales que hace el framework. | |
| **Parámetros Recibidos** | |  |  |  | | --- | --- | --- | | Nombre | Tipo | Descripción | | authcToken | AuthenticationToken | Token para la comprobar la autenticación del usuario. | | |
| **Valores de Retorno** | Tipo | AuthenticationInfo |
| Descripción | Si las credenciales son correctas se retorna un objeto que garantiza la creación de la sesión del usuario, pero sino no son correctas entonces retorna null. |

El paquete “shirofaces” provee de clases de seguridad de Apache Shiro, una de ellas es la que se describe a continuación:

* FacesAjaxAwareUserFilter: Esta clase solamente sirve para la configuración necesaria para el manejo de la verificación del usuario, es decir la redirección que se hará después que el usuario haya ingresado sus credenciales.

El paquete “utils” cuenta con la clase encargada de proveer los diferentes mensajes que el usuario encontrará en el proyecto, la clase utilizada es la siguiente:

* JsfMessages: Esta clase provee los mecanismos para generar mensajes al usuario mientras este interactúa con la aplicación web, estos mensajes pueden ser de éxito, advertencia o errores en la realización de operaciones en la aplicación.

El paquete “web” se compone del paquete “service”, este paquete se compone de dos paquetes más, el primero es “dto” que está compuesto por las siguientes clases:

* InventoryDetailsDto: Es una clase que es utilizada para enviar información del detalle de inventario a través de los Web Service, es decir en ella están definidos los campos que pertenecen al inventario.
* InventoryDetailsRequest: Esta clase es utilizada para crear peticiones y obtener detalles de un inventario a partir de id de la ambulancia.
* InventoryDetailsResp: Esta clase contiene la información del detalle del inventario de una ambulancia cuya petición se hace desde la aplicación móvil.
* Operation: Esta clase es utilizada para obtener el detalle de una operación, ya sea entrada o salida de inventario que se realizó en la aplicación móvil
* SaveOperationRequest: Esta clase obtiene el detalle de las operaciones que se desean registrar en la base de datos de una determinada ambulancia.
* SinabInventoryRequest: Esta clase hace la petición hacia el webservice del SINAB, que se encarga de entregar el detalle de los vales, que contienen el detalle de la entrega de medicamentos a una determinada ambulancia.
* ValeSinab: Esta clase representa el detalle de los medicamentos que son recibidos por una determinada ambulancia.

El segundo paquete que compone el paquete “service” se llama “ws”, que está compuesto por las clases que permiten el intercambio de datos con el Web Service del SINAB, y la comunicación con la aplicación móvil SAPHCA, estas clases son:

* InventoryDetailsWS: Esta clase representa la definición del Web Service que se encarga de recibir y procesar las peticiones de detalle de inventario realizadas desde la aplicación móvil y de enviar la información del detalle del inventario de acuerdo a la ambulancia que hizo la petición.
* SaveOperationWS: Esta clase representa la definición del Web Service que se encarga de recibir y procesar las peticiones de entradas o salidas de inventario en la base de datos.
* SinabInventoryWS: Esta clase se encarga de guardar en tabla operaciones de la base de datos, las entradas de inventario generadas a partir del WebService del SINAB, esta su vez registra en la base de datos el detalle del inventario ingresado para después poder ser asignado a la ambulancia que le corresponde.

MÓDULO DE CONTROL DE INVENTARIO DE AMBULANCIAS PARA LA APLICACIÓN MÓVIL SAPHCA

Masin, René., Pleítez, César., Polanco, José., Rodríguez, Jesús., Varela, Erick.

Departamento de Electrónica e Informática, Universidad Centroamericana “José Simeón Cañas”, El Salvador evarela@uca.edu.sv

***Abstract―* La implementación del uso de la tecnología en las ambulancias del Sistema de Emergencias Médicas (SEM) ha sido un gran avance, ya que, gracias al uso de la misma, se ha logrado implementar el uso los dispositivos móviles en las ambulancias, el uso de estas es debido a que estos dispositivos manejan una aplicación móvil llamada SAPHCA, cuyo propósito principal agilizar la atención de una emergencia, ya que, por ejemplo, con el uso de la aplicación, al ser realizada una llamada por parte de una víctima o paciente, esta es atendida por el centro de llamadas de emergencia del SEM y se toman los datos del paciente, ubicación, entre otros; estos son registrados y se envía una solicitud de atención de emergencia a alguna ambulancia que esté disponible, por supuesto esta solicitud es recibida por medio del dispositivo móvil, específicamente en la aplicación móvil, al recibir la solicitud la ambulancia se traslada de inmediato a atender la emergencia, siempre y cuando no tenga alguna solicitud anterior; porque en caso de tener una solicitud anterior, la nueva solicitud pasa a modo de espera. Además, los datos de la víctima o paciente son recibidos por la aplicación móvil, anteriormente se llenaba un formulario físico con estos datos, por lo que se podían suscitar casos en los que dichos registros se extraviarán, pero gracias a la aplicación los registros quedan almacenados con éxito en la misma. De la misma manera, anteriormente cuando una emergencia era atendida los medicamentos utilizados en ella se registraban en un formulario físico, en cambio ahora con el uso la aplicación, estos son registrados en ella, pero no se cuenta con una modulo que sirva como inventario digital en ella, es por eso el motivo de la creación de dicho módulo de control de inventario en la aplicación.**

*Palabras clave*―aplicación móvil, dispositivo móvil, inventario, tecnología.

INTRODUCCIÓN

El avance tecnológico que se ha logrado en los últimos años, encamina a una era digital en la que las antiguas prácticas están quedando cada día más obsoletas, como por ejemplo para la reservación de una cita al momento de tramitar algún documento, se hacía de manera telefónica o en el peor de los casos de manera personal, hoy en día para reservar una cita basta con tener una computadora o algún dispositivo móvil conectado a una red local o inalámbrica para poder realizar dicha cita, al hacer esto los datos son tomados de manera digital y el usuario puede verificar que los datos sean correctos, y estar seguro que no se suscitara algún error, como por ejemplo la persona encargada de tomar los datos podía tener algún error de escritura y este provocar la pérdida de la cita del usuario, además todos estos registros al estar de forma física durante unos diez años, se acumulaba una inmensa cantidad de estos, con la implementación de sistemas que permiten almacenar dichos registros de manera digital, el espacio ocupado por los registros físicos se vio muy reducido.

Otro ejemplo sería el control de inventario de cierta ambulancia, el caso hipotético de una ambulancia que cuenta con un inventario asignado mensualmente, al atender una emergencia se utilizan medicamentos del inventario, una vez se finalizaba de atender la emergencia, se registraba de manera física el uso de los medicamentos estos registros tenían que ser guardados para poder saber la cantidad de medicamento o insumo que quedaban en el inventario, y así con estos reportes se tenían que pasar muchas horas para la elaboración de reportes mensuales. El cambio se presentó cuando la tecnología se empezó a utilizar para estos casos, ya que con la implementación de un sistema que permitiera el control del inventario de forma digital en las ambulancias, estos procesos de minutos se verían reducidos a segundos, con la implementación de herramientas que permiten el desarrollo de estas aplicaciones en dispositivos móviles, ya que ahora el registro del inventario se hace en la aplicación, y en vez de llenar el formulario con los medicamentos utilizados en las emergencias, ahora solo basta con ingresar a la aplicación, seleccionar del inventario actual los medicamentos utilizados y listo, al mismo tiempo que se actualiza el inventario y se ve de manera inmediata la cantidad de inventario que ha quedado.

Con respecto al tiempo utilizado para la elaboración de reportes se ve reducida de horas a minutos, ya que al llevar un registro de los medicamentos utilizados en un mes, solamente bastaría con seleccionar en la aplicación el periodo de tiempo, en este caso un mes y se mostrarían en el dispositivo el inventario utilizado en dicho periodo de tiempo, con esto bastaría que la aplicación tenga implementada una manera de imprimir los resultados mostrados en pantalla para tener un reporte de los medicamentos utilizados en un mes, todo ello en cuestión de un par de minutos.

En la figura 1, se muestra un ejemplo sencillo de un control de inventario en una aplicación diseñada para un dispositivo móvil, en ella se puede observar la identificación de la ambulancia, para llevar un registro por separado para cada una, la descripción del medicamento, acompañado de la fecha de vencimiento y la cantidad disponible, con dos opciones muy necesarias como por ejemplo el ajuste del inventario, ya que se puede suscitar el caso que el medicamento sufra algún incidente por lo que sea necesario dar de baja el mismo, la otra opción aún más vital como lo es el registro de consumo de medicamentos, en la que el usuario registrará la salida de medicamento utilizado en una emergencia, como se puede observar todo esto parece muy sencillo, pero el desarrollo de dicha aplicación también conlleva cierto tiempo de elaboración, después un período de tiempo en el que debe ser probada por los desarrolladores de la misma que intentarán buscar problemas en su funcionamiento con el fin de corregir fallas y que sea lo más confiable posible. Además, se requiere de software especializado para el desarrollo de estas, y herramientas que por ejemplo permitan que los registros que se almacenan en los dispositivos móviles pasen a una computadora central donde estos registros son almacenados, y al obtener estos registros en un solo punto se podrían elaborar registros comparativos del consumo de inventario de todas las ambulancias.

Una de las herramientas para la consumación de este proceso es Rest API, que permite la transferencia de esta información desde un dispositivo móvil a una computadora central haciendo uso de conexión a internet. Toda esta información es manipulada a través de otra herramienta muy importante como lo son las bases de datos, que permite mantener un control centralizado de los datos, además de ofrecer seguridad a los datos y un mejor manejo de los mismos.

*Fig. 1 Ejemplo de un control de inventario de una ambulancia.*

MATERIALES Y MÉTODOS

1. *Requerimientos*

Para el desarrollo del módulo para dispositivos Android es necesario tener instalado:

* Java developer Kit 8(JDK)
* Android Studio
* Android SDK
* Google Chrome

1. *Pasos para el desarrollo*

Para la creación e implantación del módulo de control de inventario se siguieron los siguientes pasos:

* Crear un proyecto nuevo en Android Studio.
* Agregar al archivo Gradle las dependencias las cuales son las siguientes:

'com.android.support:appcompat-v7:27.1.1'  
'com.android.support.constraint:constraint-layout:1.1.1'  
'junit:junit:4.12'  
'com.android.support.test:runner:1.0.2'  
'com.android.support.test.espresso:espresso-core:3.0.2'

*Fig. 2 Dependencias que se deben agregar al archivo Gradle*

En el proyecto, crear una clase nueva para el manejo de la Base de datos dentro del módulo de inventario, la clase debe extenderse de la interfaz SQLiteOpenHelper, para el desarrollo del módulo se usarán dos tablas, una para una lista completa de medicamentos, y otra para el inventario especifico de cada ambulancia.

En el proyecto, crear las vistas necesarias para el funcionamiento de la aplicación, en este caso se usaron 3 vistas principales, una vista Main, una vista para Ajuste de inventario y otra para Ingreso de consumo de medicamentos.

Crear las clases que hagan referencia a las entidades en la base de datos SQLite, en este caso, medicamento e inventario.

Crear las clases que servirán como métodos para las entidades en la base de datos. Dentro de estas clases, deben elaborarse un método el cual convertirá una respuesta en formato JSON proveniente del Web Service a las entidades creadas anteriormente.

Crear una clase el cual servirá como controlador para establecer la conexión entre el Web Service y la aplicación Android, la clase deberá extenderse de la interfaz AsyncTask. Es necesario asegurarse que los siguientes métodos de la interfaz AsyncTask deben ser implementados para lograr un resultado óptimo:

* onPreExecute(): Para mostrar al usuario una barra de progreso mientras se realiza el consumo del web service.
* doInBackground(): En este método se obtendrá la respuesta proveniente del Web Service.
* onPostExecute(): para manejar la respuesta proveniente del Web Service.

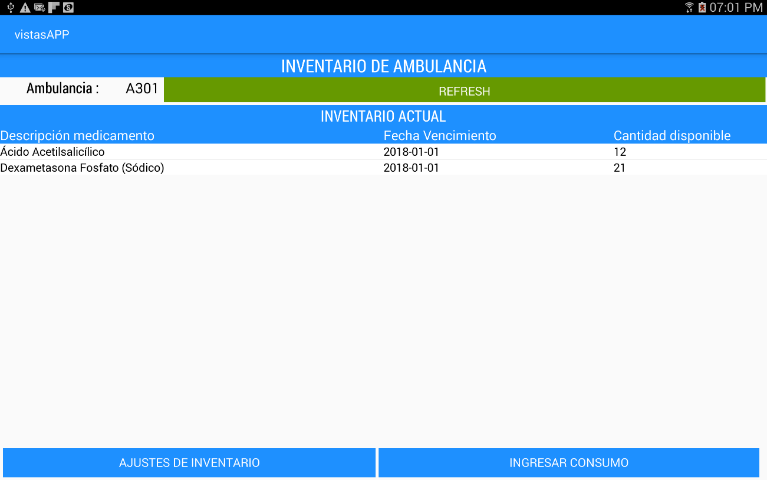
Dentro de este método se maneja otro adicional el cual es el siguiente:

* onFinish(): Si la respuesta al Web Service fue exitosa, se ejecutará u obtendrá el resultado de la operación realizada.

Desde las vistas principales, se debe hacer la llamada a la clase creada para el controlador de la red, indicándole la URL del servidor web donde se encuentra el Web Service, y alguno de los siguientes métodos de la API:

* SaveOperation: Recibe una lista en formato JSON de la operación realizada, envía como respuesta un código de éxito o error, dependiendo de la ejecución del método.
* InventoryDetails: Recibe un Sting equivalente al número de identificación de la ambulancia, envía como respuesta un código de éxito, y una lista de la entidad inventario en formato JSON, o bien, un código de error y una lista vacía.

RESULTADOS Y DISCUSIÓN



*Fig. 3. Vista principal del módulo, nombrada Main.*

Con la creación de clase Main, que maneja la pantalla principal del módulo de consulta de inventario se obtiene algo parecido a la vista de la Figura 3, pero en este caso se la han dado unos detalles visuales, para ser más agradable a la vista del usuario, esta pantalla despliega el inventario actual que posee la ambulancia, los resultados se muestran como una tabla, en la que los datos que se brindan son: Descripción de medicamento, fecha de vencimiento y cantidad disponible en el inventario, además a esta vista se le agrego un botón que se llama refresh, este botón permite actualizar los datos del inventario.



*Fig. 4 Función onCreate() de la clase Main..*

El método **onCreate** de la clase Main, permite obtener los datos del inventario de la ambulancia, todo esto gracias a que este método se conecta a través de una red inalámbrica al Web Service que provee los datos del inventario, en caso no haya conexión de red, le muestra un mensaje de error al usuario, indicando en él que el dispositivo necesita estar conectado a una red para poder obtener dicha información, que se conecte a una red y lo vuelva a intentar otra vez.

Para realizar la verificación si el dispositivo móvil está conectado o no a una red, existe un método nombrado **chkStatus**, que se encarga de esa función, para ello verifica los parámetros de red de la conexión de tipo Wifi o tipo mobile, esta pasa por una condicional en la que se pregunta si el dispositivo está conectado a una red wifi o red móvil, en caso no estar conectado a ninguna de las dos, entonces devuelve un valor falso que manda la advertencia al usuario que no hay conexión de red.

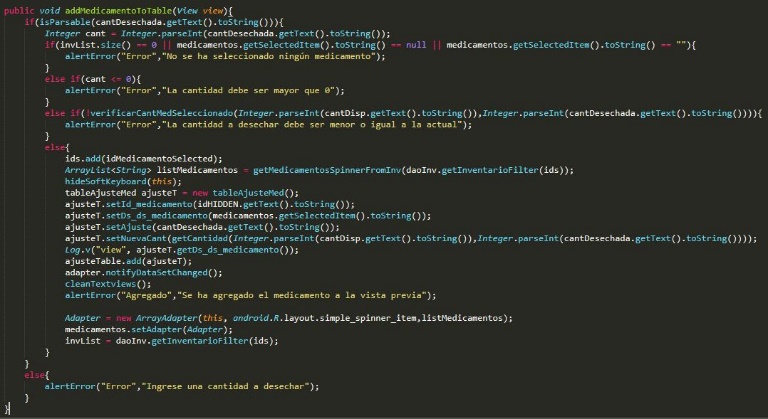
*Fig. 5. Método chkStatus() de la clase Main.*

El resultado de la creación de la clase que permite controlar la vista de Ajuste de inventario, se obtendría algo parecido a la vista de la figura 6, siempre exceptuando los detalles que se han dado para que visualmente sea más atractiva y acorde a la configuración visual que tiene la aplicación principal, la cual se le creo un combo box encargado de desplegar todos los medicamentos existentes en el inventario, un botón agregar que permite agregar a detalle de ajuste una visualización del medicamento, el consumo y la nueva cantidad que quedaría en el inventario, además de un botón que permite eliminar la selección en caso haya un error.



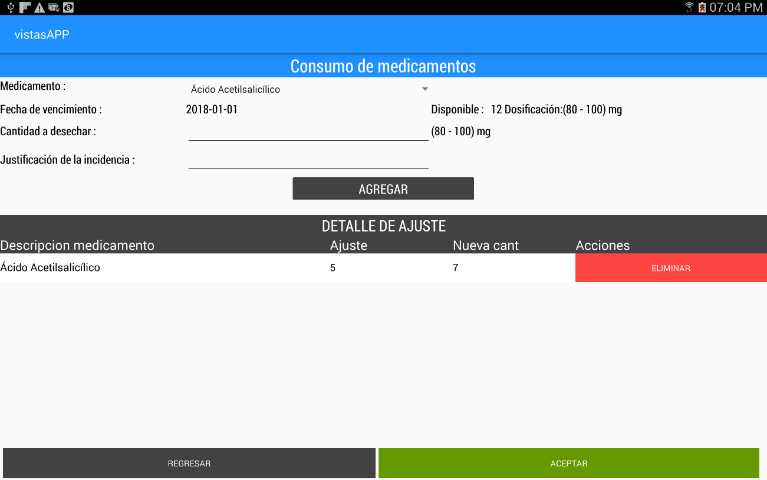
*Fig. 6. Visualización previa del resultado de la vista de ajuste de inventario.*

Al seleccionar el usuario el botón de agregar, este tiene un método que se encarga de los datos que desea eliminar sean correctos, en la figura 7 se puede apreciar el método **addMedicamentoToTable** encargada de esa comprobación.



*Fig. 7. Clase addMedicamentoToTable encargada de comprobar que los datos que se desean eliminar sean correctos.*

La tercera vista creada es la de ingresar un consumo de medicamentos, que se utiliza cuando se atendió una emergencia médica y se hizo consumo de ciertos medicamentos, por los que se tienen que dar de baja en el inventario. A diferencia de la vista creada para el ajuste de inventario, esta no posee el campo de justificación. La clase encarga de controlar esta vista es **salida\_inven,** esta vista se puede apreciar en la figura 8.

****

*Fig. 8. Visualización previa del resultado de la vista de ingreso de consumo.*

CONCLUSIONES

* Se adquirió el conocimiento necesario acerca de las herramientas que se utilizaron para la realización del proyecto y se realizó un estudió a profundidad para el entendimiento del código fuente de la aplicación, con el fin de que el módulo de control de inventario de ambulancias del SEM siguiera los estándares de programación que posee el código ya escrito de la aplicación.
* Se desarrolló un módulo para el sistema SAPHCA del MINSAL, llamado módulo de control de inventario para ambulancias, la implementación de este módulo hace hincapié en el buen uso que se le puede dar a la tecnología, ya que el proceso pasó de ser manual a digital con lo se podrá llevar a cabo un mejor control de los insumos y medicamentos que utiliza cada ambulancia del SEM; esto da paso a que se pueda llevar un registro en ciertos períodos de tiempo y hacer proyecciones de la cantidad de medicamentos e insumos que se requieren en cada ambulancia

REFERENCIAS

1. (2015) Consumir un servicio web Rest desde Android [online]. Available: http://www.hermosaprogramacion.com/2015/12/consumir-un-servicio-web-rest-desde-android/
2. (1999) Introducción a JSON [online]. Avalilable:

https://www.json.org/json-es.html

1. (2018) SQLiteOpen Helper [online]. Available: https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper