



UNIVERSIDAD AMERICANA DE ACAPULCO

“EXCELENCIA PARA EL DESARROLLO”

FACULTAD DE INGENIERÍA EN COMPUTACIÓN

INCORPORADA A LA UNIVERSIDAD NACIONAL

AUTÓNOMA DE MÉXICO

CLAVE DE INCORPORACIÓN 8852-16

**“SISTEMA DISTRIBUIDO APLICADO A
UN CONTROL DE INVENTARIO”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTAN

**MELINA CASTAÑEDA PÍREZ
SILVERIO QUIÑONEZ GARCÍA**

DIRECTOR DE TESIS

DR. RENÉ EDMUNDO CUEVAS VALENCIA



ACAPULCO, GUERRERO JUNIO 2017.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A **Dios** porque Él da la sabiduría, y de su boca viene el conocimiento y la inteligencia. Mi sustento diario durante toda la licenciatura y hasta ahora.

A **mis Padres**, mi impulso para cumplir las metas, por estar conmigo a través de todo este proceso y brindarme su apoyo incondicional.

A la **Universidad Americana de Acapulco**, por ser mi segunda casa por más de 4 años y contribuir en mi desarrollo profesional.

A la **Ing. Eloisa Mercedes Vivas Villasana** por el apoyo recibido de su parte durante toda la licenciatura y por supervisar este trabajo de tesis.

Al **Dr. René Edmundo Cuevas Valencia** por aceptar ser mi asesor, las facilidades y aliento que ha dado para culminar este trabajo de titulación.

A los **profesores** de la Facultad de Ingeniería, por su aportación y los conocimientos compartidos para mi desarrollo profesional.

Al **Ing. Alejandro Mendoza Viveros**, por su apoyo y las facilidades ofrecidas para realizar exitosamente este proyecto.

Al **Ing. Jorge Ortiz García**, por sus constantes asesorías durante la realización de este proyecto.

A **Silverio Quiñonez García**, por ser parte de este proyecto y estar en los momentos más críticos, pero también en los más dichosos que se han tenido durante la realización de este proyecto.

Melina Castañeda Pérez

AGRADECIMIENTOS

A **Dios** porque de Él, y por Él, y para Él, son todas las cosas, quien me ha dado la sabiduría y la fortaleza todos los días de mi vida hasta este momento.

A **mis Padres**, por acompañarme en mi formación, por ser la base de mis valores, por brindarme su apoyo incondicional y así ayudándome a cumplir mis sueños.

Al **Dr. René Edmundo Cuevas Valencia** por el haber aceptado asesorarme, por el apoyo que me ha dado y así haber concluido con este trabajo de titulación.

A la **Ing. Eloisa Mercedes Vivas Villasana** por el apoyo durante los 4 años de la carrera, por supervisar este trabajo de titulación

A los **profesores de la Facultad de Ingeniería**, por dedicar su tiempo para aportar nuevos conocimientos.

Al **Ing. Alejandro Mendoza Viveros**, por su apoyo y las facilidades ofrecidas para realizar exitosamente este proyecto.

Al **Ing. Jorge Ortiz García**, por sus constantes asesorías durante la realización de este proyecto.

A **Melina Castañeda Pirez**, por la contribución al proyecto, por la motivación para finalizar este proyecto aún en las circunstancias más difíciles.

Silverio Quiñonez García

DEDICATORIA

A **mis Padres, mi Hermano y mis Abuelos**, por estar apoyándome en cada momento hasta el final de la realización de este proyecto.

A **Alejandro Castañeda Mariano**, por estar en cada momento de la duración de la licenciatura apoyándome y ayudándome.

Al **M. C. José Mario Martínez Castro[†]**, por las bases otorgadas en la programación tanto Web como Móvil y por enseñarme el mundo de los sistemas distribuidos.

Melina Castañeda Pírez

DEDICATORIA

A **mis Padres**, porque gracias a ustedes pude finalizar esta etapa de mi vida que me ha marcado por siempre.

A **Simón Quiñones**, por alentarme a terminar mi carrera.

Al **M. C. José Mario Martínez Castro[†]**, por haber dedicado su vida a la enseñanza para las personas que buscaban conocimiento, porque gracias a él obtuvimos las bases para el inicio de este proyecto.

Silverio Quiñonez García

ÍNDICE GENERAL	Página
AGRADECIMIENTOS	2
DEDICATORIA	4
INTRODUCCIÓN	8
PLANTEAMIENTO DEL PROBLEMA	9
JUSTIFICACIÓN	10
HIPÓTESIS	11
OBJETIVO GENERAL	12
OBJETIVOS ESPECÍFICOS	12
CAPÍTULO I. ESTADO DEL ARTE: LA ZONA DE OPERACIÓN DE TRANSMISIÓN GUERRERO – MORELOS	13
1.1 Introducción.....	15
1.2 Historia	16
1.3 Ubicación.....	21
1.4 Misión	22
1.5 Visión.....	22
1.6 Objetivos e Indicadores	22
1.7 Organigrama de la ZOTGM.....	23
1.8 Sistemas actuales	25
CAPÍTULO II. CASO DE ESTUDIO	34
2.1 Estado Actual de la Herramienta.....	35
2.2 Metodologías de Desarrollo.....	44
2.3 Metodología de los Sistemas Distribuidos	46
2.3.1 Especificaciones de alto nivel.	48
2.3.2 Estándares.....	50
2.3.3 Herramientas de Diseño	52
2.3.4 Documentación.....	54

CAPÍTULO III. CONCEPTOS Y DEFINICIONES BÁSICAS	59
3.1 Herramientas para Programación Web	60
3.1.1 Front – End	66
3.1.2 Back – End	73
3.2 Herramientas para Desarrollo de Aplicaciones Móviles	80
3.3 Control de versiones	90
3.3.1 GIT	94
3.3.2 Bonobo GIT Server	97
3.4 Sistemas Distribuidos	98
CAPÍTULO IV. DESARROLLO E IMPLEMENTACIÓN	101
4.1 Especificaciones de Alto Nivel.....	102
4.2 Estándares	107
4.3 Herramientas de Diseño.....	114
4.4 Documentación	149
4.5 Implementación	152
CAPÍTULO V. RESULTADOS Y TRABAJO A FUTURO	168
5.1 Integración de las aplicaciones en el Sistema Distribuido	169
5.2 Trabajo Futuro.....	189
CAPÍTULO VI. CONCLUSIONES	191
REFERENCIAS	194
ÍNDICE DE FIGURAS	199
ÍNDICE DE TABLAS	205
ANEXO A. Cuestionario de especificaciones del sistema.	207
ANEXO B. Diccionario de datos del Activo	210
ANEXO C. Diccionario de datos de la Autenticación	215

INTRODUCCIÓN

El principal objetivo de la ingeniería es resolver problemas y los sistemas de información han resuelto una innumerable cantidad de problemas inmediatamente después del surgimiento de la primera computadora.

Uno de los principales problemas en el área administrativa de las empresas es llevar un control de todos los equipos, instrumentos y materiales que tienen, ya que un error en el control podría ocasionar desde un diminuto hasta un gasto muy extenso y la Zona de Operación y Transmisión Guerrero Morelos (ZOTGM) no es la excepción.

La ZOTGM utiliza una aplicación el cual lleva el control de cada inmueble con los que cuentan desde computadora, video-muros, sillas, teléfonos IP, hasta los archiveros. Dicha aplicación se encarga de almacenarlos (conocidos como activos) que serán de utilidad para la persona encargada de llevar administración de la empresa y de los posibles gastos a realizar en un futuro. Si requiere de un elemento faltante en su área de trabajo o de sus empleados, consulta este sistema y obtiene la información solicitada con prontitud.

Debido a la importancia que recae en este sistema, se decide actualizar toda la estructura y cada uno de los componentes que lo conforma. De igual manera se desarrolla una aplicación móvil para poder realizar operaciones con la información que se almacena en este sistema.

PLANTEAMIENTO DEL PROBLEMA

La tendencia de este siglo es automatizar todos los procedimientos administrativos. Cada vez las empresas están abandonando las prácticas en papel y comienzan a realizar sus procesos a través de un sistema informático. No basta solamente con sistematizar sus procedimientos, sino de darles mantenimiento de manera que el sistema mantenga sus componentes actualizados sin problemas de compatibilidad con el paso de las actualizaciones de las TIC.

Debido a esto, se percibe la necesidad de actualizar el sistema que utiliza la Zona de Operación de Transmisión Guerrero – Morelos como Control de Inventario ya que las operaciones básicas que realiza (insertar, modificar, consultar y eliminar registros) cuentan con carencias que afectan el funcionamiento de éste. De igual manera, existen módulos dentro del sistema los cuales se encuentran incompletos.

En lo que refiere a la interfaz gráfica del sistema, el diseño y posición de los elementos de la página no son los adecuados para que se catalogue como amigable e intuitiva, característica importante que debe de estar presente en el software.

Se debe de realizar la migración de la aplicación a una plataforma más actualizada, ya que en la que se encuentra actualmente implementada próximamente será descontinuada y perderá el soporte técnico que se le ha proporcionado.

JUSTIFICACIÓN

Se aborda este tema de tesis debido a que el diseño e implementación de sistemas de software tanto de base como de aplicación se encuentran dentro de las actividades que desarrollará el egresado de Ingeniería en Computación, de acuerdo al perfil de egreso de la Universidad Nacional Autónoma de México.

Se realizará la migración del Sistema de Control de Inventario a una versión actualizada de la plataforma de desarrollo. Se diseñará nuevo modelo para la Base de Datos que defina de mejor manera la información que se maneja.

Utilizando los nuevos patrones de diseño de Aplicaciones Web, se creará una interfaz atractiva para los usuarios que utilicen este sistema.

Al generar reportes requeridos por el administrador en formatos de Hoja de Cálculo y PDF, se ofrecerá efectividad en este proceso administrativo y al implementar seguridad a través de roles de usuario se otorgará protección a la información almacenada.

Se ha encontrado un área de oportunidad para que este sistema sea extendido de una sencilla aplicación Web a un sistema distribuido añadiendo una aplicación móvil y así mejorar la funcionalidad del sistema dentro de este centro de trabajo.

HIPÓTESIS

Administrar los activos y pasivos de productos, accesorios, y equipos de cómputo de una dependencia federal, usando un sistema de control de inventarios apoyados por plataformas Web y móviles; facilitará la atención a supervisiones externas realizadas en la dependencia.

OBJETIVO GENERAL

Desarrollar un sistema distribuido compuesto por un módulo Web y móvil que opere bajo las nuevas tendencias de diseño y desarrollo de sistemas, mejore la imagen y funcionalidad adquiriendo mayor importancia en los procesos que se realizan.

OBJETIVOS ESPECÍFICOS

- I. Analizar las diferentes metodologías de desarrollo a fin de escoger la que más se adecue al problema a resolver.
- II. Analizar el estado actual del Sistema de Control de Inventarios.
- III. Realizar propuestas de rediseño.
- IV. Investigar componentes del lado del cliente que permita la visualización de información de manera agradable al usuario.
- V. Implementar un Servidor de Control de Versiones que permita el trabajo en conjunto sobre una misma aplicación.
- VI. Desarrollar el nuevo Sistema de Control de Inventarios con características responsivas que favorezcan la funcionalidad desde cualquier dispositivo que deseen acceder, así como la aplicación móvil para éste.

CAPITULO 1: Estado del Arte

CAPÍTULO I. ESTADO DEL ARTE: LA ZONA DE OPERACIÓN DE TRANSMISIÓN GUERRERO – MORELOS

En este capítulo se hará una reseña sobre la historia del centro de trabajo en el que se implementará el Sistema de Control de Inventario, además de describir las funciones que se realizan.

Se abordarán los siguientes puntos:

- Historia
- Misión
- Visión
- Ubicación
- Política
- Objetivos e Indicadores
- Organigrama

La información proporcionada en la presente sección está disponible en el servidor local de la Zona de Operación de Transmisión Guerrero – Morelos (ZOTGM), con la siguiente dirección IP: <https://10.25.117.65>. Las secciones que se tomaron como referencias fueron: “Historia”, “Organigrama”, “Objetivos e Indicadores”, “Misión y Visión” y “¿Quiénes Somos?”.

Se incluirá un apartado sobre los sistemas que existen actualmente para el control de inventario.

1.1 Introducción

La Zona de Operación de Transmisión Guerrero Morelos (ZOTGM) se constituyó de manera oficial el 20 de Septiembre de 1976 según consta en el convenio CFE-SUTERM No. 127-74. Se encuentra ubicada en la ciudad de Acapulco Guerrero y desde su fundación formó parte del Área de Control Oriental. En la Figura 1.1 se muestra la fachada de este centro de trabajo.



Figura 1.1 Fachada de la ZOTGM.

Obtenida de: (ZOTGM, 2016)

Su ámbito de influencia lo ejerce sobre las redes de 230, 115, 85 y 69 Kilovoltios (KV's) en los estados de Guerrero y Morelos. Es operada

por personal altamente calificado y comprometido en el desarrollo de sus actividades.

En la ZOTGM están convencidos de que el desarrollo de los empleados, el estricto cumplimiento de las normas, reglamentos, la satisfacción de sus clientes, el apego a los objetivos y metas de la CFE, así como la búsqueda de la mejora continua de procesos y servicios los acerca cada vez más a su compromiso de ser una empresa de clase mundial.

El Centro Nacional de Control de Energía es el encargado de coordinar, administrar y supervisar la operación del Sistema Eléctrico Nacional (SEN) a través del Centro Nacional, los Centros de Control de Área y Subáreas. Su misión es operar bajo los criterios de seguridad, continuidad, calidad y economía.

Con la interconexión del Sistema Colotlipa-Acapulco al Sistema Interconectado Nacional en el año de 1973, surge la necesidad de crear un organismo encargado de cumplir con estos criterios operativos. Dicha necesidad se vio cubierta con la creación del Despacho de Carga Acapulco. Años más tarde este organismo extendería su ámbito de influencia al territorio nacional ocupado por los estados de Guerrero y Morelos, quienes finalmente darían nombre al centro de trabajo como el que hoy se conoce.

1.2 Historia

Considerada como la primera Subárea de Control en el territorio Nacional, La Subárea de Control Guerrero Morelos se ubica en la ciudad de Acapulco Guerrero. Fue creada a partir de la interconexión del

Sistema Colotlipa-Acapulco en Mayo de 1973 al Sistema Interconectado Nacional.

La historia hablada de la Subárea comenta que surge en el tablero de la subestación “El Quemado”, junto al operador de la misma. Los primeros relatorios de los que se tiene referencia datan de Abril de 1974 y como encargado de la operación el Ing. Andrés Betanzos Reyes.

Se hace oficial su existencia el 20 de Septiembre de 1976 a través del convenio CFE-SUTERM No. 127-76, en donde se constituye de manera oficial el Despacho nacional de carga, así como los Despachos de Carga Occidental, Peninsular, Noroeste, Noreste, Norte, Mexicali y el Despacho de Carga Acapulco (hoy Subárea de Control Guerrero Morelos).

Según consta en este documento, la Subárea de Control fue constituida con una plantilla que consistía de cinco ingenieros operadores de Subárea, una secretaria, un mozo-mensajero-chofer y un dibujante.

Aun cuando en el momento de la firma del convenio, el ingeniero Betanzos fungía como encargado de la operación, no llegaría a ser oficialmente Jefe de Subárea, debido a que realizaría los trámites para su jubilación ese mismo año, quedando a cargo de la Subárea de Control el Ingeniero Paz Alanís Rodríguez, quien permanecería poco tiempo como responsable de la misma, ya que en breve dejaría la Subárea para ir a ocupar un lugar en el Área de Control Peninsular como Subjefe de Operación Sistema, y de esta manera quedaría al frente de la Subárea el Ingeniero Juan Cruz Martínez Moran, quien gestionaría la compra del predio que actualmente ocupa el edificio de la Subárea de Control.

La construcción del edificio se realizó en dos etapas. El primer nivel se edificó entre los años 1987 y 1989, quedando oficialmente terminada la obra del primer nivel el 04 de Febrero de 1989. El segundo nivel se edificó entre los años 1993 y 1994. Al momento de la firma del convenio, se encontraban como operadores los ingenieros:

- Ing. Luis Marquina Figueroa
- Ing. Cesar García Calderón
- Ing. Juan Cruz Martínez Moran

La plantilla de operadores de la Subárea de Control la llegarían a completar los ingenieros:

- Ing. Cesáreo Ibáñez
- Ing. Salvador Medina Mercado.

El resto de la platilla de la Subárea quedaría cubierta por:

- Sra. Aurea Salado Morales (Secretaria)
- Sr. Pedro Téllez Falcón (Mozo-Mensajero-Chofer)
- Sr. Carlos Carbajal Acheluz (Dibujante)

Para el año de 1980, entra en operación el primer Sistema de Control Supervisorio de lógica cableada marca Weston, con el que se tuvo control de las subestaciones Taxco e Iguala. Este sistema funcionaría hasta el año de 1984, en que fue reemplazado por otro sistema de control de la Marca Fuji.

En 1986, se habilitan las oficinas utilizadas por la supervisión de construcción en la ampliación del banco tres de la subestación “El Quemado”, para separar del tablero de la subestación al ingeniero Operador de Subárea.

El primer turno de operación realizado en las instalaciones que hoy ocupa la Subárea de Control, fue realizado el día 13 de Febrero de 1989 en el horario de 15:00 a 22:00 hrs.

También a finales de la década de los 80's, se crean las plazas de Auxiliar Técnico en Mantenimiento, Se reclasifica la plaza de Mozo-Mensajero-Chofer a Auxiliar de Servicios Chofer, se crea una segunda plaza para este perfil y se crean las Oficinas de Turnos de Operación y Programación y Equipos, quedando como Supervisor de Operación el Ing. Galdino Velázquez Gómez y como Supervisor de Programación y Equipos el Ingeniero Francisco Melchor Pérez Ponce de León. Además se crearía una plaza para un sexto Ingeniero Operador de Subárea.

En 1994 se reemplaza el sistema de Control Supervisorio Fuji por uno de la Marca VALMET, con lo cual se logró una supervisión más efectiva del Sistema bajo resguardo de la Subárea de Control. Ese mismo año se inaugura el segundo nivel del edificio.

Las necesidades de la Subárea de Control siguieron en aumento, y en el año de 2004 se integran a la platilla de trabajo dos Ingenieros en las Áreas de Análisis y Aplicaciones de Sistemas de Potencia.

En el mes de Febrero de 2009, se pone en Operación el Sistema de Control Supervisorio Marca SIEMENES (Power TG), primer sistema de

Control que incluye Aplicaciones de Potencia, entre ellas un Estimador de Estados.

La necesidad de seguir incrementando la plantilla de personal especializado, originó la reclasificación una plaza de Auxiliar de Servicios Chofer, a fin de que esta pudiera ser ocupada por otro profesionista en Sistemas Computacionales, mismo que se integró al equipo de trabajo en Agosto de 2010 al área de Programación y Equipos.

Actualmente la Subárea de Control está conformada por el Jefe de la Subárea, dos Supervisores, seis Operadores, tres Profesionistas, dos Auxiliares Especializados, un Auxiliar Administrativo y un Auxiliar de Servicios Chofer, lo que da un total de 16 trabajadores.

Jefes de la Subárea de Control

Ing. Andrés Betanzos Reyes	1973-1976
Ing. Paz Alanís Rodríguez	1976
Ing. Juan Cruz Martínez Moran	1976-1984
Ing. Humberto Abraham Pineda Figueroa	1984-2000
Ing. Emilio Luna Quiroz	2001-2005
Ing. Javier Sotelo Ochoa	2006-2011
Ing. Sergio Arturo Casillas Camacho	2012 a la fecha

1.3 Ubicación

La ZOTGM se encuentra ubicada en la calle Reyes Católicos #14, Frac. Magallanes, C.P. 39670, Acapulco, Guerrero.

En la Figura 1.2 se encuentra la ubicación satelital que proporciona por Google Maps.



Figura 1.2 Ubicación de la ZOTGM.

Obtenida de: (ZOTGM, 2016)

1.4 Misión

Asegurar dentro de un marco de competencia y actualizado tecnológicamente el servicio de energía eléctrica en condiciones de cantidad, calidad y precio, con la adecuada diversificación de fuentes de energía.

Optimizar la utilización de su infraestructura física y especial de recursos humanos.

Proporcionar una atención de excelencia a nuestros clientes.

Proteger el medio ambiente, promover el desarrollo social y respetar los valores de las poblaciones donde se ubican las obras de electrificación.

1.5 Visión

Una empresa de clase mundial que participa competitivamente en la satisfacción de la demanda de la energía eléctrica nacional e internacional, que optimiza el uso de su infraestructura física y comercial, a la vanguardia en tecnología, rentable, con imagen de excelencia, industria limpia y recursos humanos altamente calificados.

1.6 Objetivos e Indicadores

Para la ZOTGM es importante medir la calidad del servicio que están brindando y el ambiente de trabajo en el centro de trabajo, para ello se crearon objetivos e indicadores que se enlistan en la Tabla 1.1.

Tabla 1.1 Objetivos e Indicadores de la ZOTGM

Objetivos	Indicadores
1. Satisfacer los requisitos y expectativas del cliente.	<ul style="list-style-type: none"> • Satisfacción del cliente. • Compromisos de servicio. • Inconformidades por cada 1000 usuarios (improcedentes).
2. Operar sobre las bases de indicadores en materia de productividad y competitividad.	Tiempo de interrupción por usuario (sin afectaciones). Pérdidas de energía. Usuarios por trabajador de operación. Costo del kWh.
3. Promover la alta calificación y el desarrollo profesional de los trabajadores.	<ul style="list-style-type: none"> • Días anuales de capacitación. • Trabajadores evaluados en competencia laboral.
4. Garantizar la seguridad, salud y bienestar del personal.	<ul style="list-style-type: none"> • Frecuencia. • Gravedad. • Ausentismo por enfermedad general • Clima organizacional.
5. Optimizar la administración de recursos materiales y financieros.	<ul style="list-style-type: none"> • Ejecución del despacho económico. • Cobranza.
6. Implementar acciones que contribuyan al desarrollo sustentable.	<ul style="list-style-type: none"> • Cumplimiento de la legislación ambiental. • Cumplimiento de metas ambientales.
7. Mejorar continuamente la eficacia del Sistema Integral de Gestión.	<ul style="list-style-type: none"> • Eficacia del SIG.

1.7 Organigrama de la ZOTGM

En la Figura 1.3 se muestra la jerarquía administrativa de este centro de trabajo y este proyecto se desarrolla en el departamento de Programación y Equipos.

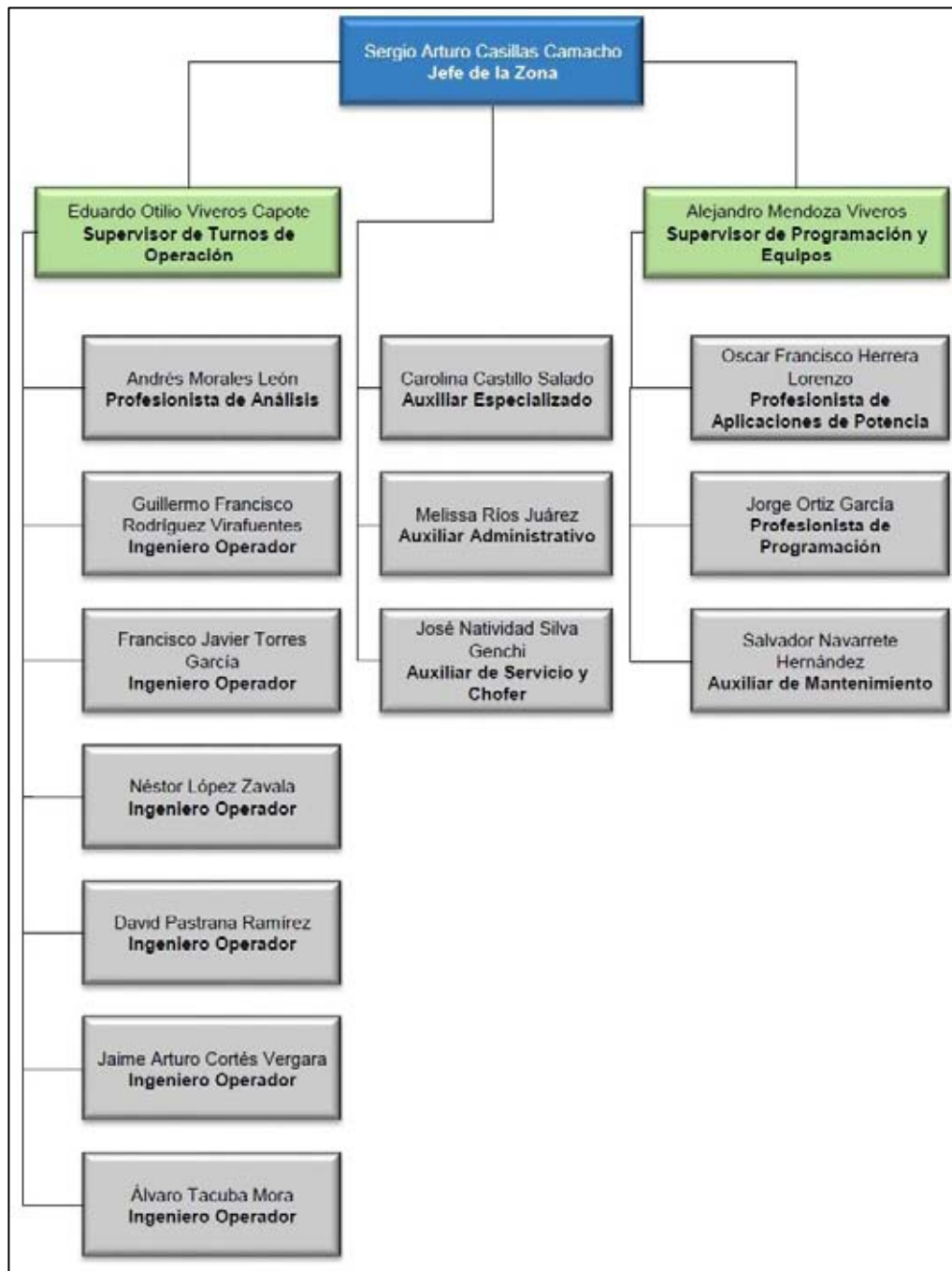


Figura 1.3 Organigrama de la ZOTGM.

Obtenida de: Elaboración propia.

1.8 Sistemas actuales

En el mercado se encuentran diferentes sistemas para la gestión de los Inventarios de una empresa, cuyas interfaces y funcionalidades de cada una cuentan con características tanto comunes como diferentes. Esta investigación se realiza con el fin de conocer los sistemas que actualmente se encuentran, realizando una crítica hacia cada uno de ellos y para finalizar, la razón por la cual no se optó por utilizar alguno de estos y necesitar un software a la medida.

Los sistemas para el control de inventarios que se encontraron en el mercado son:

- Gestionix
- Assets
- Bind erp

Gestionix

Es un sistema inteligente en línea creado en el 2012 para el control de inventarios, creado por la empresa mexicana que lleva el mismo nombre. Con este sistema se puede conocer al momento la existencia de cada producto, cual se vende mejor y cual está generando pérdida. También entrega información acerca de las sucursales, productos y almacenes controlados desde un mismo lugar, con la información siempre segura y respaldada. En la Figura 1.4 se ilustra el logotipo de esta empresa.



Figura 1.4 Logotipo de Gestionix.

Obtenida de: (Gestionix, 2016)

Entre los beneficios que otorga esta herramienta son los siguientes:

- Evitar pérdidas por robos o por un mal control de entradas y salidas.
- Los ajustes de inventario son automáticos conforme se vende y compra.
- Cargar en minutos el catálogo de productos.
- Soporte técnico las 24 horas al día.

A parte de un Sistema de control de inventarios, se agrega un punto de venta que está conectado a un lector de códigos de barra. Los puntos de venta pueden ser accedidos por cualquier persona con las credenciales correctas. También contiene un panel de administración y contabilidad, a fin de tener un mejor manejo de todos los procesos realizados por la empresa.

Como se puede notar, este sistema está enfocado hacia PYMES, en donde es necesario llevar el control de los procesos administrativos.

En la Figura 1.5 se encuentra un ejemplo de la vista de los catálogos de productos que ofrece una empresa.

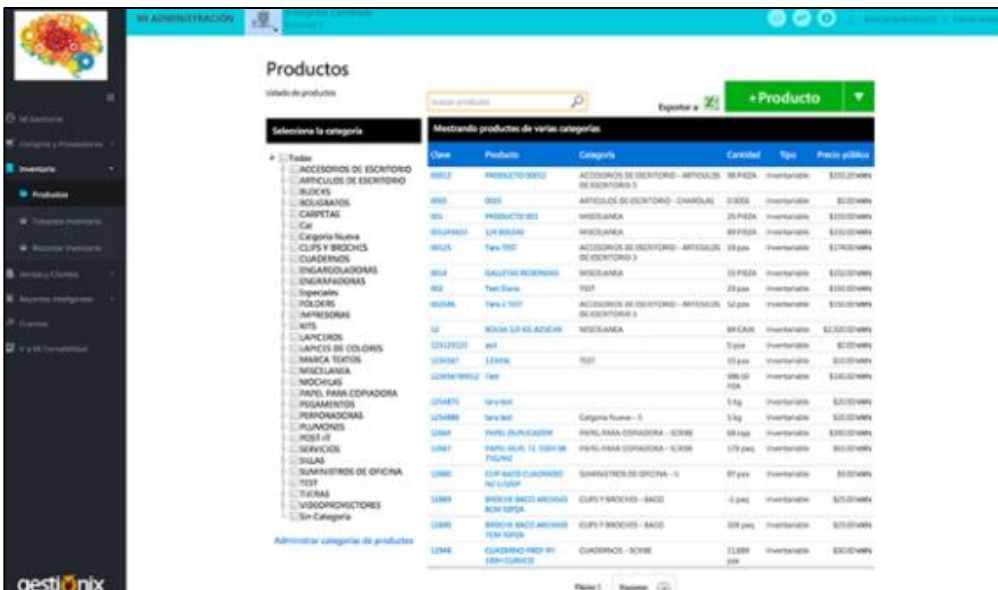


Figura 1.5 Ejemplo de vista de catálogos de Gestionix.

Obtenida de: Elaboración propia.

Assets

Assets (logo en Figura 1.6) es un software creado por la empresa Argentina llamada invgate para controlar, mantener, y administrar la infraestructura de TI. El objetivo de la aplicación es contar con información confiable, precisa, y actualizada sobre activos de TI para

reducir el TCO (Total Cost of Ownership) y maximizar el ROI (Return of Investment). Es un sistema montado completamente en Web, fácil de utilizar y de implementar. Su diseño otorga una navegación ágil, dinámica e intuitiva.



Figura 1.6 Logotipo de Assets.

Obtenida de: (Soluciones empresariales IT, 2016)

Assets es un conjunto de soluciones inteligentes para el control de infraestructuras de TI. Por mencionar alguna de estas se encuentran:

- Inventario
- Búsqueda de dispositivos
- Escritorio remoto
- Distribución de software

- Uso de software
- Ciclo de vida y finanzas
- Reportes avanzados
- Licenciamiento
- Administración de cambios.

En la Figura 1.7 se ejemplifica la ventana de esta aplicación Web. Esta cuenta con gráficas en la pantalla principal donde se visualizan las ventas actuales, cantidad de activos en inventario entre otra información relevante al usuario. Esta pantalla puede ser considerada como un dashboard o tablero.



Figura 1.7 Página principal de Assets.

Obtenida de: Elaboración propia.

Bind erp

Creado por la empresa mexicana Minnt Solutions, Bind ERP (logotipo en la Figura 1.8) es un sistema diseñado para mostrar la información más importante de inventarios y existencias cuando se necesite debido a que está implementado en la nube.

Con las herramientas disponibles se puede controlar y supervisar todos los movimientos que se realicen en los inventarios sin importar cuanto se tenga o en que parte del país estén ubicados. La interfaz intuitiva y fácil de usar permite empezar a registrar existencias, almacenes y sucursales en poco tiempo.



Figura 1.8 Logotipo de Bind erp.

Obtenida de: (itnnoacion, 2016)

Este sistema de control de inventarios en la nube ayuda a eliminar los extravíos de artículos y mantener almacenes bien organizados. Las características principales son:

- Multi Sucursal y Multi Almacén: Conectar los almacenes con otras sucursales sin necesidad de una infraestructura especializada ya que solo se necesita una computadora con internet.
- Reportes en tiempo real de existencias
- Transferencias y ajustes de producto
- Importar información en archivos de Hojas de Cálculo

En la Figura 1.9 se muestra la pantalla que incluye el formulario de registro de proveedores en la aplicación.

Figura 1.9 Ejemplo de pantalla de registro en Bind erp.

Obtenida de: Elaboración propia

Los sistemas mostrados están enfocados directamente a industrias de servicio como tiendas o restaurantes. Debido a esto,

ninguna de las 3 propuestas soluciona de manera completa la necesidad que se tiene en este centro de trabajo, por lo tanto es preciso realizar un software a la medida.

Cabe destacar que ninguno de los sistemas presentados cuenta con un módulo que permita realizar operaciones a través de un dispositivo inteligente como tabletas o celulares, ya sea mediante una aplicación móvil o diseño responsivo.

CAPITULO 2:

Caso de Estudio

CAPÍTULO II. CASO DE ESTUDIO

En este capítulo se hizo un análisis del estado actual del Sistema de Control de Inventario en la ZOTGM. Se destacaron sus especificaciones técnicas, áreas de oportunidad que existen en la herramienta y también sus puntos fuertes.

Se muestran los resultados de la entrevista realizada al supervisor del departamento de programación y equipos acerca del estado actual de la herramienta y este apartado se finaliza con una crítica acerca de la aplicación Web.

En el siguiente subtema se hablará acerca de la metodología de desarrollo a utilizar para este proyecto, mostrando los resultados de una previa investigación realizada para poder elegir la metodología, conociendo características generales de éstas.

También se mostrarán los argumentos para defender la metodología a utilizar que es la propia para realizar un sistema distribuido, así como se describen cada una de las fases que integran esta metodología que son:

- Especificaciones de alto nivel
- Estándares
- Documentación
- Herramientas de diseño

2.1 Estado Actual de la Herramienta

El Sistema de Control de Inventario se encuentra implementado en un Servidor Windows Server 2008 utilizando la herramienta para la administración de Sitios Web: Internet Information Services en su versión 7.0. Se encuentra desarrollado en la plataforma .NET versión 4.0.

El sistema manejador de Base de Datos (por sus siglas en inglés DBMS) que se utiliza para gestionar la información almacenada en este sistema es Microsoft SQL Server 2008 R2 con licencia gratuita.

Los procesos que actualmente realiza este sistema son los siguientes:

- Alta, edición, consulta y baja de Catálogos del Sistema.
- Alta, edición, consulta parametrizada y baja de Activos.
- Configuración del título en el encabezado del sistema.
- Configuración de la conexión a la Base de Datos.

Cuenta con la interfaz de los módulos Software Instalado y Pasivo, sin embargo se encuentran inoperables, ya que sólo está la plantilla de estos módulos.

En las siguientes imágenes se mostrarán la Interfaz de Usuario actual del Sistema de Control de Inventario, así como la descripción detallada de los módulos a los que hacen referencia.

En la Figura 2.1 se muestra la página principal del sistema, es decir, la primera que se muestra al abrir esta aplicación. Cabe destacar que los logos que se encuentran en el encabezado no son los oficiales, así como los títulos que hacen referencia al centro trabajo tampoco son los

nombres actuales. A pesar de contar con la opción para iniciar sesión, esta no funciona.

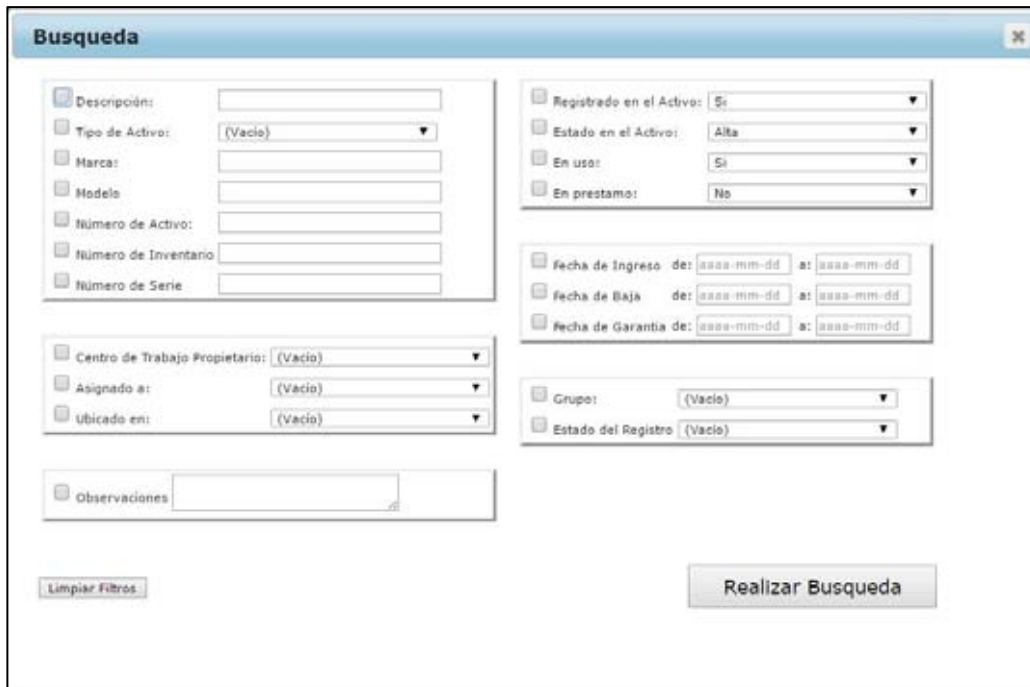


Figura 2.1 Página Principal del Sistema Actual.

Obtenido de: Elaboración propia

En la siguiente captura de pantalla (ver Figura 2.2) se encuentra el formulario para realizar la búsqueda mediante filtros dentro de los registros de la tabla Activo. Esta ventana se muestra inmediatamente al dirigirse al módulo de Activos. En esta se pueden seleccionar los datos por los cuales se va a filtrar la información; pueden elegirse diversos campos y el resultado se mostrará en la tabla que aparece en la figura 2.4 (más adelante se hará una descripción de esta). Si el usuario no desea realizar ningún filtro o simplemente cierra el dialogo, la tabla no mostrará ni un dato, es decir saldrá vacía y sólo serán visibles los encabezados de la tabla. Los botones de este formulario de búsqueda se

encuentran con tamaños desproporcionados, sin estilo, ni diseño y por lo tanto la experiencia del usuario no es agradable.



The image shows a search form titled "Busqueda" with a close button in the top right corner. The form is organized into several sections:

- Left Column:** A vertical list of search criteria, each with a checkbox and an input field:
 - Descripción: [Text input]
 - Tipo de Activo: [Dropdown menu, value: (Vacío)]
 - Marca: [Text input]
 - Modelo: [Text input]
 - Número de Activo: [Text input]
 - Número de Inventario: [Text input]
 - Número de Serie: [Text input]
- Right Column (Top):** A vertical list of filters, each with a checkbox and a dropdown menu:
 - Registrado en el Activo: [Dropdown menu, value: Si]
 - Estado en el Activo: [Dropdown menu, value: Alta]
 - En uso: [Dropdown menu, value: Si]
 - En préstamo: [Dropdown menu, value: No]
- Right Column (Middle):** Three date range filters, each with a checkbox and two date input fields (format: yyyy-mm-dd):
 - fecha de Ingreso de: [Date input] a: [Date input]
 - fecha de Baja de: [Date input] a: [Date input]
 - fecha de Garantía de: [Date input] a: [Date input]
- Right Column (Bottom):** Two more filters, each with a checkbox and a dropdown menu:
 - Grupo: [Dropdown menu, value: (Vacío)]
 - Estado del Registro: [Dropdown menu, value: (Vacío)]
- Bottom Left:** A checkbox and a text area for "Observaciones".
- Bottom Center:** A "Limpiar Filtros" button.
- Bottom Right:** A "Realizar Búsqueda" button.

Figura 2.2 Formulario de búsqueda en Activos

Obtenido de: Elaboración propia

El dialogo que contiene el formulario para dar de alta un nuevo registro dentro de la tabla Activo se encuentra en la Figura 2.3. Como se puede notar, existe similitud en el diseño del dialogo anterior para realizar búsqueda en el mismo como el orden de los componentes, esto para que el usuario se encuentre familiarizado entre los distintos formularios que existen en el sistema.

Otro punto a describir es que en la parte superior derecha se encuentra una casilla para elegir la cantidad de registros que insertarán, por

ejemplo, si darán de alta 3 discos duros con las mismas características se colocará el número en esa casilla. A pesar de poseer una buena funcionalidad, la posición dentro de la ventana no es adecuada ya que puede pasar desapercibido.

The screenshot shows a web application window titled "Activo" with a sub-header "Nuevo registro". The form is organized into two main columns. The left column contains text input fields for "Descripción:", "Marca:", "Modelo:", "Número de Activo:", "Número de Inventario:", and "Número de Serie:". Below these are dropdown menus for "Tipo Activo:", "Centro de Trabajo:", "Asignado a:", "Ubicación:", "Grupo:", and "Estado del Registro:". The right column features radio button groups for "Registrado en el activo:", "Estado en el activo:", "Equipo en uso:", and "Equipo prestado:". It also includes date input fields for "Fecha de Ingreso:", "Fecha de baja:", and "Fecha de Garantía:", and a large text area for "Observaciones:". At the bottom, there are three buttons: "Cerrar", "Limpiar Campos", and "Agregar". A checkbox labeled "Ingresar bloque de" is positioned at the top right of the form area.

Figura 2.3 Formulario registro en Activos.

Obtenido de: Elaboración propia

En la Figura 2.4 se muestra tabla que despliega la información de los registros de la Base de Datos al momento de realizar una búsqueda a través de filtros. En el encabezado principal se encuentran las opciones para editar los campos que se seleccionen de la tabla y el botón para agregar registros.

En otras opciones se desplegará un menú lateral (Figura 2.5) y al presionar el botón de Buscar se abrirá la ventana de dialogo mostrada en la Figura 2.2.

Entre otros datos que se muestran en esta página son los registros totales que se encontraron y opciones para la paginación de la tabla (total de páginas, barras de navegación y total de registros a mostrar por página).

Descripción	Tipo de Activo	Marca	Modelo	Número de Activo	Número de Inventario	Número de Serie	Fecha Ingreso	Fecha Baja	Fecha Garantía	Et A
A.C MOTOR THERMALLY PROTECTED HD 5.2VE 214	REFACCION	LG	SK49UN6084S			XWJ110137				Si
A.C MOTOR THERMALLY PROTECTED HD 5.2VE 214		LG	SK49UN6084S			TWJ220340				Si
A.C MOTOR THERMALLY PROTECTED HD 5.2VE 214	REFACCION	LG	SK49UN6084S			XWJ110106				Si
A.C MOTOR THERMALLY PROTECTED HD 5.2VE 214	REFACCION	LG	SK49UN6084S			TWJ220362				Si
ACCES POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-AP1131AG-N-K9	40030216	575622	FTX1353N00A	2010-02-15			Si
Access Point Cisco	Comunicaciones	Cisco	AIR-CAP2602I-N-K9	46027859	00039951	FTX1740J30Y	2014-04-14			Si
ACCESS POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-LAP1131AG-A-K9			FTX1353N00A			2016-09-23	Si
ACCESS POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-AP1131AG-N-K9			FTX1343N33F				Si
ACTUALIZACIÓN DE AUTOCAD 2009(A.AUTOCAD)	Software			43016727	351858	342-43122618				Si
AIRE ACONDICIONADO	Aire Acondicionado			33026375	291128				1988-11-01	Si
Aire Acondicionado (38XCA36228-S-I)	Aire Acondicionado	CARRIER	FKGC183C O	33026373	00356682	2604N75164			2004-12-17	Si
Aire Acondicionado (38XCA362258-S-I)	Aire Acondicionado	CARRIER	FKGC183C O	33026372	00356681	2604N75167	2004-12-17			Si

Figura 2.4 Tabla de resultados de la búsqueda en Activos.

Obtenido de: Elaboración propia

En la Figura 2.5 se presenta un menú lateral que cuenta con dos pestañas: La primera contiene las opciones para que el usuario pueda decidir que columnas de la tabla desea visualizar y cuales no; en la

segunda pestaña están las opciones para exportar los datos mostrados o elegir qué información desea guardar en un archivo, en formato PDF pero estas opciones aún no están habilitadas.

El desplegar una barra de opciones de manera lateral genera un espacio estrecho entre cada uno de los elementos haciendo la interfaz visualmente cargada y genera una perspectiva de desorden dentro de ésta.

Activo Editar Seleccionados (Por Bloque) Agregar Registro

Resultados de la Búsqueda (661 Registros) Otras Opciones

del 1 al 20 Página: 1 Registros por página: 20 Buscar

Descripción	Tipo de Activo	Marca	Modelo	Número de Activo	Número de Inventario	Número de Serie
A-C MOTOR THERMALLY PROTECTED HD 52VE 214	REFACCION	LG	5K49UN6084S			XWJ119
A-C MOTOR THERMALLY PROTECTED HD 52VE 214		LG	5K49UN6084S			TWJ220
A-C MOTOR THERMALLY PROTECTED HD 52VE 214	REFACCION	LG	5K49UN6084S			XWJ119
A-C MOTOR THERMALLY PROTECTED HD 52VE 214	REFACCION	LG	5K49UN6084S			TWJ220
ACCES POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-AP1131AG-N-K9	40030216	575622	FTX113
Access Point Cisco	Comunicaciones	Cisco	AIR-CAP2002I-N-K9	45627859	00939951	FTX117
ACCES POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-LAP1131AG-A-K9			FTX113
ACCES POINT PARA INTERIORES	UNIFICA(Access Point)	CISCO	AIR-AP1131AG-N-K9			FTX114
ACTUALIZACIÓN DE AUTOCAD 2009.A.AUTOCAD	Software			43016727	351858	342-431
AIRE ACONDICIONADO	Aire Acondicionado			33626375	291128	
Aire Acondicionado (8XCAS226-C-1)	Aire Acondicionado	CARRIER	FKGC183C O	33626373	00356682	2604N7
Aire Acondicionado (8XCAS226-C-1)	Aire Acondicionado	CARRIER	FKGC183C O	33626372	00356681	2604N7

Figura 2.5 Opciones de personalización de la Tabla de Activos.

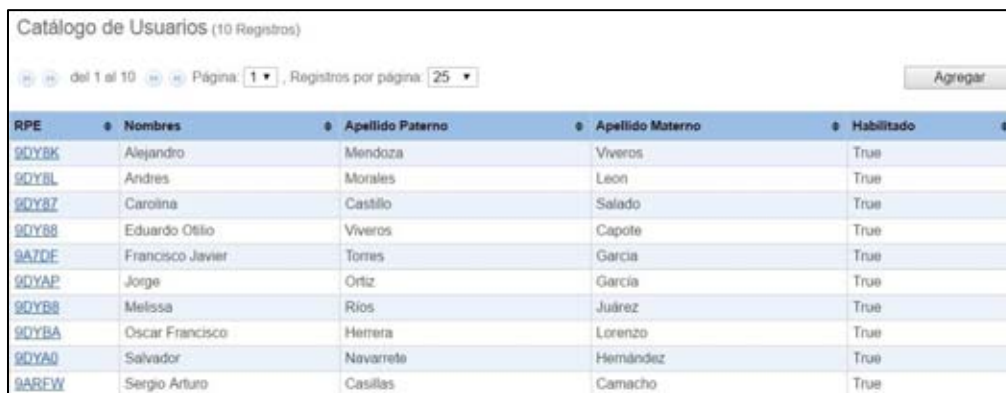
Obtenido de: Elaboración propia

En la Figura 2.6 se encuentra el diseño de las páginas para los catálogos, se utilizó como ejemplo el de los usuarios de este Centro de Trabajo. La tabla utiliza el mismo diseño que el contenido anterior, al igual

que se respetan las opciones de paginado. Se encuentra el botón para agregar un registro al catálogo pero está inhabilitado.

Para realizar la modificación de algún registro se debe de dar clic sobre el RPE (que es el número de empleado en la empresa) y este realiza una carga dirigiéndose a otra página para realizar el cambio pertinente, haciendo laborioso este procedimiento y poco intuitivo.

En el encabezado de la tabla se encuentra el total de datos a mostrar y la paginación. Usar paginación en la tabla no es funcional debido a que la cantidad de registros mostrados no es muy grande.



Catálogo de Usuarios (10 Registros)

del 1 al 10 | Página: 1 | Registros por página: 25 | Agregar

RPE	Nombres	Apellido Paterno	Apellido Materno	Habilitado
90YBK	Alejandro	Mendoza	Viveros	True
90YBL	Andres	Morales	Leon	True
90YBT	Carolina	Castillo	Salado	True
90YBB	Eduardo Osilio	Viveros	Capote	True
90YDE	Francisco Javier	Torres	Garcia	True
90YAP	Jorge	Ortiz	Garcia	True
90YBB	Melissa	Rios	Juárez	True
90YBA	Oscar Francisco	Herrera	Lorenzo	True
90YAD	Salvador	Navarrete	Hernández	True
90YEW	Sergio Arturo	Casillas	Camacho	True

Figura 2.6 Interfaz gráfica del Catálogo de Usuarios.

Obtenido de: Elaboración propia

En la Figura 2.7 se encuentra el módulo de configuración de la interfaz de la aplicación. El objetivo de este módulo es que el administrador del sistema pueda realizar cambios al texto, los colores y tamaños de letra que están ubicados en los encabezados de la aplicación, sin necesidad de que se encuentre un especialista en el área

de desarrollo de sistemas o que se tenga que recurrir al código fuente para modificarlo y tener que volver a publicar la solución.

En la Figura 2.8 se muestra la configuración para colocar la dirección de la Base de Datos a la que se conectará el sistema; esto se realizó por si llegara a cambiar la dirección del servidor de base de datos en un futuro.

La Figura 2.7 y 2.8 se encuentran en un panel de administración que resulta muy útil, ya que garantiza que la aplicación puede ser portable y administrable por los mismos usuarios aunque no tengan nociones de programación.

Interfaz Gráfica		
Título 1 - Texto:	Área de Control Oriental	
Título 1 - Color (#8Hex):	#00A969	(Ej. #00A969)
Título 2 - Texto:	Subárea de Control Guerrero Morelos	
Título 2 - Color (#8Hex):	#8DC63F	(Ej. #8DC63F)
Título 3 - Texto:	Control de Inventario	
Título 3 - Color (#8Hex):	#000000	(Ej. #000000)

Figura 2.7 Configuración de la Interfaz Gráfica.

Obtenido de: Elaboración propia

Base de Datos		
Dirección IP del Servidor:	127.0.0.1	(Ej: 127.0.0.1)
Puerto del Servidor:	1433	(Ej: 1433)
Nombre Base de Datos:	CFE_SGM_Inventario	
Usuario de Acceso:		
Password de Acceso:		

Figura 2.8 Configuración de la conexión a la Base de Datos.

Obtenido de: Elaboración propia

A continuación, se realizó un listado general de las problemáticas encontradas en este sistema, las cuales a través del mismo se desean resolver.

En primer lugar, el tiempo que el usuario debe esperar para que se visualice el contenido de la página es tardado. Se necesitan optimizar los tiempos de carga.

En el diseño de la página no se aprovecha todo el ancho de la pantalla, lo cual hace que los tamaños de los elementos sean pequeños imposibilitando al usuario la visualización de los mismos. De igual manera se detectó que la combinación de colores afecta la visión puesto que para algunos elementos los colores son muy brillantes.

Siguiendo con los problemas en el diseño, la posición de los elementos no es adecuada ya que, aparte de que no se utiliza todo el ancho de la página, los botones y herramientas se sobreponen unos con otros en la interfaz.

Se observa que la base de datos necesita ser reestructurada, ya que el problema de latencia recae en la deficiencia para realizar consultas, entre otras operaciones básicas.

El sistema no es capaz de generar reportes de la información que tiene almacenada en la base de datos.

Existen módulos en el menú principal, como: "Software Instalado" y "Pasivo", que no están terminados.

No se contempló restricción alguna en la seguridad en la aplicación para la manipulación en los usuarios.

El sistema realiza de forma cíclica peticiones que gastan recursos del servidor.

En la página de inicio no existe información acerca de la aplicación o un manual en donde se le explique al usuario como debe utilizarla correctamente.

Debido a que el tamaño físico del tipo de insumos puede llegar a ser considerable y esto complicar el registro desde una computadora de oficina, se detectó la posibilidad de crear una aplicación móvil en la cual se puedan realizar las operaciones de altas, modificaciones y bajas.

2.2 Metodologías de Desarrollo

La necesidad de utilizar una metodología de software recae en los múltiples requerimientos que realizaba quien necesitaba un sistema de software. Según Dante Cantone, en su *libro La Biblia del Programador (2008)* una metodología de desarrollo es:

“Modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Esta sistematización nos indica como dividiremos un gran proyecto en módulos más pequeños llamadas etapas, y las acciones que corresponden en cada una de ellas, nos ayuda a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo en que administraremos el proyecto”

Para la realización de este proyecto inicialmente se optó por elegir una metodología de desarrollo para cada una de las aplicaciones que componen este sistema.

Después de analizar los requerimientos del proyecto y haber realizado una investigación de las metodologías existentes para desarrollo Web (RUP, XP, SCRUM, Prototipos) y Móvil (Cascada, TDD, Mobile – D), se decidió por elegir las siguientes:

- Para la aplicación Web: Desarrollo por prototipos,
- Para la aplicación móvil: Mobile – D.

En la Tabla 2.1 se muestran las etapas de desarrollo correspondientes a las metodologías que se eligieron, esto con el fin de comparar el proceso que se debe realizar en cada una de ellas.

Tabla 2.1 Comparación de las etapas de desarrollo entre las metodologías seleccionadas.

PROTOTIPOS	MOBILE – D
<p>1. Recolección y refinamiento de requisitos</p> <p>(Las siguientes etapas pueden ser repetitivas)</p> <p>a. Diseño rápido</p> <p>b. Construcción del prototipo</p> <p>c. Evaluación del prototipo por el cliente</p> <p>d. Refinamiento del prototipo</p> <p>2. Producto de Ingeniería</p>	<p>1. Exploración</p> <p>2. Iteración</p> <p>3. Producción</p> <p>4. Estabilizar</p> <p>5. Pruebas y reparación en el sistema</p>

Ambas se adaptaban a la forma en la que se desarrollará el proyecto; sin embargo, al manejar diferentes tipos de metodologías se rompe con la definición de integración de aplicaciones, propia del tipo de sistema.

Así que, realizando una investigación acerca de la manera en que se desarrolla un sistema distribuido se encontró que existe una propia metodología, basada en el modelo de 3 capas para el desarrollo de software.

Finalmente, se decide por utilizar esta ya que no interfiere con la definición principal del funcionamiento de este sistema y cumple con los requisitos que el sistema.

2.3 Metodología de los Sistemas Distribuidos

Los Sistemas Distribuidos es una colección de equipos independientes que aparece a sus usuarios como un único sistema coherente (Tanenbaum & Van Steen, 2007). Esta definición tiene varios aspectos importantes. La primera de ellas es que un sistema distribuido consta de componentes (es decir, ordenadores) que son autónomas. Un segundo aspecto es que los usuarios (ya sean personas o programas) piensan que se trata de un único sistema. Esto significa que de un modo u otro los componentes autónomos necesitan colaborar. ¿Cómo establecer esta colaboración? Esta es la parte central del desarrollo de sistemas distribuidos.

Debido a esta definición, se precisó en ocupar una metodología que logre incorporar esta definición en sus fases y la mejor decisión fue ocupar la propia del sistema.

Las fases fundamentales para el desarrollo de un sistema distribuido son:

1. Especificación del Alto Nivel,
2. Estándares,
3. Herramientas de Diseño,
4. Documentación.

Esta metodología ocupa un caso particular de la Arquitectura Cliente – Servidor, que es una tendencia que actualmente está surgiendo en los Sistemas Web. En la Figura 2.9 se muestra el diagrama de los elementos que componen esta arquitectura.

Sus componentes son los usuarios (que puede ser un visualizador de páginas Web o un dispositivo móvil), el gestor de comunicaciones que son los protocolos de transferencia de datos a utilizar; el tercer elemento son la interfaz de usuario (ya sea HTML para páginas Web o el lenguaje XAML para los layouts de la aplicación móvil), la lógica de negocio está representada por un servidor de aplicaciones (por ejemplo IIS, Apache o Tomcat) y por último la Base de Datos que será cualquier gestor de Base de Datos de SQL (MySQL, SQLServer, MariaDB). Todos estos componentes deberán estar presentes dentro de las aplicaciones con las que contará un sistema distribuido y de esta manera poder validar correctamente su funcionamiento (Baudes Rubio, 2002).

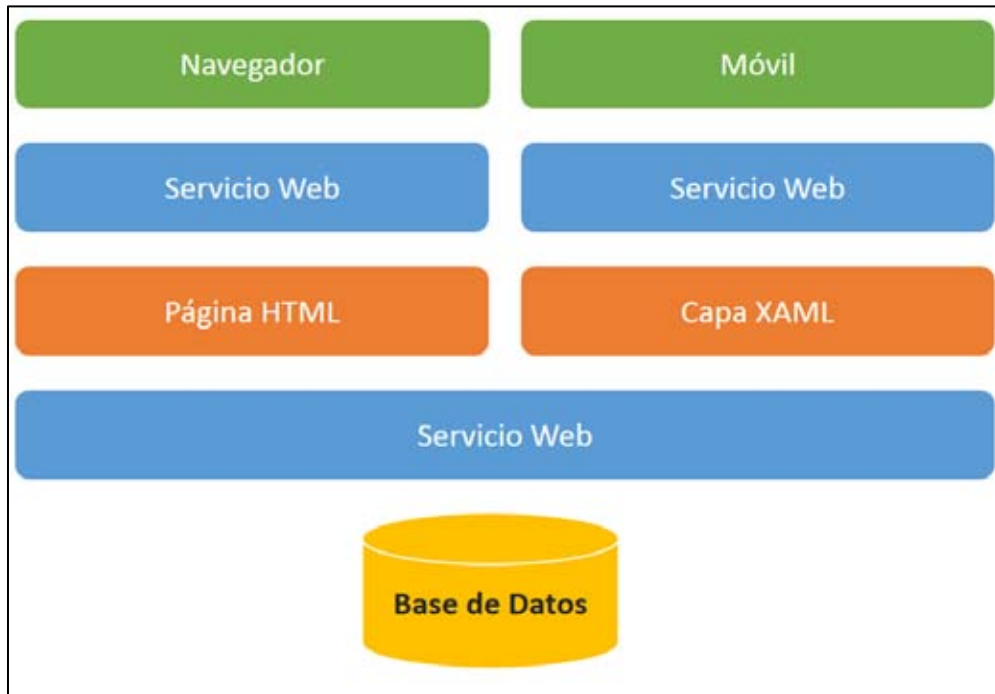


Figura 2.9 Estructura de caso particular de la Arquitectura Cliente – Servidor.

Obtenido de: Elaboración propia

2.3.1 Especificaciones de alto nivel.

Esta parte trata lo relacionado a las especificaciones de aplicaciones distribuidas, que comúnmente tienen un gran número de requerimientos de desempeño que los hace complejo de explicar.

Resulta relativamente fácil describir un sistema distribuido dando una explicación detallada de su implementación, por ejemplo; dónde se localiza la información, cuántas réplicas de la información existen, cómo se procesan las peticiones y cómo se comunican todas las piezas del sistema. Así como las especificaciones del usuario para programas

secuenciales, las especificaciones de sistemas distribuidos deberían expresarse en términos orientados al usuario y deberían ser libres de detalles de implementación.

Las especificaciones deben describir todas las constantes relevantes en su comportamiento observable; incluyendo el comportamiento de las operaciones invocadas por los usuarios y si el sistema está activo, las operaciones que el sistema realizará internamente.

El usuario de un sistema podría ser un programa o proceso, o si el sistema es interactivo, una persona; puede tener usuarios que trabajen concurrentemente y todos estos usuarios pueden invocar sus operaciones en paralelo.

En las especificaciones, cada operación es vista como una acción atómica. Estas operaciones atómicas poseen dos propiedades importantes: serializabilidad y totalidad. La serializabilidad se refiere a que la ejecución concurrente de un grupo de acciones es equivalente a la ejecución secuencial de las mismas operaciones. La totalidad quiere decir que cada operación, o se ejecuta total y exitosamente, o falla y no tiene efecto sobre el estado del sistema.

La técnica básica que es usada en la escritura de especificaciones es introducir especificaciones no deterministas en el modelo que describa la propagación de la información sobre todas las réplicas y los efectos de no serializabilidad de las operaciones concurrentes. También es de ayuda estructurar las especificaciones, de manera que se puedan distinguir los efectos esperados y no deseados, de los inesperados. Esta distinción es importante tanto para usuarios como para los

implementadores del sistema, y hace que las especificaciones sean fáciles de entender.

La implementación funciona procesando cada operación en un sólo nodo, cada uno de los cuales cuenta con una copia del diccionario. Si se realiza una operación de inserción o eliminación en un nodo, sólo se actualiza la copia de éste nodo, y posteriormente, por medio del envío de mensajes, se propagan las modificaciones a los demás nodos del sistema. Debido a que un nodo puede no saber de las operaciones ejecutadas en otros nodos, se requiere que los usuarios se aseguren que los elementos dados sean insertados al menos una vez, y que un elemento sea borrado sólo si este ha sido insertado anteriormente, y el nodo que realizará la eliminación sepa de la inserción (Anónimo, Metodología de desarrollo, 2009; Anónimo, Metodología de desarrollo, 2009) (Perez Santillan, Nieto Sanchez, Flores Iñiguez, & Ortiz Gómez, 2009).

2.3.2 Estándares

Los estándares surgieron primero como un campo de la ingeniería y se formaron grupos para concordar en las unidades de medición y en prácticas de la ingeniería. La Interconexión de Sistemas Abiertos (OSI, por sus siglas en inglés) surgió en forma similar, y esta arquitectura ha ganado importancia al ofrecer una plataforma alternativa en comunicaciones para las arquitecturas dominantes.

Para los sistemas distribuidos son necesarios los estándares por dos razones:

1. Diversidad. Por naturaleza, los seres humanos cotidianamente son 'distribuidos': trabajan en diferentes lugares y la recolección y almacenamiento de la información se realiza en diferentes locaciones. Para lograr la mejor interacción hombre – computadora, respuestas rápidas e independencia, es necesario colocar a las computadoras cerca de las personas que las usan, lo que lleva al concepto de computación personal. Sin embargo, las personas requieren de compartir información y procesos. Los sistemas distribuidos se tienen que adaptar a diferentes requerimientos operacionales. Por ejemplo, la automatización de una fábrica requiere respuestas en tiempo real y un alto grado de confiabilidad; la aplicaciones en una oficina requieren alta seguridad en lugar de respuestas en tiempo real.

2. Fragmentación. Hasta ahora, no existe una arquitectura abierta que facilite la construcción de sistemas distribuidos en una base de diversos vendedores, la cual pueda expandirse a diversos dominios de aplicación (Perez Santillan, Nieto Sanchez, Flores Iñiguez, & Ortiz Gómez, 2009).

Estos aspectos se concretizan en dos tipos de objetivos del proyecto (Anónimo, Metodología de desarrollo, 2009):

Objetivos de coordinación.

- Estándares. Construir y mantener en los proyectos una conciencia de contribuciones técnicas dentro del proceso de estándares internacionales.
- Transferencia de tecnología. Proveer conocimiento técnico para asistir en la formulación de una explotación industrial de procesamiento abierto distribuido.

- **Diseminación.** Publicar anualmente los resultados del proyecto, en forma de actualizaciones y versiones extendidas del Manual de Referencia ANSA y liberaciones posteriores del ANSA Testbench.

Objetivos técnicos.

- **Fundaciones.** Desarrollar conceptos de arquitectura, técnicas de modelado y taxonomías por las cuales se pueda describir consistentemente a los sistemas de procesamiento distribuido.
- **Ingeniería.** Desarrollar y especificar métodos de ingeniería que serán requeridas por diseñadores de aplicaciones distribuidas.
- **Productividad.** Mejorar la productividad en la construcción e interconexión de aplicaciones integradas de procesamiento por medio de la provisión de un conjunto de bloques de construcción.

2.3.3 Herramientas de Diseño

Los principios comunes del diseño, proveen un modelo consistente de procesamiento de información a través de un sistema, el cual facilita la tarea de integrar diversos paquetes de aplicaciones en un sistema coherente.

Para los operadores de sistemas, el diseño marca un acercamiento a la administración diaria de los sistemas, además del crecimiento integral del sistema para ajustar los nuevos requerimientos y los cambios en la industria de la tecnología de información.

Para los desarrolladores de sistemas, los principios de diseño deben aumentar la productividad, mejorar la reusabilidad del software y facilitar la generación automática de software, a partir de sentencias declarativas de requerimientos. El uso de los principios de diseño reduce también, el tramo que separa la interconexión de sistemas separados, simplificando la complejidad de interconexión y extendiendo el rango de componentes que pueden ser interconectados (Perez Santillan, Nieto Sanchez, Flores Iñiguez, & Ortiz Gómez, 2009).

Los principios de diseño deben aumentar la productividad, mejorar la reusabilidad del software y facilitar la generación automática de software, a partir de sentencias declarativas de requerimientos. El uso de los principios de diseño reduce también, el tramo que separa la interconexión de sistemas separados (Anónimo, Metodología de desarrollo, 2009).

Los diagramas que se proponen para representar los diseños del sistema distribuido son los siguientes:

- Modelo de Programación
- Modelo de datos
- Modelo de procesos
- Modelo de comunicación
- Modelo de sincronización
- Modelo de E/S

2.3.4 Documentación

El ciclo de vida del desarrollo del software contempla a la documentación como una etapa de gran importancia para el producto final. Resulta crucial guardar las especificaciones establecidas que fundamentan el funcionamiento del software, así como de los componentes a partir de los que se forma la aplicación final. Una metodología tiene la finalidad de dirigir al desarrollador a una ruta racional, apuntando a las necesidades a satisfacer, en qué orden y cuánto tiempo debe tomar cada actividad.

Durante el análisis de requerimientos se debe desarrollar una especificación de tales requerimientos, plasmada en un documento que sirve para especificar los requerimientos de los usuarios y es además, un punto de partida para el diseño. Distingue siete roles de usuarios para la documentación del diseño:

1. El administrador del proyecto requiere información para planear, controlar y administrar el proyecto. Debe estar en posibilidades de identificar cada componente del sistema y entender su propósito y funcionamiento.
2. El administrador de configuración necesita información para poder ensamblar varios componentes en un solo sistema y poder controlar los cambios.
3. El diseñador requiere de información acerca del uso y funcionamiento de cada componente, y su interfaz con otros componentes.

4. El programador debe conocer los algoritmos que se utilizarán, las estructuras de datos y la comunicación entre componentes.
5. Se requiere que el probador de unidades conozca información detallada de los componentes, como algoritmos y datos requeridos.
6. Al probador de integración le corresponde conocer las relaciones entre componentes y la función y uso de los componentes envueltos.
7. El programador de mantenimiento debe tener una visión de cómo se satisfacen los requerimientos usando todos los componentes.

Cada módulo (entidades o personas enroladas en el desarrollo) cuenta con una serie de atributos, entre los que se mencionan los siguientes:

- Identificación. Un nombre para hacer referencia a un componente. Debe ser único.
- Tipo. El tipo del componente, puede ser un procedimiento, un archivo.
- Función. Para lo que el componente fue diseñado.
- Subordinados. De cuales entidades se compone un módulo.
- Dependencias. Es una descripción de las relaciones con otros componentes.
- Interfaz. Es una representación de la interacción entre los componentes.
- Recursos. Son entidades externas al diseño, tales como memoria, impresoras, procesadores.

- Procesamiento. Se refiere a los algoritmos utilizados y el tratamiento a las excepciones.
- Datos. Una descripción de la representación, uso, formato y réplicas de los datos.

Como una buena práctica para un sistema distribuido, se recomienda establecer desde los documentos de especificación, todo lo referente al ambiente técnico en el que se planea funcione el sistema. Se deben establecer al menos los siguientes aspectos técnicos:

- Las plataformas de hardware a utilizarse.
- El o los sistemas operativos utilizados en cada nodo.
- El administrador de base de datos a usar.
- El software de red elegido.
- Los lenguajes seleccionados para el desarrollo.
- El medio o tecnología de red para conectar a los nodos.

En la Figura 2.10 se encuentra un diagrama para ilustrar cada una de las etapas de esta metodología.



Figura 2.10 Fases de la Metodología para Sistemas Distribuidos.

Obtenido de: Elaboración propia

CAPITULO 3:

Conceptos y definiciones básicas

CAPÍTULO III. CONCEPTOS Y DEFINICIONES BÁSICAS

Dentro de este capítulo se explicará toda la información referente a los conceptos necesarios para entender el desarrollo de este sistema distribuido.

Como este proyecto consta de una aplicación Web y una aplicación Móvil, se dividirá el contenido de este capítulo como:

- **Herramientas para la Programación Web:** El entorno de desarrollo a utilizar, programación del lado del cliente (Front – End) y programación del lado del servidor (Back – End).
- **Herramientas para la Programación Móvil:** La definición de aplicación móvil, el entorno de desarrollo a utilizar y el lenguaje a utilizar.
- **Control de Versiones:** Para la realización de este proyecto se implementó un programa para controlar el código de proyecto y poder trabajar paralelamente ante el mismo proyecto. Debido a esto, en esta sección se definirá este concepto y el servidor de control de versiones a utilizar.
- **Sistemas Distribuidos:** Es necesario conocer la definición y un panorama general de este tipo de sistemas ya que será el producto final de este trabajo. En cada una de las definiciones se especificará la manera en la que se aplicará a este proyecto.

3.1 Herramientas para Programación Web

Para el desarrollo de un Sistema Web existen diferentes herramientas que se pueden utilizar según el lenguaje de programación seleccionado. Cabe aclarar que cuando se habla de herramientas se está haciendo referencia a los entornos de desarrollo, editores de texto, servidores Web, sistemas operativos para servidores, manejadores de bases de datos, entre otros. A continuación se explicarán estos conceptos aplicado a lo que se utilizará para la creación de este proyecto.

Visual Studio

Visual Studio es un entorno de desarrollo creado por Microsoft. Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles.

Los lenguajes que soporta son Visual Basic, Visual C# y Visual C++. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML (Microsoft, 2012). En la Figura 3.1 se encuentra el logotipo de Visual Studio en su versión 2015.



Figura 3.1 Logotipo de Visual Studio.

Obtenido de: (Tower, 2015).

Existen diferentes versiones de este entorno. La versión actual es la 2015, que será la elegida para trabajar. En esta versión existen diferentes ofertas del producto:

- Community: Versión gratuita de este IDE, pensada en desarrolladores emprendedores y pequeñas empresas. Esta es la que se utiliza para el desarrollo del proyecto.
- Professional: Versión de paga, pensada para las grandes empresas.
- Enterprise: Esta versión contiene mayores herramientas para el análisis de los procesos, consumo de recursos y documentación instantánea, otorgando distintos diagramas UML bajo los estándares de Microsoft.

En la Figura 3.2 se muestra la tabla proporcionada por Microsoft comparando las características de cada una de las versiones de este entorno.

	Visual Studio Community	Visual Studio Professional	Visual Studio Enterprise	Visual Studio Test Professional	MSDN Platforms
+ Supported Usage Scenarios	●●●○	●●●●	●●●●	●●●●	●●●●
+ Debugging and Diagnostics	●●●○	●●●○	●●●●	○○○○	○○○○
+ Testing Tools	●○○○	●○○○	●●●●	●●○○	●●○○
+ Integrated Development Environment	●●●○	●●●○	●●●●	○○○○	○○○○
+ Development Platform Support	●●●●	●●●●	●●●●	○○○○	○○○○
+ Xamarin Mobile Development	●●○○	●●○○	●●●●	○○○○	○○○○
+ Architecture and Modeling	●○○○	●○○○	●●●●	○○○○	○○○○
+ Lab Management	○○○○	○○○○	●●●●	●●●●	●●●●
+ Team Foundation Server features	○○○○	●●●○	●●●●	●●●●	●●●●
+ Collaboration Tools	●●●●	●●●●	●●●●	●●●○	●●●○
+ Team Collaboration Benefits	Included with all subscriptions				
+ Subscriber Benefits	Included with annual cloud subscriptions and standard subscriptions				
+ Visual Studio Dev Essentials Benefits	Free for all developers ¹⁹				

Figura 3.2 Tabla comparativa de productos de Visual Studio.

Obtenido de: (Microsoft, 2012).

Visual Studio Community

Es un completo IDE extensible y gratuito para crear aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube (Microsoft, 2016). Ofrece todas las funciones de la versión Professional sólo que completamente gratis.

Esta versión será la que se utilizará para desarrollar el sistema de control de inventario.

Las características de esta versión de Visual Studio son:

- Flexibilidad: Crear aplicaciones para cualquier plataforma.
- Productividad: Diseñadores, editores, depuradores y generadores de perfiles en una sola herramienta.
- Ecosistema: Acceso a miles de extensiones.
- Lenguajes: Programar en C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python, etc.

A parte de las herramientas para generar aplicaciones de escritorio y móviles, se integran características de Xamarin que completan la experiencia de desarrollo de aplicaciones móviles listas para usar para Android, iOS y Windows. En la Figura 3.3 se muestra el logo de esta versión de Visual Studio.



Figura 3.3 Logotipo de Visual Studio Community 2015.

Obtenido de: (Microsoft, 2016).

Plantillas de Visual Studio

Visual Studio facilita plantillas que proporciona una estructura para los diferentes tipos de aplicaciones. Esta estructura incluye:

- Código inicial para construir y crear rápidamente aplicaciones funcionales.
- Soporte a componentes y controles dependiendo del tipo de proyecto seleccionado.
- Configurar el IDE dependiendo del tipo de aplicación que sea.
- Agregar referencias a los ensamblados que se requieran.

Internet Information Services (IIS)

Es un potente servidor Web proporcionado por Windows que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web (Microsoft, 2016). La versión más actual es la 10.0 pero para este proyecto se ocupará IIS 8.0 y su logo se encuentra ilustrado en la Figura 3.4.



Figura 3.4 Logotipo de IIS.

Obtenido de: (Microsoft, 2016).

Windows Server

Es un sistema operativo diseñado para servidores de Microsoft. Ofrece más control sobre la infraestructura de servidores y red, mejor hosting, protección del sistema operativo y el entorno de red, herramientas administrativas intuitivas, facilidad de consolidación, virtualización de servidores y aplicaciones (AEvitas, 2013). La última versión es la 2016, para este proyecto se ocupará la 2012 R2 (Logo en la Figura 3.5).



Figura 3.5 Logotipo de Windows Server.

Obtenido de: (AEvitas, 2013).

Concluyendo, las herramientas presentadas para la programación Web han sido elegidas según el lenguaje de programación a utilizar. Existen diferentes herramientas para un mismo lenguaje, según las necesidades del desarrollador es la que se elige. La programación Web se divide en dos tecnologías: Front – End y Back – End.

3.1.1 Front – End

Trabaja del lado Cliente, en el navegador, en el lado de lo que se ve. Principalmente se ocupa de los componentes externos del sitio Web o de la aplicación Web (campusMVP, 2015).

CSS

Cascading Style Sheets (CSS) u Hojas de Estilo en Cascada (Logo en la Figura 3.6) es un lenguaje que describe el estilo y como se deben mostrar los elementos de un documento HTML. Puede controlar el diseño de varias páginas web en un solo archivo e incluso las vistas en variaciones en la pantalla para diferentes dispositivos y tamaños de pantalla. Realmente sin este componente, lo único que sería visible en una página en Internet sería el esqueleto, sin ningún color o formato, sería solamente texto plano. La última versión es CSS3 (w3Schools, 2016).



Figura 3.6 Logotipo de CSS3.

Obtenido de: (w3Schools, 2016).

HTML5

Hyper Text Markup Language (HTML) o Lenguaje de Marcas de Hipertexto, es un lenguaje que describe la estructura de las páginas Web utilizando etiquetas, las cuales los navegadores Web no muestra (w3schools, HTML Introduction, 2016). La última versión de HTML es HTML5 (Logo en la Figura 3.7).



Figura 3.7 Logotipo de HTML5.

Obtenido de: (w3schools, HTML Introduction, 2016).

Bootstrap

Bootstrap (Logo en la Figura 3.8) es el Framework más conocido, creado y liberado por Twitter que utiliza HTML, CSS y JavaScript para el desarrollo de aplicaciones Web Responsivas, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla manteniendo un diseño agradable (getbootstrap, 2016).



Figura 3.8 Logotipo de Bootstrap.

Obtenido de: (World Vector Logo, 2016).

Diseño Web Responsivo

El diseño Web Responsivo hace que las páginas Web se ajusten a todos los dispositivos (equipos de escritorio, dispositivos móviles y tabletas), es sobre todo el uso de CSS y HTML para ajustar, mostrar, esconder o mover el contenido para hacer que este luzca presentable en cualquier pantalla (w3schools, HTML Responsive Web Design, 2016).

Dentro de los patrones de diseño responsivo se encuentra SOLID, acrónimo inventado por Robert C. Martin para establecer los cinco principios básicos de la programación orientada a objetos y diseño.

S - Responsabilidad simple (Single responsibility): Este principio trata de destinar cada clase a una finalidad sencilla y concreta. En muchas ocasiones estamos tentados a poner un método reutilizable que no tienen nada que ver con la clase simplemente porque lo utiliza y nos pilla más a mano. En ese momento pensamos "Ya que estamos aquí, para que voy a crear una clase para realizar esto. Directamente lo pongo aquí".

O - Abierto/Cerrado (Open/Closed): Principio atribuido a Bertrand Meyer que habla de crear clases extensibles sin necesidad de entrar al código fuente a modificarlo. Es decir, el diseño debe ser abierto para poderse extender pero cerrado para poderse modificar.

El uso más común de extensión es mediante la herencia y la reimplementación de métodos.

L - Sustitucion Liskov (Liskov substitution): Este principio fue creado por Barbara Liskov y habla de la importancia de crear todas las clases derivadas para que también puedan ser tratadas como la propia clase base.

I - Segregacion del interface (Interface segregation): Este principio fue formulado por Robert C. Martin y trata de algo parecido al primer principio. Cuando se definen interfaces estos deben ser específicos a una finalidad concreta.

D - Inversión de dependencias (Dependency inversion): También fue definido por Robert C. Martin. El objetivo de este principio conseguir desacoplar las clases. En todo diseño siempre debe existir un acoplamiento pero hay que evitarlo en la medida de lo posible. Un sistema no acoplado no hace nada pero un sistema altamente acoplado es muy difícil de mantener (Rubira, 2011).

Existen otros 5 los patrones existentes ya definidos (Azaustre, 2015): Tiny Tweaks, Mostly Fluid, Column Drop, Layout Shifter y Off Canvas.

Se resaltaré el uso del patrón Mostly Fluid. Cuando se visualiza una aplicación desde un dispositivo móvil, todo forma una única columna, y en varias filas quedan colocados los distintos bloques.

Según vaya creciendo la pantalla, los distintos bloques se agrupan ocupando toda la pantalla disponible.

En la Figura 3.9 se muestra el esquema de funcionamiento de este patrón. Se puede notar como es que los elementos van anidándose hacia abajo para poder adaptarse según la resolución de la pantalla: Smartphone (a), Tablet (b), Monitor CRT (c) y Pantalla LCD (d).

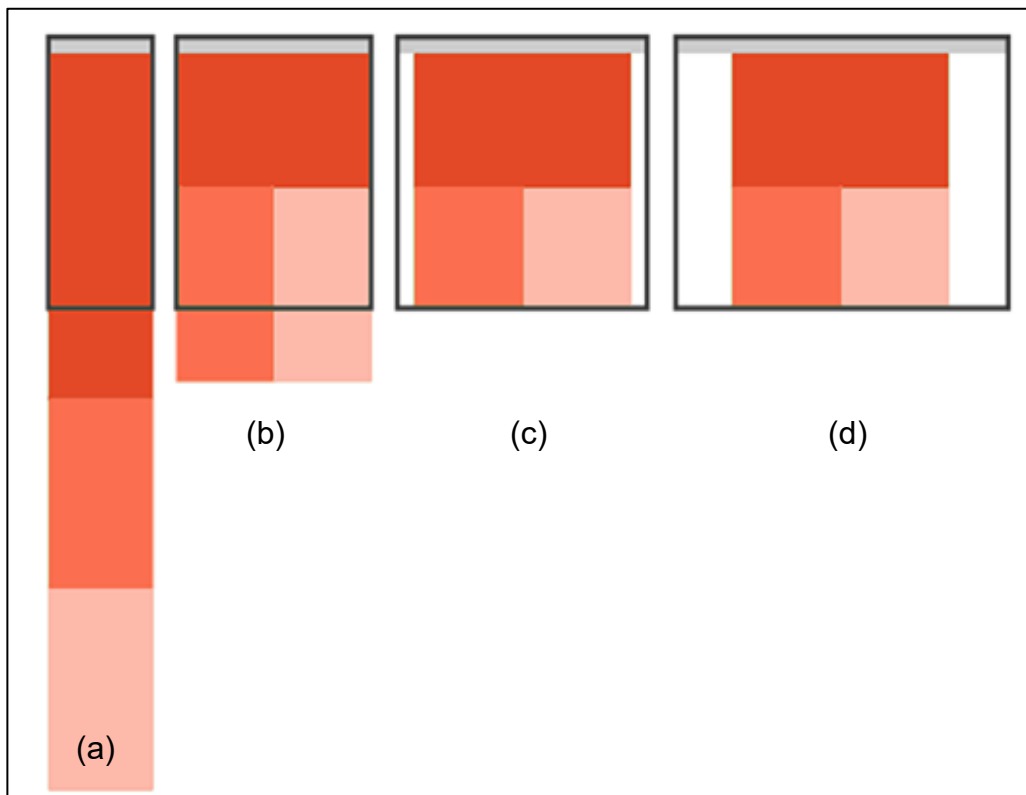


Figura 3.9 Esquema del patrón Mostly Fluid.

Obtenido de: (Azaustre, 2015).

JavaScript

JavaScript (Logo en la Figura 3.10) es el lenguaje de programación HTML y de la Web, hace que los ordenadores hagan lo que uno desee y es fácil de aprender (w3schools, JavaScript Tutorial, 2016).



Figura 3.10 Logotipo de JavaScript.

Obtenido de: (w3schools, JavaScript Tutorial, 2016).

jQuery

jQuery (Logo en la Figura 3.11) es una biblioteca rápida, pequeña y rica en funciones de JavaScript. Hace cosas como HTML, manejo de eventos, animaciones y Ajax mucho más simples con una API fácil de utilizar que trabaja con múltiples Navegadores (jquery, 2016).



Figura 3.11 Logotipo de JQuery.

Obtenido de: (virendra, 2018).

jQuery UI

Es un complemento que permite implementar componentes diversos para generar interfaces de usuario en páginas web, además de otras funcionalidades básicas para crear aplicaciones web enriquecidas. Como su propio nombre indica, está basado en el popular framework de Javascript. (Logo en la Figura 3.12) (Álvarez, 2010).

Entre los componentes que destacan de este framework son los temas e iconos que utiliza para generar las interfaces; éstos vienen prediseñados o a través de la página oficial diseñar el propio como fue el caso de este proyecto.



Figura 3.12 Logotipo jQuery UI.

Obtenido de: (jQuery, 2016).

Ajax

AJAX (Logo en la Figura 3.13) permite actualizar partes de una página Web mediante el intercambio de pequeñas cantidades de datos con el servidor en segundo plano sin necesidad de recargar la página entera, es una técnica utilizada para la creación de páginas Web rápida y dinámica (w3schools, AJAX Introduction, 2016).



Figura 3.13 Logotipo de JQuery.

Obtenido de: (virendra, 2018).

3.1.2 Back – End

Se refiere a toda la programación del lado Servidor, detrás del escenario, permitiendo que el usuario disfrute de su experiencia. Sin esto, el desarrollo llevado a cabo por el Front – End no se sostendría (campusMVP, 2015).

Dentro de este apartado se explicarán todas las herramientas ocupadas para el desarrollo del lado del servidor utilizando como lenguaje de programación C#.

ASP.net

Es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo; forma parte del .NET Framework.

El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc. (Microsoft, 2007). Su logo se puede apreciar en la Figura 3.14.

Para la realización de este proyecto se utilizará ASP.NET 4.6.1.



Figura 3.14 Logotipo ASP.net.

Obtenido de: (Microsoft, 2007).

C#

C# (Logo en la Figura 3.15) es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. Es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones permiten desarrollar aplicaciones rápidamente y mantener el estilo de los lenguajes de estilo de C (Microsoft, 2015).



Figura 3.15 Logotipo del lenguaje C#.

Obtenido de: (Cross Browser Testing, 2016).

Web Application

Es una plantilla proporcionada por ASP.net que proporciona la configuración del ambiente, herramientas, referencias y código inicial para desarrollar aplicaciones ASP.net. Podemos crear aplicaciones Web Forms, MVC, Web API y otro tipo de aplicaciones ASP. El desarrollo de esta aplicación será hecho en una aplicación Web Forms (TICapacitación, 2016).

Entity Framework (EF)

Es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Permite a los desarrolladores trabajar con datos en forma de objetos y propiedades específicos del dominio, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos; los desarrolladores pueden trabajar en un nivel mayor de abstracción cuando tratan con datos, y pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. Logotipo en la Figura 3.16.



Figura 3.16 Logotipo de Entity Framework.

Obtenido de: (Programming Apps, 2016).

Las aplicaciones que ocupan esta tecnología se pueden ejecutar en cualquier equipo en el que esté instalado .NET Framework a partir de la versión 3.5 SP1 (Microsoft, 2016). En general, EF es una técnica para convertir los modelos de Base de Datos como objetos y así poder utilizar un lenguaje de programación de bases de datos realizar operaciones CRUD (Create, Read, Update and Delete).

Controladores

Es el proceso que se ejecuta como respuesta a una solicitud realizada a una aplicación Web ASP.NET. Cuando los usuarios solicitan un archivo, la página procesa la solicitud a través del controlador de páginas. Puede crear sus propios controladores HTTP que representen el resultado personalizado en el explorador (Microsoft, 2010). En este proyecto se utilizará como el elemento de origen de datos para desplegar información hacia el usuario (como por ejemplo, tablas).

Microsoft SQL Server

Es un sistema creado por Microsoft para la administración y análisis de bases de datos relacionales para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. Para este proyecto se ocupará la versión 2016 (logo en la Figura 3.17) ya que mejoró las funciones críticas de la versión anterior de tal manera que proporciona rendimiento, disponibilidad y facilidad de uso. (Microsoft, 2014).



Figura 3.17 Logotipo de Microsoft SQL Server.

Obtenido de: (Microsoft, 2014).

Servicios Web

Es una tecnología que permite que las aplicaciones se comuniquen en una forma que no depende de la plataforma ni del lenguaje de programación. Es una interfaz de software que describe un conjunto de operaciones a las cuales se puede acceder por la red a través de mensajería XML estandarizada.

Usa protocolos basados en el lenguaje XML con el objetivo de describir una operación para ejecutar o datos para intercambiar con otro servicio Web. Un grupo de servicios Web que interactúa de esa forma define la aplicación de un servicio Web específico en una arquitectura orientada a servicios (SOA) (IBM, 2016).

Librerías

Es un archivo o conjunto de archivos que se utilizan para facilitar la programación. Las librerías consisten en archivos de código a los que llamamos al principio de la página, por ejemplo, una librería JavaScript será un archivo de extensión js que insertamos al principio de la página (ANYELGUTI, 2016).

En esta sección se explicarán las librerías más importantes que se ocuparán en el desarrollo de este proyecto.

ITextSharp

Es una librería (ver logo en Figura 3.18), de código abierto (Open Source) y específicamente para .Net, que permite crear y modificar

documentos PDF (Torres, 2014). Es una excelente librería, de libre de costo, que permite crear y modificar documentos PDF y añadirle todo tipo de información (incluyendo imágenes), lo que lo hace una herramienta, casi necesaria, en este desarrollo.



Figura 3.18 Logo de iTextSharp.

Obtenido de: (Nuget, 2016).

ZXing

Es una librería procesadora de imágenes multiformato en 1D/2D y de código abierto. Actualmente es capaz de reconocer los formatos UPC-A, UPC-E, EAN-8, EAN-13, Códigos 39, 93, 128, ITF, Codabar, RSS-14 (en todas sus variantes), Matriz de datos (Data Matrix), Aztec, PDF 417 y por supuesto los populares códigos QR, muy usados desde que la tecnología móvil está presente (Alvarez Tech, 2016).

En la Figura 3.19 se encuentra ilustrado su logotipo.

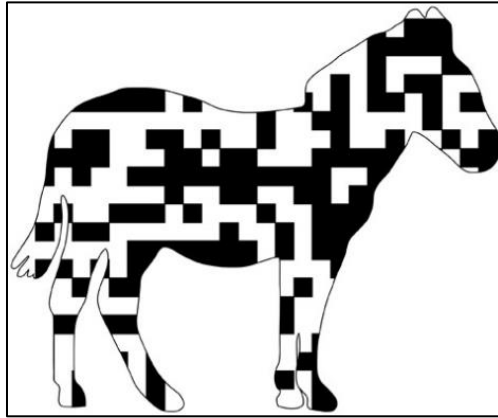


Figura 3.19 Logotipo ZXing.

Obtenido de: (Alvarez Tech, 2016).

Así pues, al enlistar los elementos a ocupar para la aplicación Web se da por concluida la parte correspondiente a este módulo y a continuación se describe, utilizando el mismo formato ocupado para esta sección, las herramientas para el desarrollo del siguiente elemento del sistema que es la aplicación Móvil.

3.2 Herramientas para Desarrollo de Aplicaciones Móviles

Para el desarrollo de la aplicación móvil se utilizó el entorno de desarrollo Android Studio, en el cual se desarrollan aplicaciones para dispositivos electrónicos como tabletas, celulares, televisiones o relojes que utilicen el Sistema Operativo Android.

Toda la aplicación móvil está diseñada bajo los estándares de Material Design y se utiliza el servicio de base de datos SQLite.

Aplicación Móvil

Mayormente conocida como App, es una aplicación de software que se instala en dispositivos móviles o tablets para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento, a diferencia de una Web App que no es instalable (QODE, 2012).

Las aplicaciones móviles están clasificadas en 3 tipos:

1. Aplicaciones Nativas:

Aquellas desarrolladas bajo un lenguaje y entorno de desarrollo específico, lo cual permite, que su funcionamiento sea muy fluido y estable para el sistema operativo que fue creada.

2. Aplicaciones Web (Web Application):

Esta modalidad consiste en la creación de un sitio o aplicación Web, optimizada para móviles. No necesariamente requiere de la creación de una aplicación instalable para dispositivos móviles. No se utilizarán características propias de los móviles como los sensores (acelerómetro, giroscopio) o cámara.

3. Aplicaciones Híbridas:

Son construidas usando tecnologías y estándares Web con los que el desarrollador probablemente ya esté familiarizado, bastante semejantes y compatibles con las que emplea al desarrollar aplicaciones Web, por ejemplo: HTML 5, CSS 3, Javascript. Para

soportar este modelo de desarrollo, existen frameworks como Apache Cordova, Adobe PhoneGap o Ionic Framework

Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA (Android, 2016). En la figura 3.20 se encuentra ilustrado el logotipo de Android Studio.



Figura 3.20 Logotipo de Android Studio.

Obtenido de: (Iber Mega, 2016).

Java

Java (ver logo en la Figura 3.21) es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos,

desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes (Oracle Corporation, 2016).



Figura 3.21 Logotipo de Java.

Obtenido de: (Oracle Corporation, 2016).

Para poder desarrollar aplicaciones en este lenguaje, es necesario tener instalado en la máquina a utilizar el JDK. JDK incluye Java Runtime Environment, el compilador Java y las API de Java (Oracle Corporation, 2016).

Lenguaje XAML

El lenguaje de marcado de aplicaciones extensibles, o XAML, es un lenguaje de marcado basado en XML desarrollado por Microsoft.

XAML (logotipo en figura 3.22) es el lenguaje que subyace a la presentación visual de una aplicación desarrollada en Microsoft Expression Blend, al igual que HTML es el lenguaje que subyace la presentación visual de una página web. La creación de una aplicación en Expression Blend supone tener que escribir código XAML, ya sea de forma manual o visual, mediante la vista Diseño de Expression Blend (Microsoft, 2016).



Figura 3.22 Logotipo del lenguaje XAML.

Obtenido de: (Microsoft, 2016).

Material Design

Es una guía integral para el diseño visual, de movimientos y de interacción en distintas plataformas y dispositivos. Android ahora es compatible con las aplicaciones de Material Design (Google, 2016).

En Android, se proporcionan los siguientes elementos que te permitirán crear aplicaciones en Material Design:

- Un tema nuevo;
- Nuevos widgets para vistas complejas;

- Nuevas API (interfaces de programación de aplicaciones) para sombras y animaciones personalizadas.

En cuestiones de temas, ofrece un nuevo estilo para las aplicaciones y widgets del sistema que permiten configurar la paleta de colores y animaciones predeterminadas para información táctil y transiciones de actividades.

Además de las propiedades X e Y, las vistas de Android ahora poseen una propiedad Z. Esta propiedad nueva representa la elevación de una vista, que determina lo siguiente:

- El tamaño de la sombra: las vistas con valores Z más elevados proyectan sombras más grandes.
- El orden del dibujo: las vistas con valores Z más elevados aparecen sobre las otras vistas.

Las nuevas API de animaciones permiten crear animaciones personalizadas para la información táctil en los controles de IU, además de realizar cambios en el estado de las vistas y transiciones entre actividades.

Estas API permiten:

- Responder a los eventos táctiles de las vistas mediante animaciones de información táctil;
- Ocultar y mostrar vistas con animaciones con efecto circular;
- Alternar entre actividades con animaciones personalizadas de transición de actividades;

- Crear animaciones más naturales con movimiento curvo;
- Animar los cambios en una o más propiedades de las vistas con las animaciones de cambio de estado de la vista;
- Mostrar animaciones en los elementos de diseño de la lista de estados entre los cambios de estado de las vistas.

Las animaciones de la información táctil se concentran en diferentes vistas estándar, como los botones.

Estas nuevas capacidades para los elementos de diseño permiten implementar aplicaciones de Material Design:

- Los asset se pueden escalar sin perder definición y son perfectos para los iconos de las aplicaciones de un solo color.
- El teñido de los asset permite definir mapas de bits como máscaras alfa y pintarlos con un color durante el tiempo de ejecución.
- La extracción de color permite extraer automáticamente colores prominentes de una imagen del mapa de bits.

En la Figura 3.23 se encuentran ejemplos de pantallas de una aplicación creada utilizando el patrón de diseño de Material Design. En esta se puede apreciar el uso de los assets en los íconos, el conjunto de colores y las nuevas animaciones para los menús desplegables.

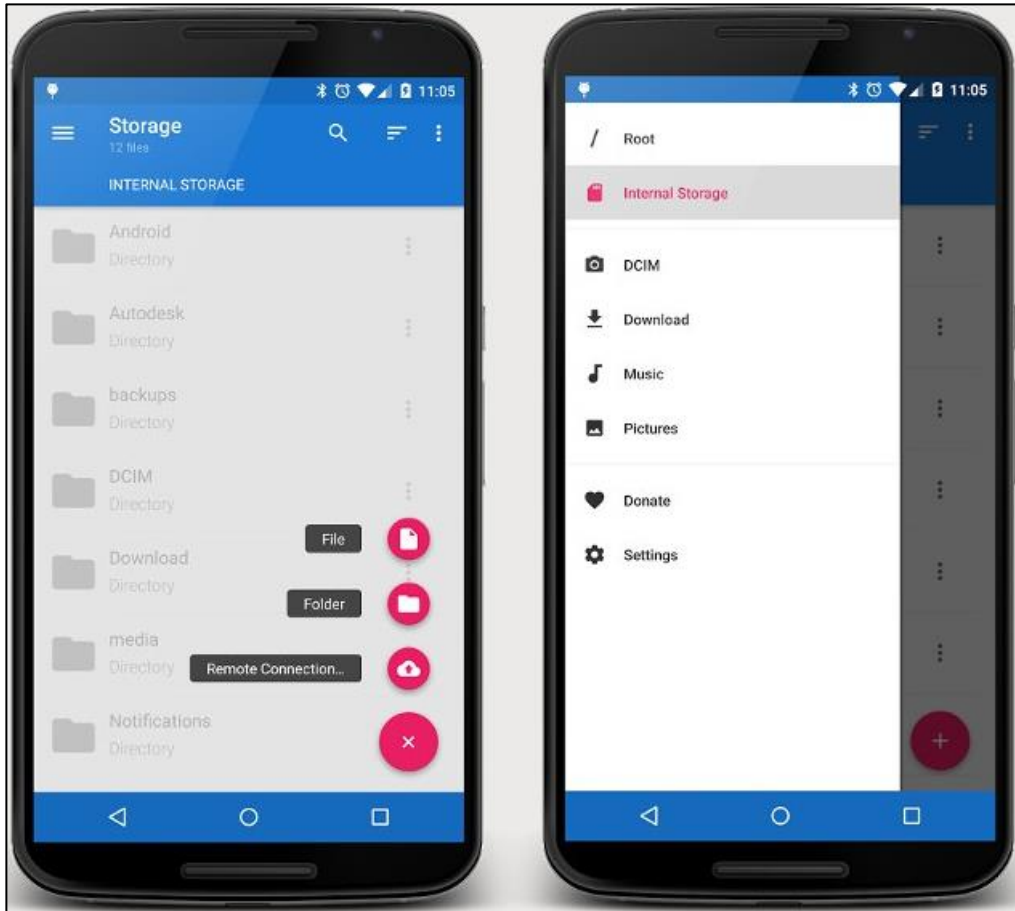


Figura 3.23 Ejemplo de pantallas creadas con Material Design.

Obtenido de: (Kaudal, 2016).

SQLite

SQLite (logotipo en la Figura 3.24) es una biblioteca en proceso que implementa autonomía, sin servidor, sin configuración y motor de base de datos SQL transaccional (SQLite, 2016).

SQL es opensource (código abierto al público), por lo tanto es libre para cualquier propósito comercial o privado.



Figura 3.24 Logotipo de SQLite.

Obtenido de: (SQLite, 2016).

Interoperabilidad entre sistemas

La interoperabilidad entre lenguajes es la posibilidad de que el código interactúe con código escrito en un lenguaje de programación diferente. La interoperabilidad entre lenguajes puede ayudar a maximizar la reutilización de código y, por tanto, puede mejorar la eficacia del proceso de programación.

Los compiladores y las herramientas de lenguaje dirigidos a Common Language Runtime se benefician de la compatibilidad que integra el motor en tiempo de ejecución para la interoperabilidad entre lenguajes.

Common Language Runtime ofrece la base necesaria para la interoperabilidad entre lenguajes al especificar e imponer un sistema de tipos común, y al suministrar metadatos.

Debido a que todos los lenguajes dirigidos a Common Language Runtime siguen las reglas del sistema de tipos común para definir y utilizar los tipos, la utilización de tipos es coherente entre todos los lenguajes.

Los metadatos habilitan la interoperabilidad entre lenguajes mediante la definición de un mecanismo uniforme para almacenar y recuperar la información sobre tipos.

Los compiladores almacenan la información sobre tipos como metadatos y Common Language Runtime usa esta información para proporcionar servicios durante la ejecución; el motor en tiempo de ejecución puede administrar la ejecución de aplicaciones de varios lenguajes porque toda la información de tipos se almacena y recupera de la misma forma, independientemente del lenguaje en que se haya escrito el código (Microsoft, 2010).

Ksoap2

kSOAP2 es una biblioteca de peso ligero para uso en dispositivos limitados. No pretende ser un do-it-all biblioteca abarca todo. Sin embargo, está destinado a interactuar con la mayoría de los motores de SOAP populares (KSOAP, 2014).

En la Figura 3.25 se explica el funcionamiento de kSOAP.

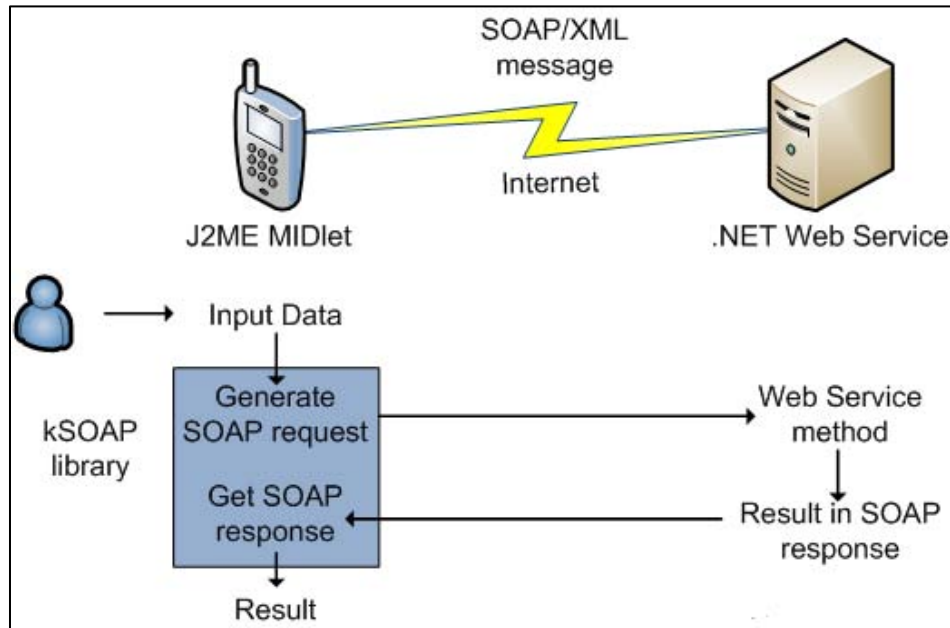


Figura 3.25 Funcionamiento de kSOAP.

Obtenido de: Elaboración propia.

3.3 Control de versiones

Un sistema de control de versiones (o sistema de control de revisiones) es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas Web.

Actualmente todo el mundo espera que al menos el código fuente del proyecto esté bajo un control de versiones y probablemente no se tomen el proyecto seriamente si no se utiliza este sistema con un mínimo de competencia (Anónimo, Control de Versiones, 2016).

Clasificación

Los sistemas de control de versiones se dividen en 3 grupos: Local, centralizado y distribuido.

Los sistemas locales son aquellos en que los archivos se guardan en un repositorio dentro de la computadora del host o desarrollador.

Usualmente contienen en una base de datos los registros de todos los cambios realizados sobre los archivos (GIT, 2009). Esto se puede ver explicado gráficamente en la Figura 3.26.

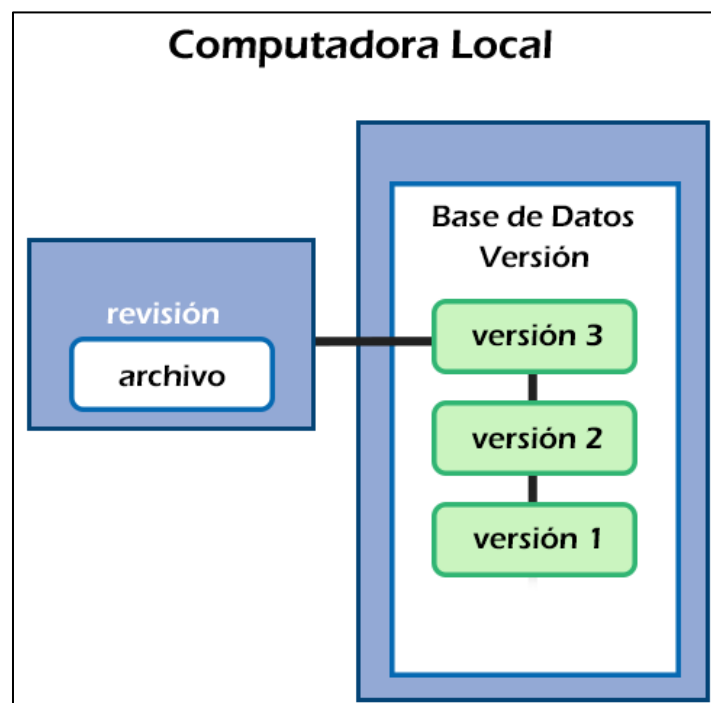


Figura 3.26 Control de versiones local.

Obtenido de: Elaboración propia.

El tipo CVCSs (Centralized Version Control Systems) o centralizados se utiliza cuando se va colaborar con desarrolladores en otros sistemas.

Estos sistemas tienen un único servidor que contiene todos los archivos versionados, y varios clientes que descargan los archivos desde ese lugar central. Durante muchos años éste ha sido el estándar para el control de versiones. En la Figura 3.27 se explica el funcionamiento del CVCS.

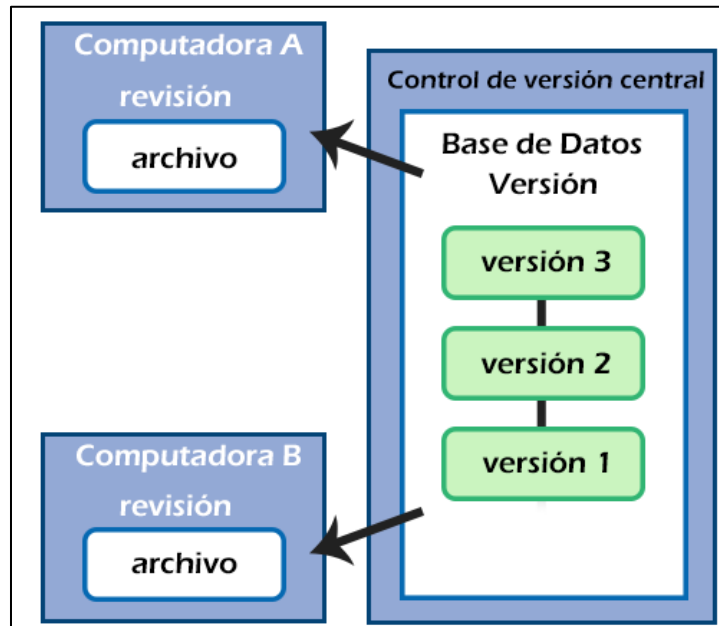


Figura 3.27 Control de versiones centralizadas.

Obtenido de: Elaboración propia.

Una de las ventajas de esta configuración es que todo el mundo puede saber en qué están trabajando los otros colaboradores del

proyecto. Los administradores tienen control detallado de qué puede hacer cada uno y no se tiene que lidiar con bases de datos locales en cada cliente.

Sin embargo, la mayor ventaja radica en que si este servidor se cae durante una hora, entonces durante esa hora nadie puede colaborar o guardar cambios versionados de aquello en que están trabajando (GIT, 2009).

En los sistemas de control de versiones distribuidos los clientes no sólo descargan la última versión de los archivos: Replican completamente el repositorio. Así, si un servidor muere, y estos sistemas estaban colaborando a través de él, cualquiera de los repositorios de los clientes puede copiarse en el servidor para restaurarlo.

Cada vez que se descarga una instantánea, en realidad se hace una copia de seguridad completa de todos los datos. Este tipo es el que se utilizará para el desarrollo de este sistema.

Las ventajas de utilizar este tipo de sistema son las siguientes:

- Diseñado para manejar proyectos grandes en equipo.
- Autenticación criptográfica del historial.
- Formato sencillo y compacto.
- La herramienta se encuentra en el entorno de desarrollo elegido.
- Se pueden realizar cambios de manera local sin estropear el trabajo de los demás.

El diagrama de funcionamiento se encuentra descrito en la Figura 3.23 (GIT, 2009).

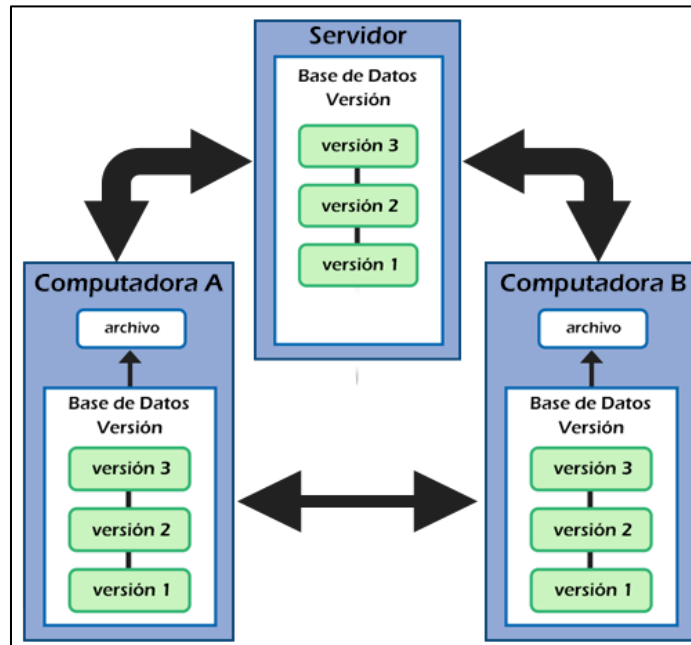


Figura 3.28 Control de versiones distribuido.

Obtenido de: Elaboración propia.

3.3.1 GIT

Git (logotipo en la Figura 3.29) es un sistema de control de versiones distribuido; fue impulsado por Linus Torvalds y el equipo de desarrollo del Kernel de Linux. Git es multiplataforma, por lo que puedes usarlo y crear repositorios locales en todos los sistemas operativos más comunes, Windows, Linux o Mac.

Existen multitud de GUIs para trabajar con Git, no obstante para el aprendizaje se recomienda usarlo con línea de comandos (Alcázar & Álvarez, 2014).

Git modela sus datos como un conjunto de instantáneas de un sistema de archivos. Cada vez que confirmas un cambio, o guardas el estado de un proyecto en Git, él guarda el aspecto de todos los archivos en ese momento, y guarda una referencia a esa instantánea.

Para ser eficiente, si los archivos no se han modificado, Git no almacena el archivo de nuevo, sólo un enlace al archivo anterior idéntico que ya tiene almacenado (GIT, 2009).



Figura 3.29 Logotipo del control de versiones git.

Obtenido de: (GIT, 2009)

Git tiene tres estados principales en los que se pueden encontrar los archivos:

- **Confirmado (committed):** Significa que los datos están almacenados de manera segura en la base de datos local.
- **Modificado (modified):** Significa que se ha modificado el archivo pero todavía no lo han confirmado a la base de datos.

- Preparado (staged): Significa que han marcado un archivo modificado en su versión actual para que vaya en la próxima confirmación.

Esto lleva a las tres secciones principales de un proyecto de Git: el directorio de Git (Git directory), el directorio de trabajo (working directory), y el área de preparación (staging area). En la Figura 3.30, se muestra un diagrama acerca del flujo de datos de la información.

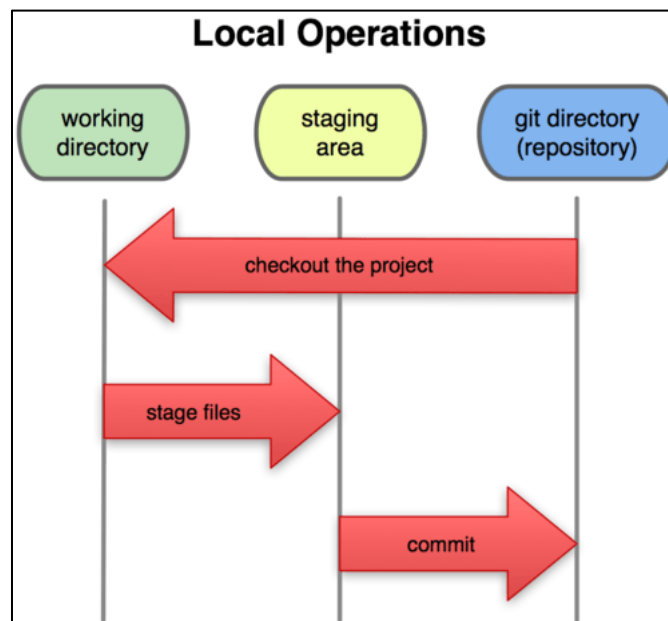


Figura 3.30 Diagrama de flujo de información en Git.

Obtenido de: (GIT, 2009)

El directorio de Git es donde se almacenan los metadatos y la base de datos de objetos para un proyecto.

El directorio de trabajo es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que se puedan usar o modificar.

El área de preparación es un sencillo archivo, generalmente contenido en el directorio de Git, que almacena información acerca de lo que va a ir en la próxima confirmación (GIT, 2009).

3.3.2 Bonobo GIT Server

En un servidor de Git alojado en IIS para Windows. Este programa permite mantener un control total sobre los repositorios y usuarios a través de una red interna y con una interfaz gráfica amigable. Es un proyecto de código abierto que está disponible con una licencia MIT. El código fuente está disponible en GitHub y es muy extensible - se puede influir en su futuro. Libre para Windows (Bonobo Git Server, 2014). En la Figura 3.31 se muestra el logo de esta aplicación.



Figura 3.31 Logotipo del servidor Bonobo Git server.

Obtenido de: (Bonobo Git Server, 2014)

3.4 Sistemas Distribuidos

Los sistemas distribuidos son "sistemas cuyos componentes hardware y software, que están en computadoras conectadas en red, se comunican y coordinan sus acciones mediante el paso de mensajes, para el logro de un objetivo. Se establece la comunicación mediante un protocolo preestablecido" (Cinvestav, 2016). En la Figura 3.32 se muestra un diagrama que ilustra la estructura de un sistema distribuido.

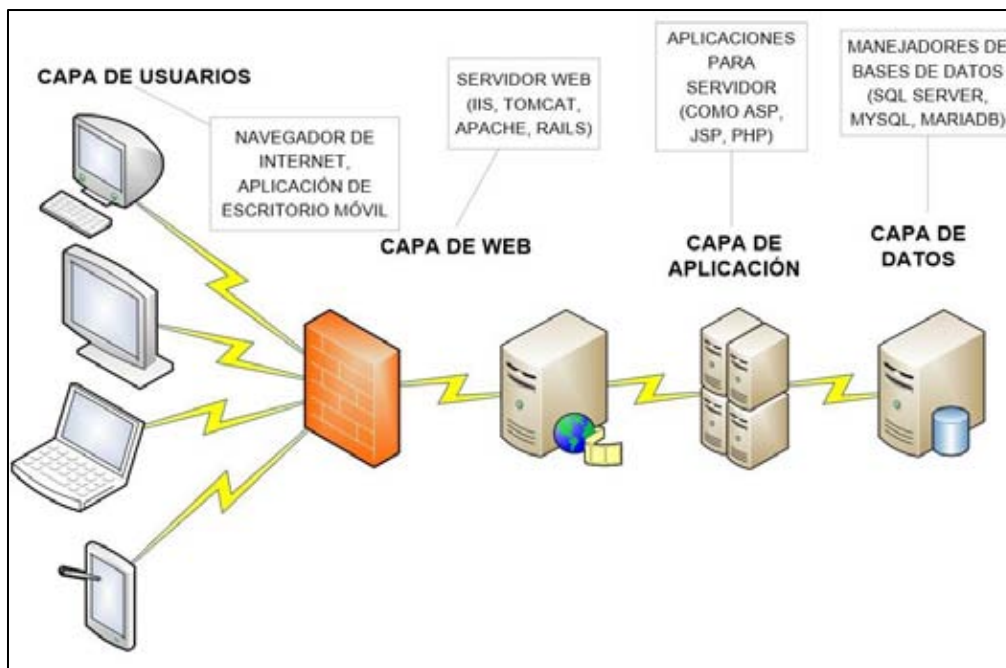


Figura 3.32 Esquema de un sistema distribuido en modelo 4 capas.

Obtenido de: Elaboración propia.

Un sistema distribuido que pretenda ofrecer una visión de sistema único deberá cumplir con 3 propiedades. (sc.ehu, 2016)

- **Transparencia:** Proporcionar al usuario y a las aplicaciones una visión de los recursos del sistema como gestionados por una sola máquina virtual.
- **Escalabilidad:** Su modularidad, lo que le permite una gran flexibilidad y posibilita su escalabilidad, definida como la capacidad del sistema para crecer sin aumentar su complejidad ni disminuir su rendimiento.
- **Fiabilidad y Tolerancia a fallos:** Capacidad para realizar correctamente y en todo momento las funciones para las que se ha diseñado.

CAPITULO 4:

Desarrollo e Implementación

CAPÍTULO IV. DESARROLLO E IMPLEMENTACIÓN

En el presente capítulo se describe el proceso de desarrollo del sistema de control de inventario, el cual se ha dividido según las fases de la metodología de sistemas distribuidos, lo que da como resultado las secciones:

- **Especificaciones de Alto Nivel:** En este apartado se presentan los requisitos del cliente y se describe el funcionamiento de todo el sistema distribuido, a partir de las especificaciones dadas.
- **Estándares:** Se especifican los objetivos de coordinación y técnicos con los que se trabaja, con esto mantener una administración correcta de la manera de trabajo entre los integrantes del equipo.
- **Herramientas de Diseño:** Se describen los patrones de diseño elegidos, así como los diagramas que modelan el comportamiento del sistema, la forma de comunicación y también la arquitectura de la red que será el ambiente del sistema.
- **Documentación:** Se describirán los aspectos técnicos del sistema como: el sistema gestor de Base de Datos a utilizar, características del hardware, los lenguajes seleccionados, entre otros atributos.
- **Implementación:** Se mostrará el funcionamiento de cada una de las aplicaciones que conforman el sistema.

Cabe mencionar que todas las figuras que se encuentran en este capítulo son elaboración propia.

4.1 Especificaciones de Alto Nivel

Siguiendo las fases de la metodología, como primer paso es recabar información necesaria a fin de describir los objetivos y descripción detallada acerca del Sistema Distribuido. Para esto, se realizó una entrevista al Supervisor del departamento de Programación y Equipos de la ZOTGM, a la par de contar con la asesoría del Profesionista de Programación de este centro de trabajo y así, obtener una descripción detallada y correcta del producto esperado.

Se realizó una entrevista al Ing. Alejandro Mendoza Viveros, supervisor del departamento de Programación y Equipos y así conocer las necesidades que se tienen respecto a la actualización del Sistema de Control de Inventario. En el Anexo A. Cuestionario de especificaciones del sistema, se muestra un conjunto de reactivos y respuestas aplicadas en donde se deduce que la aplicación Web, además de lo que actualmente realiza, necesita incluir nuevas operaciones, generar reportes pertinentes al usuario, así como implementar seguridad para que solamente el personal autorizado pueda modificar la información almacenada en la base de datos.

Para la realización de este proyecto es necesario especificar las operaciones que se realizarán en cada una de las aplicaciones que conformarán este sistema, específicamente el módulo móvil ya que sólo operaciones esenciales se podrán hacer a través de éste para no saturar de procesos el dispositivo.

En la Figura 4.1 se especifica cada una de las operaciones que podrán ser accedidas desde el sistema Web y el sistema Móvil.

El Sistema Web será el elemento activo en el proyecto, debido a que se pueden realizar todas las operaciones disponibles. El elemento pasivo será el Sistema Móvil ya que solamente ciertas operaciones podrán ser realizadas a través de la aplicación.

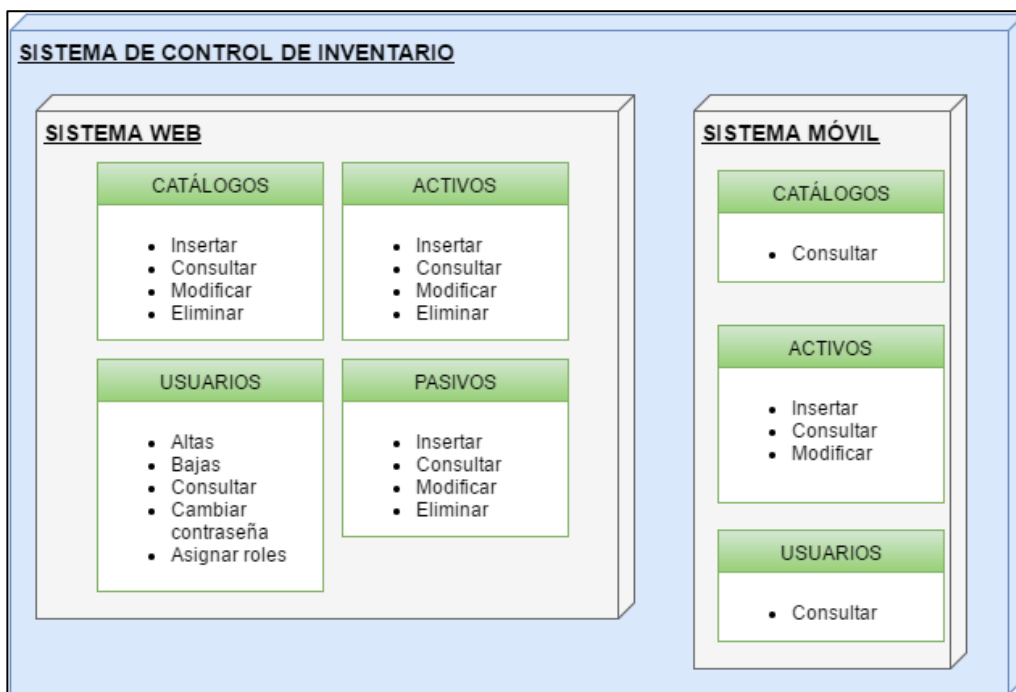


Figura 4.1 Especificación de operaciones en cada sistema.

Una vez definidas las operaciones que se realizarán en cada uno de los sistemas que componen este proyecto, se describen las acciones que pueden realizar cada uno de los usuarios que participarán en esta aplicación, basados en las especificaciones dadas por el supervisor de la zona. En la Figura 4.2 se muestra un diagrama de contexto de las operaciones que cada uno de los tipos de usuario podrán realizar.

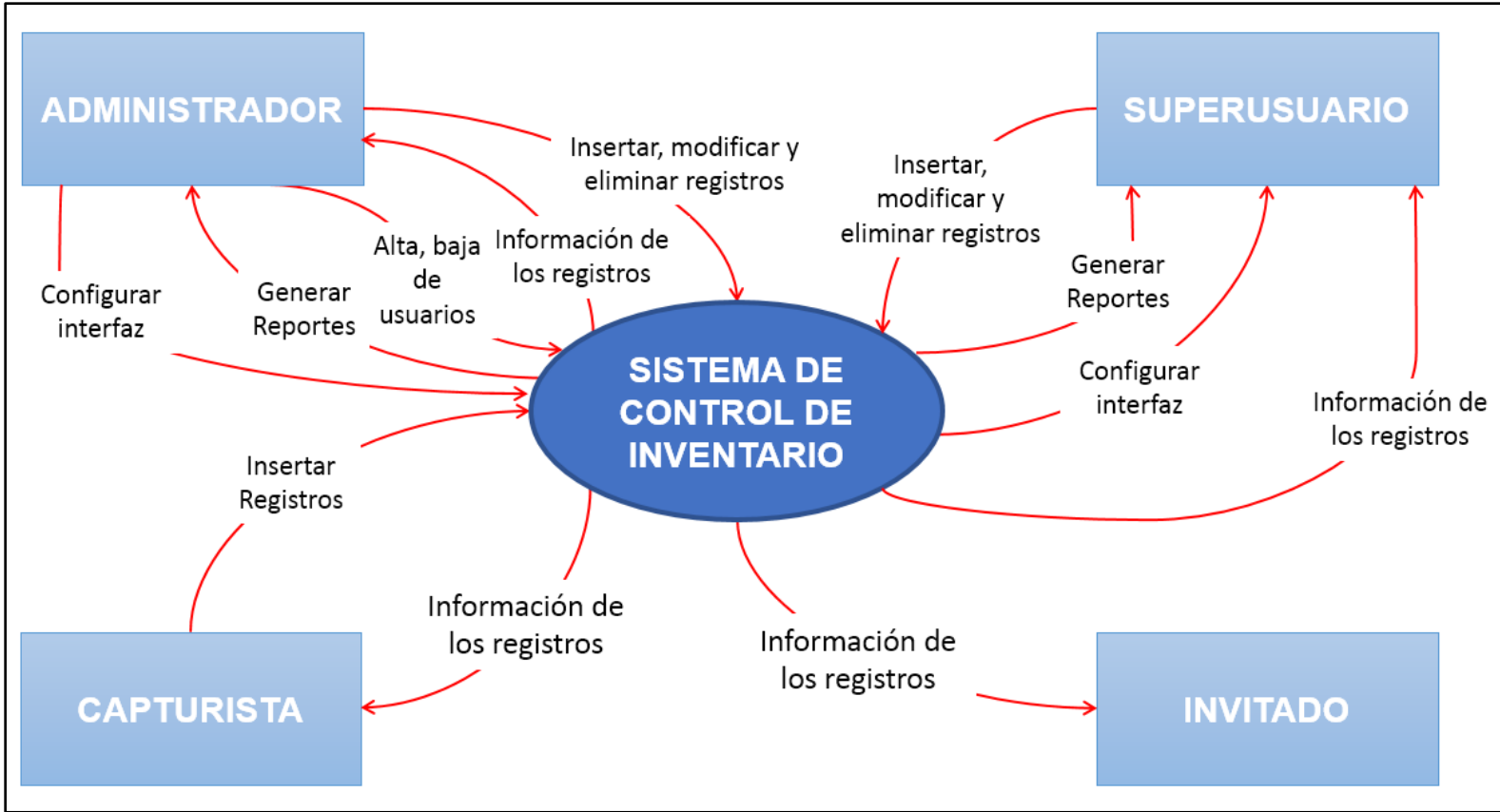


Figura 4.2 Diagrama de Contexto.

Utilizando el modelo clásico de los sistemas distribuidos y la arquitectura de 3 niveles (o modelo 3 capas), se explica el funcionamiento del proyecto. En la Figura 4.3 se representa la implementación del sistema, donde se visualiza como la interacción entre cada una de las capas.

La base de datos creada en SQL Server contiene la información almacenada por el usuario. Dentro del dispositivo móvil existe un duplicado de las tablas que funcionarán como catálogos para poder agregar información a las tablas padres o entidades fuertes; mismas que mantendrán la organización en la información y se guardarán en el dispositivo sin ocupar gran cantidad de espacio de almacenamiento, se utilizará el servicio de SQLite, siendo esto la capa conocida como datos o almacenamiento que corresponde al primer nivel de la arquitectura.

Para poder acceder a sus registros, se usarán Servicios Web, que serán los encargados de suplir las peticiones enviadas por la aplicación Web y Móvil. Al utilizar este tipo de arquitectura se aplica seguridad a los datos puesto que no existe una comunicación directa de las aplicaciones con la base de datos de SQL Server, por lo tanto el servicio Web figura como un intermediario para gestionar las peticiones. Esta capa corresponde al procesamiento de información o de negocios correspondiente al segundo nivel de la arquitectura.

Continuando con la arquitectura, en su tercer nivel está la capa de presentación o de interfaz de usuario. Esta se implementa bajo la aplicación Web y móvil, ya que presentará la información obtenida del Servicio Web al usuario.

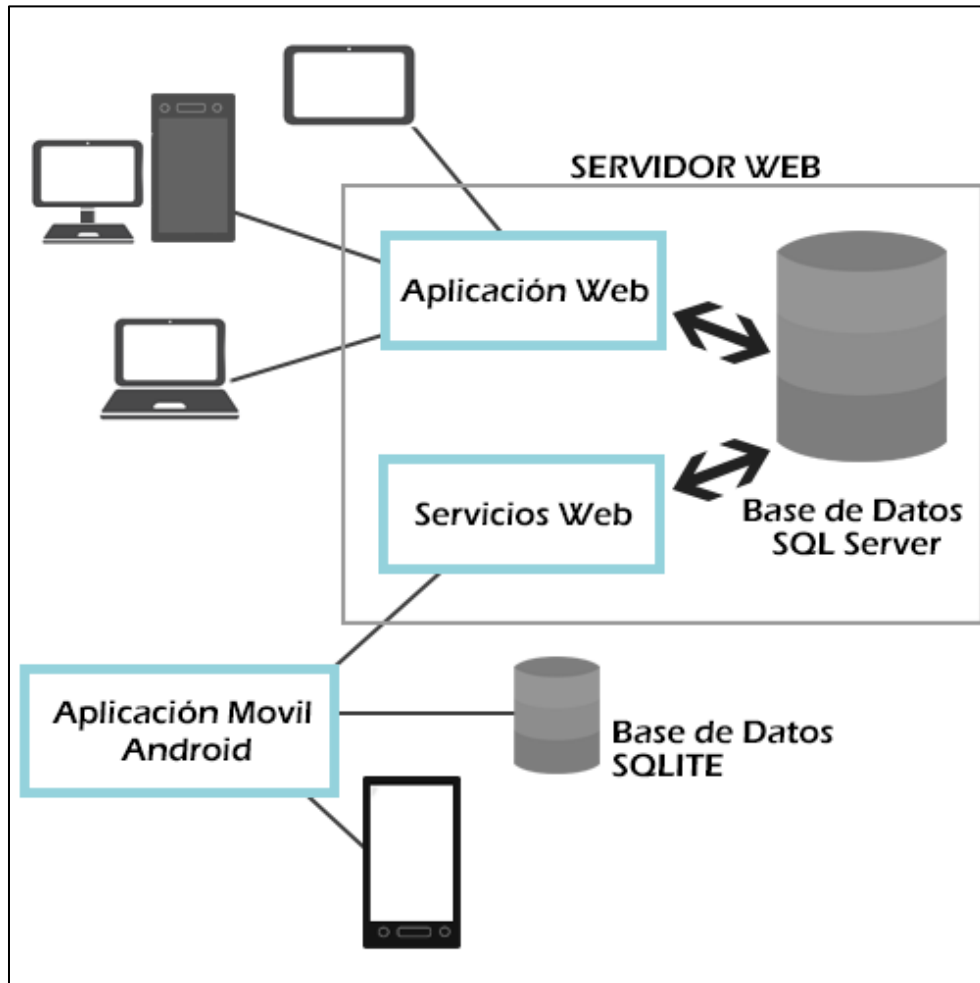


Figura 4.3 Implementación del sistema distribuido.

Teniendo en claro los requisitos del cliente, las operaciones que pueden realizar cada tipo de usuario y la manera en que interactúa cada uno de los elementos que componen el sistema, se continúa hacia la siguiente etapa de la metodología, que consiste en establecer estándares y patrones para el correcto desarrollo del proyecto y así reducir considerablemente los errores en tiempos de trabajo.

4.2 Estándares

El trabajar bajo el formato de programación usando estándares tiene como propósito establecer reglas y normas para fomentar las buenas prácticas de codificación. Para lograr esto se realizó un consenso de trabajo, estableciendo cada uno de los puntos que se presentan.

Para determinar el procedimiento de desarrollo de código, se opta por utilizar estándares de codificación pertenecientes a los lenguajes C# y JAVA que son los utilizados en este proyecto.

Partiendo de la existencia de tres tipos de nomenclatura: Camel Case (C), Pascal Case (P) y Lower Case (L), se especifica cual se utiliza en los nombres de los archivos, funciones, métodos y variables entre otros elementos (ver tabla 4.1).

Tabla 4.1 Convenciones de nomenclaturas.

Identificador	Nomenclatura	Nota
Archivo de proyecto	Pascal Case	Para el caso de imágenes, fuentes y otros archivos que no sean de código fuente
Archivo de códigos fuentes	Pascal Case	Código de Java, .Net, javascript, hojas de estilos, servicios web, etc.
Namespace	Pascal Case	Para el caso de .Net
Clases	Pascal Case	Para el caso de .Net y Java.

Identificador	Nomenclatura	Nota
Métodos (javascript)	Camel Case	Los que se encuentran en archivos con extensión js.
Identificadores de variables	Lower Case	Nombres de todas las variables declaradas, separadas con un '_'. En caso de ser información contenida en un elemento Web o Móvil se pondrá el nombre del elemento como primer lugar, por ejemplo: boton_aceptar.
Instancias de Objeto	Camel Case	Para cualquier lenguaje de programación
Parámetros	Camel Case	Cualquier parámetro declarado en funciones
Eventos	Camel Case	En cualquier lenguaje.
Paquetes	Lower Case	Para el caso de Java. Se utiliza el prefijo com.zotgm.controlinventario.
Layouts	Lower Case	Las palabras están separadas con un '_'.
Servicios Web	Camel Case	Se utilizará el prefijo WS seguido de la operación que agrupará ese servicio, por ejemplo: WSInsertar. Los que se comunicarán con la aplicación móvil estarán contenidos en uno solo llamado WSAndroid.

Identificador	Nomenclatura	Nota
Métodos Web	Camel Case	La estructura a seguir es utilizar como prefijo el nombre de la operación seguido del nombre de la tabla, por ejemplo: InsertarUsuario.
Controladores	Camel Case	La estructura a seguir es ocupar el prefijo Data seguido de la tabla correspondiente, por ejemplo DataUsuarios.

En la Tabla 4.2 se listan los estilos de codificación que se utilizan en el proyecto tanto en la aplicación Web como la Móvil, destacando elementos de código como las llaves, comentarios, variables, excepciones, entre otros.

Tabla 4.2 Estilos de codificación.

Código	Estilo
Llaves {}	Para .Net: Siempre en una nueva línea. Para Java, javascript y CSS: Seguido del nombre de la función, evento o método.
Tabulación	Espacio de 4 caracteres.
Comentarios	Para todo se utilizará /* ... */

Código	Estilo
Variables	Para .Net: grupo de variables del mismo tipo y en el caso de las clases, una variable para cada declaración. Para Javascript: Una variable para cada declaración. Para Java: Grupo de variables del mismo tipo.
Propiedades	Utilizar la sintaxis: {get; set;}
Condicionales	Utilizar igualación con datos booleanos; por ejemplo, variable == true.
Manejo de Excepciones	Se utilizan cuando se ejecutan operaciones a servicios externos, por ejemplo: bases de datos y servicios Web.
This	Se utiliza para los eventos de los botones en Java.
Ciclos	Se utiliza foreach cuando se tenga que recorrer los registros en estructuras de datos en ambos lenguajes. Se utiliza switch cuando se realizarán casteos en Java.
Final	Se utiliza en Java cuando esa variable sólo se le puede asignar un valor u objeto una única vez.
Método finish	En Java se utiliza para finalizar actividades como en el Logueo de la aplicación.

Para la sincronización del proyecto y el control de versiones se utiliza Git, que tiene como finalidad establecer los métodos necesarios para la correcta transferencia de archivos, en compañía de Bonobo que funciona como un servidor de repositorios de proyectos.

Las reglas para poder sincronizar cambios serán las siguientes:

- Únicamente se sincronizarán aquellos cambios que estén correctos y completos, para no dañar la estructura del proyecto.
- Sólo una persona llevará el control de la base de datos.
- Se deberán especificar en los comentarios el cambio que se realizó en el proyecto. No se pueden subir cambios sin comentarios.
- Si se estará probando una nueva función, se deberá realizar una rama local y una vez que se haya realizado la prueba exitosamente, se deberá incorporar ese cambio a la rama master.

Con relación a los patrones de diseño del proyecto, se utilizó el llamado SOLID, esto debido a que ofrece principios básicos y sencillos para la correcta creación de las interfaces de usuario. En el ámbito del diseño responsivo, se utilizó el patrón Mostly Fluid, ya que la interfaz la acomoda como una grilla y de este modo, puede acomodar en base a columnas la información sin tener pérdida de contenido.

En la tabla 4.3 se muestra la paleta de colores propuesta y avalada por los diseñadores, así como por los usuarios que solicitan el sistema adecuándose a los colores de la institución, así se asegura la estandarización y uniformidad en el diseño de interfaz del proyecto. Las columnas que lo conforman son:

- Muestra del color.
- Nombre del color.
- Valor en código hexadecimal.

- Descripción de en qué elemento se ocupa este color principalmente.

Tabla 4.3 Paleta de colores del proyecto.

Muestra	Color	Hexadecimal	Descripción
	Verde claro	#9dea73	Utilizado para resaltar texto.
	Gris oscuro	#333333	Utilizado en la barra de navegación y las tablas en la herramienta.
	Blanco	#000000	Utilizado para el fondo del sitio Web.
	Amarillo claro	#fff39b	Utilizado para resaltar las filas de la tabla.
	Azul	#51a5b8	Utilizado para el icono que duplicar y edita registros.
	Rojo	#ee1c30	Utilizado para el icono que elimina registros.
	Gris	#717171	Utilizado para los títulos en los catálogos.
	Gris claro	#e7e7e7	Utilizado para intercalado de filas en las tablas de los catálogos.

Por último, dentro del proyecto se contemplaron el uso de iconos Minimalistas.

Este patrón se basa en que se deben de utilizar iconos que no contengan una gama muy extrovertida de colores, estas imágenes contendrán un diseño simple y con colores preferiblemente en gris, esta combinación de iconos no satura la interfaz del usuario haciéndola visualmente más atractiva, se puede ver uniforme y debido a su tamaño, utilizan poco espacio en el sitio Web.

En la Figura 4.4 se muestra el asset de todos los iconos minimalistas utilizados en el proyecto. Estos han sido descargados con licencia gratuita (no son generación propia).



Figura 4.4 Asset de los iconos de la aplicación Web.

En la Figura 4.5 se muestran los diseños de los iconos de la aplicación Móvil. Cabe mencionar que estos son los que se encuentran por defecto para la plataforma Android, por lo tanto el uso de estos es totalmente gratuito.

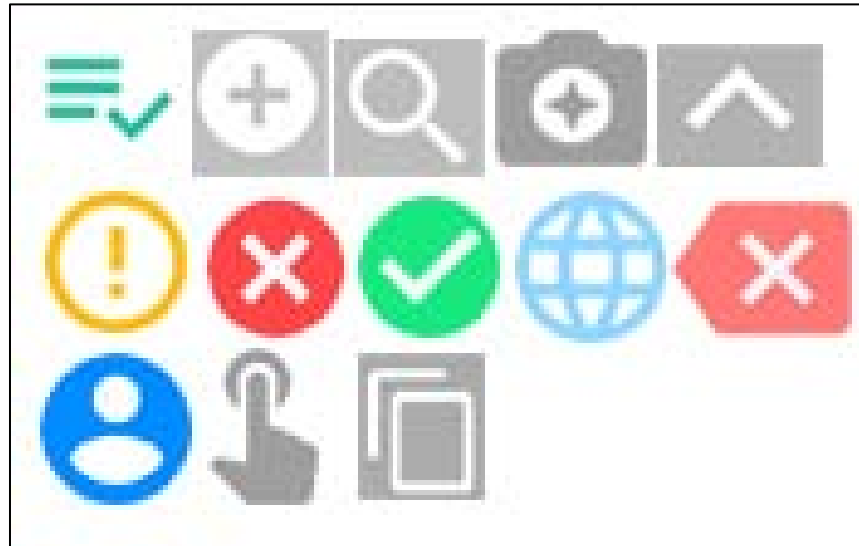


Figura 4.5 Asset de los iconos de la aplicación Móvil.

Al término de la descripción y especificando los estándares, ya se tiene claro el aspecto ergonómico del sistema a implementar; por lo que a continuación se describirán las herramientas de diseño a implementar.

4.3 Herramientas de Diseño

Existen programas que simplifican el proceso de diseño de aplicaciones distribuidas como CORBA, DCE o DCOM. Sin embargo, utilizando diagramas UML y algunas herramientas proporcionadas por el IDE Visual Studio se podrá diseñar de manera óptima los procesos realizados por el Sistema de Control de Inventario.

Modelo de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenimiento (Salinas, 2006).

Un diagrama de clases está compuesto por los siguientes elementos:

- Clase: Atributos, métodos y visibilidad.
- Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

En la Figura 4.6 se encuentra un ejemplo de la estructura de una clase.

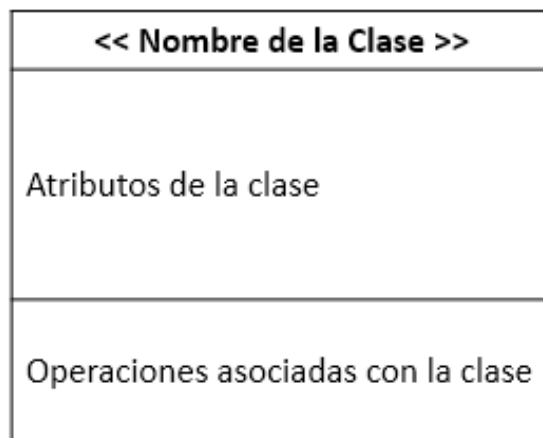


Figura 4.6 Ejemplo de estructura de una clase.

Debido de la extensión del diagrama de clases, se mostrará un diagrama general con las clases padres del proyecto Web y después se desglosará en otro diagrama cada una de las clases que conforman la clase padre.

En la Figura 4.7 se muestra el diagrama con las clases padre que integran el proyecto Web.

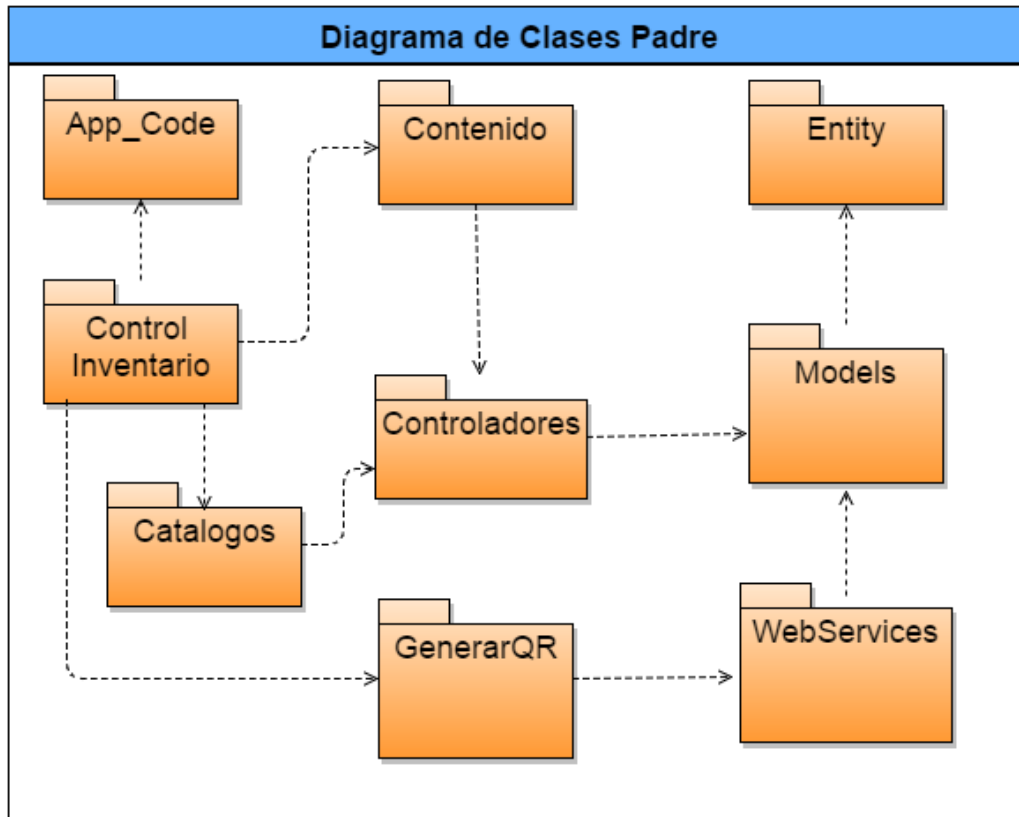


Figura 4.7 Diagrama de clases padre.

En la tabla 4.4 se describen las funciones de cada una de las clases padre representadas en la Figura 4.7 con el fin de conocer los elementos que contiene a cada una de las clases y la función a desempeñar.

Tabla 4.4 Descripción de las clases padre

Nombre de la Clase	Descripción
Control Inventario	Es la clase principal de todo el proyecto. En esta se encuentra el Site Master de la aplicación (encabezado, menú y pie de la página que son elementos estáticos) y configuraciones generales.
App_Data	En esta clase se almacena las configuraciones iniciales de los encabezados del sitio, es decir, los títulos y las imágenes.
Contenido	Contiene los accesos a todas las páginas principales del sitio (Activo, Pasivo, Pase de Salida, Administrar Usuarios, Configuración, Inicio, Login y Ayuda).
Catálogos	Contiene todos los catálogos del proyecto (Marca, Modelo, Estado de Registro, Grupo, Centro de Trabajo, Usuario, Tipo Activo y Ubicación).
Controladores	Encargados de responder las solicitudes de los usuarios a través del HTTP. Usualmente se utiliza para acceder a la información de los catálogos y tablas de la base de datos.

Nombre de la Clase	Descripción
WebServices	Se utilizan para realizar las operaciones CRUD (Create, Read, Update y Delete) del proyecto Web y Móvil.
GenerarQR	Contiene las clases necesarias para generar QR a partir de un registro en la Base de Datos.
Models	Se encuentran los modelos necesarios para la autenticación. Para este proyecto se ocupa la autenticación Identity.
Entity	Se encuentran los modelos de Entity Frameworks que representan las tablas de la Base de Datos como clases.

Los diagramas de clase que se muestran a continuación han sido extraídos desde la herramienta que proporciona el IDE Visual Studio para realizar diagramas de clases de los proyectos.

Para el caso de los diagramas de clases de la aplicación móvil, han sido extraídos desde una herramienta (plugin) gratuito que proporciona el IDE de Android Studio llamada Simple UML CE.

En la Figura 4.8 se muestra la estructura del diagrama de clases de este IDE, que contiene nuevas características comparando con la estructura tradicional.

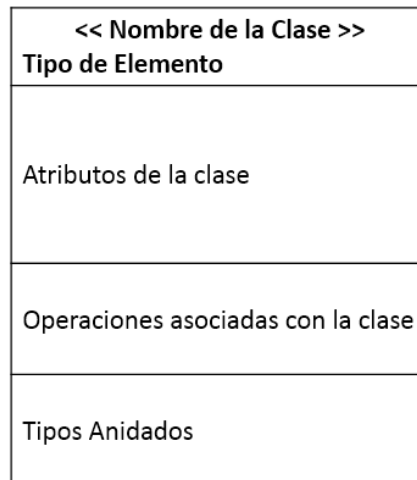


Figura 4.8 Estructura de una clase en Visual Studio.

En la Figura 4.9 se muestran las clases dentro de contenido. Aquí se encuentran las clases para realizar el Login, la Configuración, la Administración de los Usuarios, Pase de Salida, Activos, Pasivos, Otros y Acerca de (About).

Cada una de las clases contiene los campos, los métodos que utilizan y los tipos anidados. En cada una se puede notar que existe el método Page_Load, esto porque todas las páginas creadas en ASP lo tienen por defecto para poder cargar los elementos provenientes o alguna otra función del servidor al momento de abrir la página. También existen clases que solamente cuentan con el método por defecto; esto se debe a que las comunicaciones entre éstas se realizan a través de un control o un servicio Web, siguiendo la arquitectura del modelo de capas.

Los campos que se muestran son solamente los del tipo ASP. Los elementos HTML no se muestran en este tipo de diagrama ya que no mantienen una conexión directa con el servidor.

La única clase que tiene un tipo anidado es PaseSalida debido a que esta es la que genera un reporte en formato PDF desde el servidor y para colocar leyendas como encabezados y pie de página utiliza una clase independiente anidada a ésta.

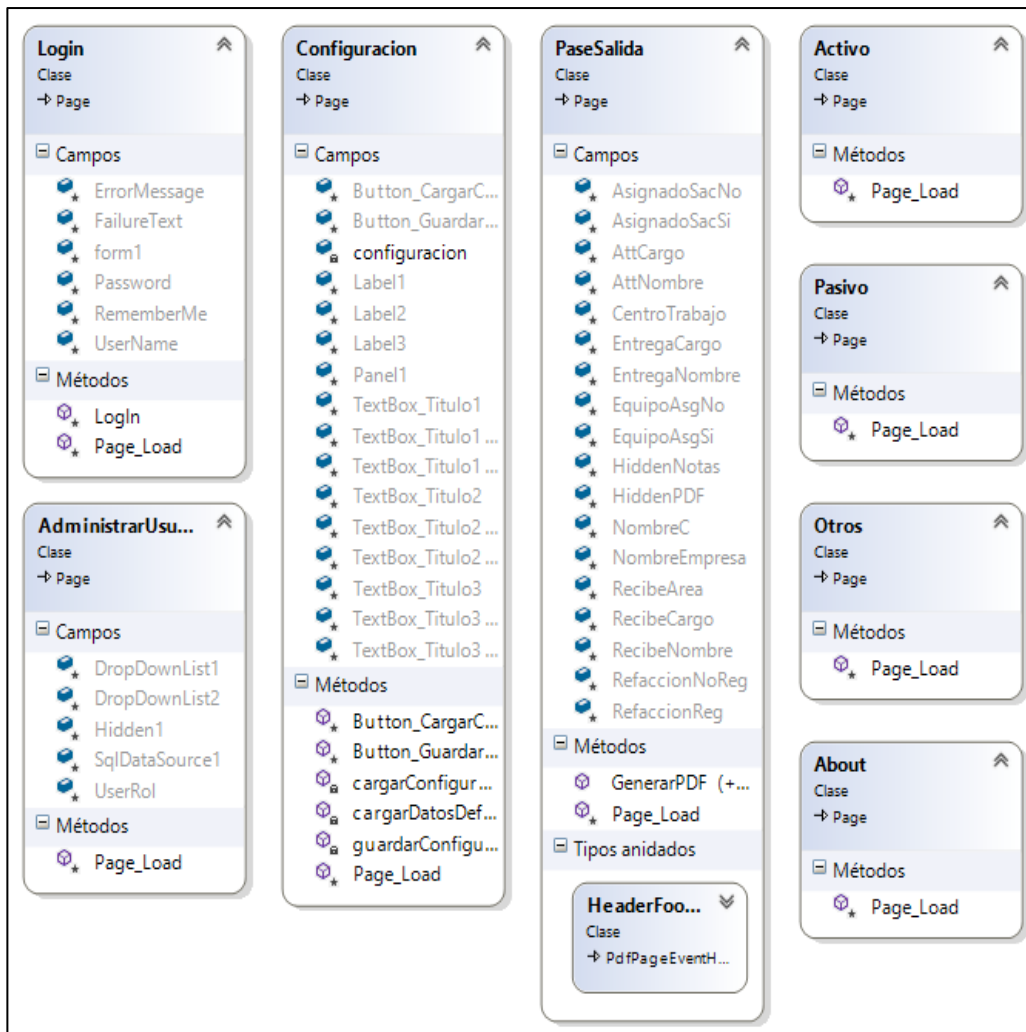


Figura 4.9. Diagrama de clases de contenido.

En la Figura 4.10 se encuentra el diagrama de las clases que integran los catálogos. Como se puede notar, no existen métodos dentro de éste debido a que toda la gestión de las peticiones es llevada por los controladores y los servicios Web.

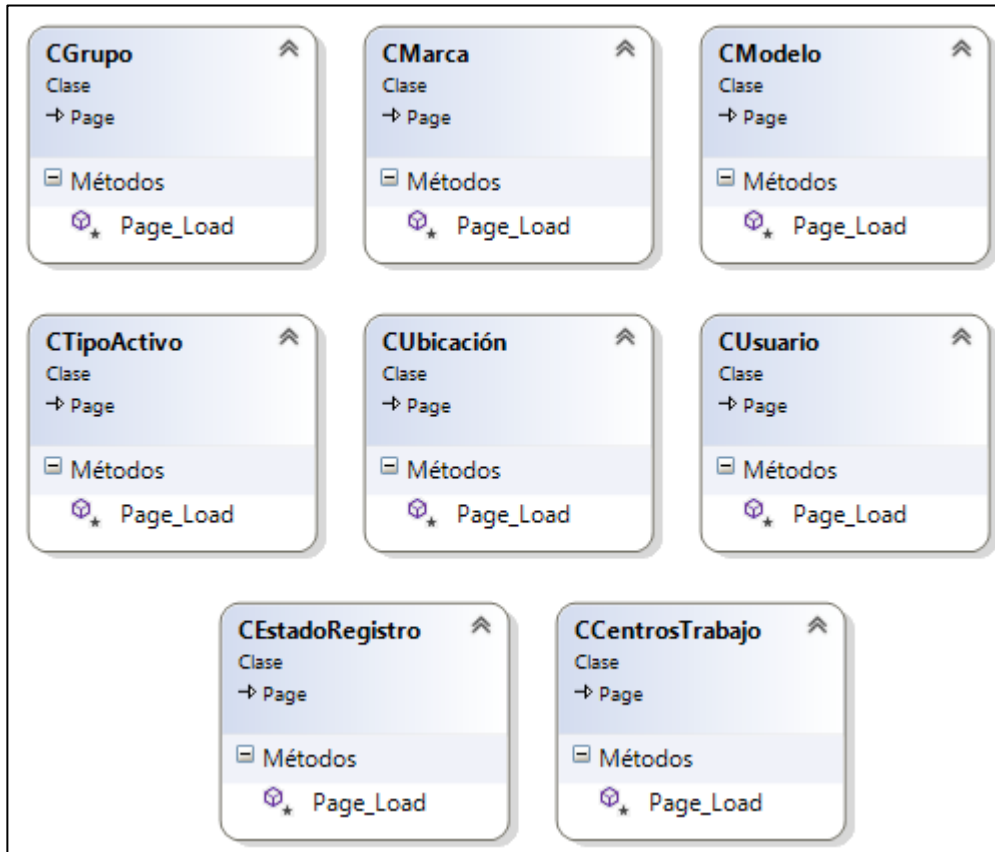


Figura 4.10 Diagrama de clases de catálogos.

En la Figura 4.11 se muestra el diagrama de clases de los controladores. A diferencia de los otros modelos, en este se encuentra en el encabezado el tipo de clase que es, es decir, un `HttpHandler`.



Figura 4.11 Diagrama de clases de los controladores.

En la Figura 4.12 se encuentra el diagrama de clases de los servicios Web. Estos han sido agrupados según su uso. Todas las operaciones de insertar se encuentran en una sola clase, así como en

modificar, eliminar y consultar. También se creó una clase que integra todos los servicios Web que utilizará la aplicación Móvil.

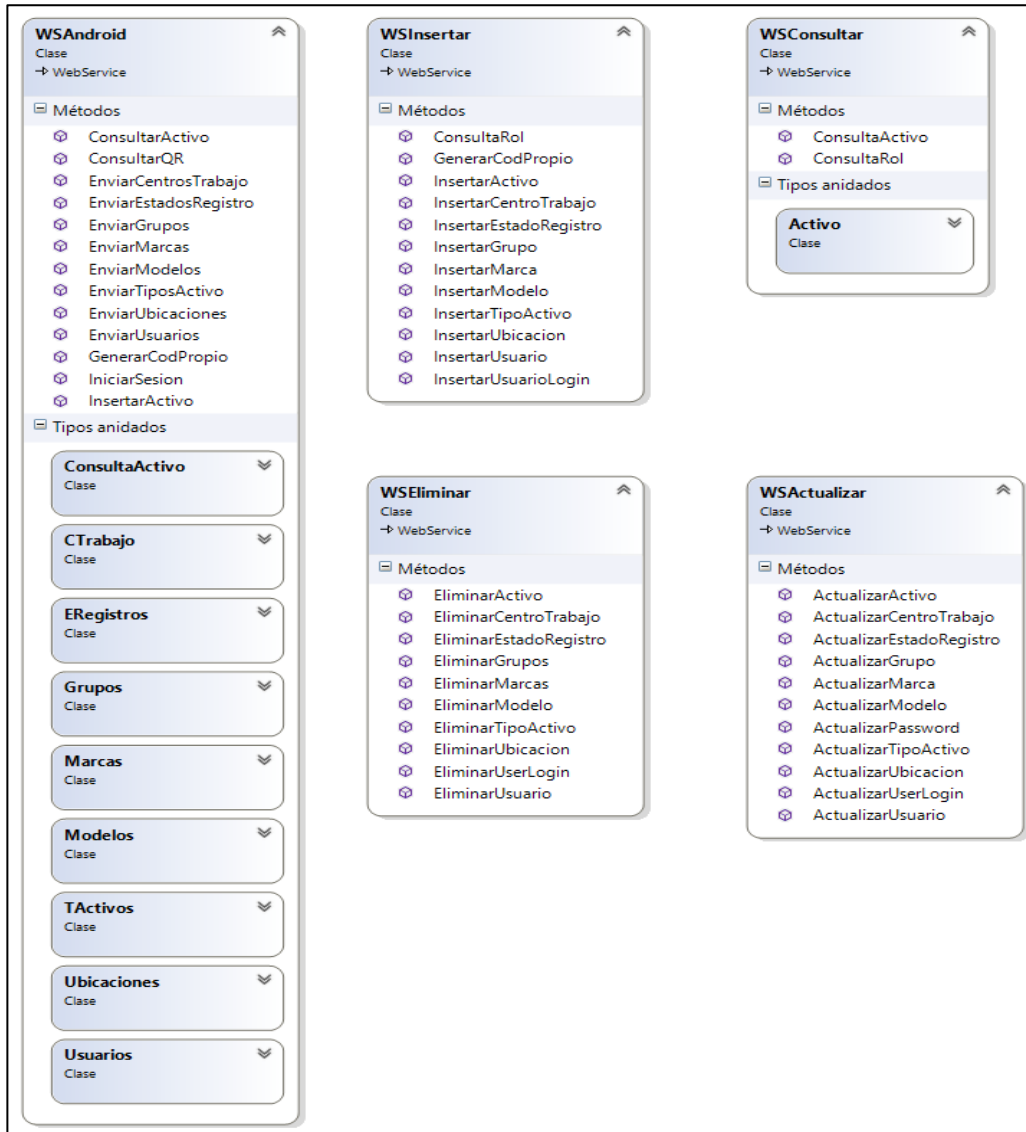


Figura 4.12 Diagrama de clases de los servicios Web.

Continuando con los diagramas, en la Figura 4.13 se encuentran los diagramas para las entidades creadas. Debido a la extensión de esta, se optó por mostrar el panorama general de las entidades. Esta información se verá más a detalle en el Modelo de Datos.



Figura 4.13 Diagrama de clases de las entidades.

En la Figura 4.14 se encuentra modeladas las clases que utiliza el método de autenticación utilizado en este proyecto, llamado Identity.

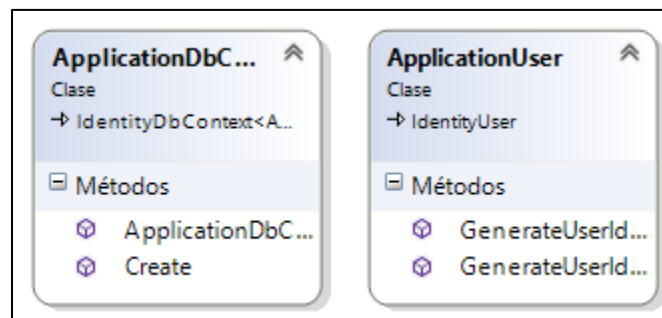


Figura 4.14 Diagrama de clases de los modelos.

A continuación en la Figura 4.15 se mostrará el diagrama de clases para poder guardar la configuración de las preferencias de usuario, es decir el diseño de los títulos de los encabezados.

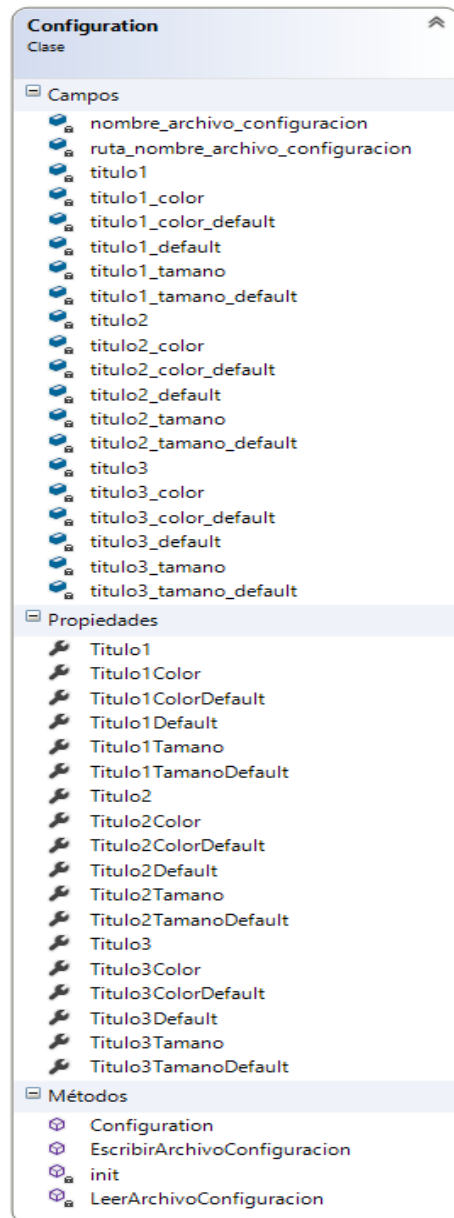


Figura 4.15 Diagrama de clases de AppData.

Por último, en la Figura 4.16 se mostrará el diagrama de clases para generar los códigos QR desde el servidor. Esta consta de una sola clase que hace todo el proceso.

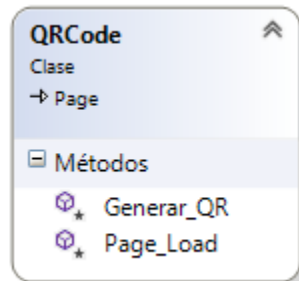


Figura 4.16 Diagrama de clase para generar códigos QR.

Ya que se han descrito todos los diagramas de clases que componen la aplicación Web, se analizará el diagrama de clases de la aplicación Móvil. Este, como se podrá notar, es más sencillo ya que la cantidad de operaciones que se realizan son menores.

En la Figura 4.17 se encuentra el diagrama de clases de la Aplicación Móvil. En esta solamente se muestra como interaccionan las clases, no se muestran los nombres de los métodos y campos, esto porque el diagrama sería muy extenso y no alcanzaría el espacio dentro de la tesis.

La mayoría de los elementos se comunican con la clase AppCompatActivity, que es la clase base para las actividades que utilizan las bibliotecas de soporte para características de la barra de acción (o eventos dentro de la aplicación); es decir, cualquier evento que se realice en la actividad ésta clase es la que va a redirigir a otro layout (o página de la aplicación).

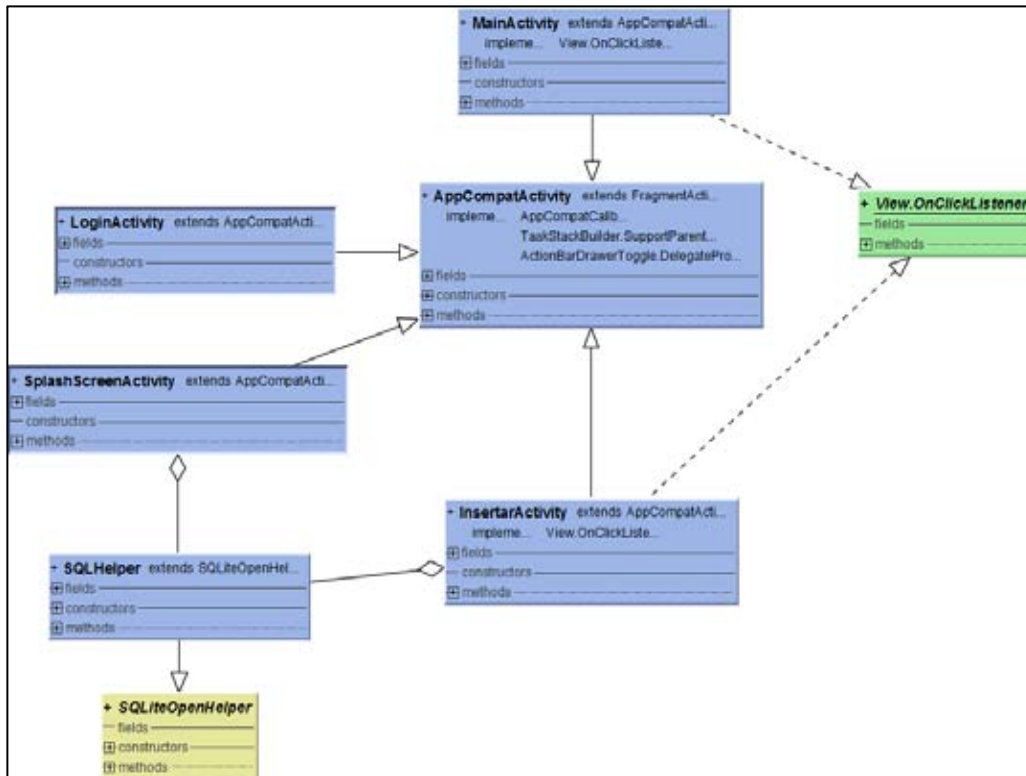


Figura 4.17 Diagrama de clases Aplicación Móvil.

En la Tabla 4.5 se describirá el uso de cada una de las clases dentro del diagrama de la Figura 4.17, esto para tener una visión más clara

Tabla 4.5 Descripción de las clases de la Aplicación Móvil.

Nombre	Descripción
SplashScreenActivity	Es la actividad que se muestra a los usuarios como pantalla de presentación a la aplicación cada vez que se ejecuta.

Nombre	Descripción
LoginActivity	Actividad que gestiona el inicio de sesión de los usuarios.
MainActivity	Actividad principal donde se muestran todas las operaciones que se pueden realizar.
InsertarActivity	Actividad que muestra el formulario para poder insertar datos al control de inventario.
AppCompatActivity	Clase base para las actividades que utilizan las bibliotecas de soporte para características de la barra de acción.
SQLHelper	Clase que contiene los métodos para realizar operaciones en la Base de Datos interna.
SQLiteOpenHelper	Método que se ejecuta cada vez que se hace una llamada al SQLHelper.
View.OnClickListener	Método que gestiona todos los eventos al dar click en los elementos de la pantalla.

Modelo de datos

El modelo de datos representa la estructura de las entidades de la Base de Datos. Para este proyecto se utilizaron dos bases de datos. La primera es de toda la información que respecta al control de inventario y

la segunda es de la autenticación o los usuarios, la cual se genera automáticamente según el tipo de autenticación elegida.

En la Figura 4.18 se muestra el modelo de datos de la información del proyecto. El diccionario de datos de éste se encuentra en el Anexo B. Diccionario de datos del Activo.

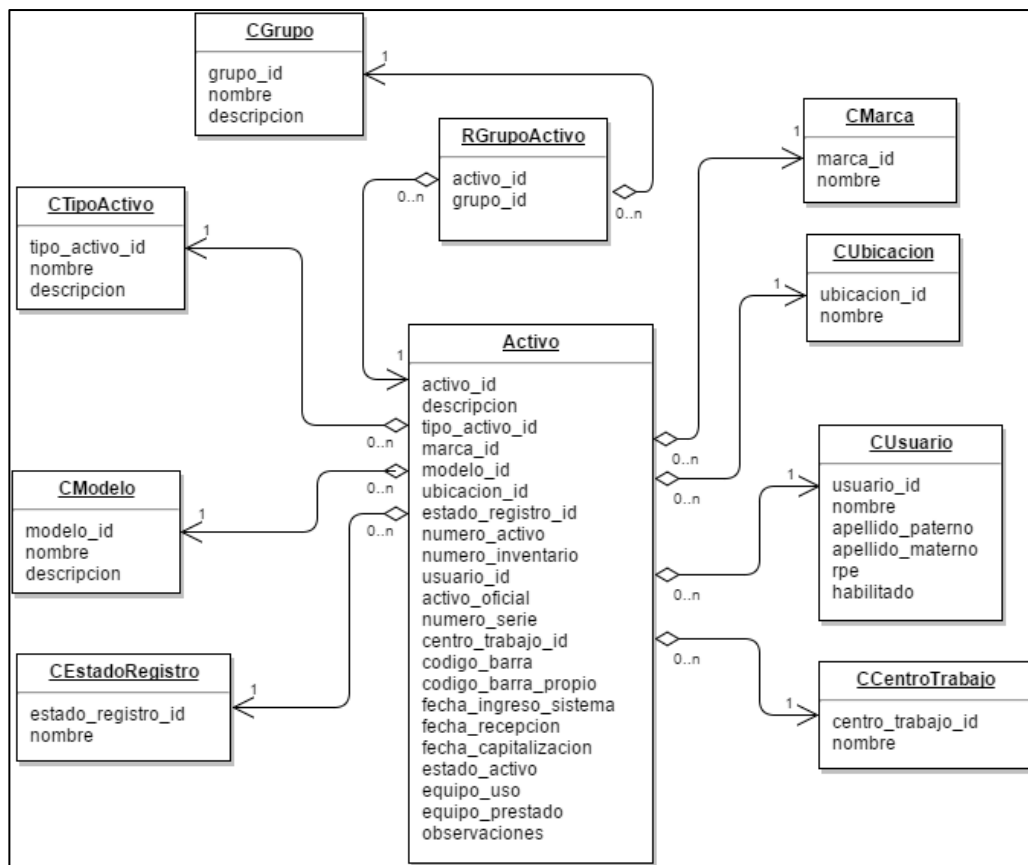


Figura 4.18 Modelo de datos del proyecto.

En la Figura 4.19 se muestra el modelo de datos de la autenticación (Identity). Esta consta de más tablas, pero las que son útiles para este proyecto son solamente tres: Usuarios, Roles y Usuarios-

Roles. El diccionario de datos de éste se encuentra en el Anexo C. Diccionario de datos de la Autenticación.

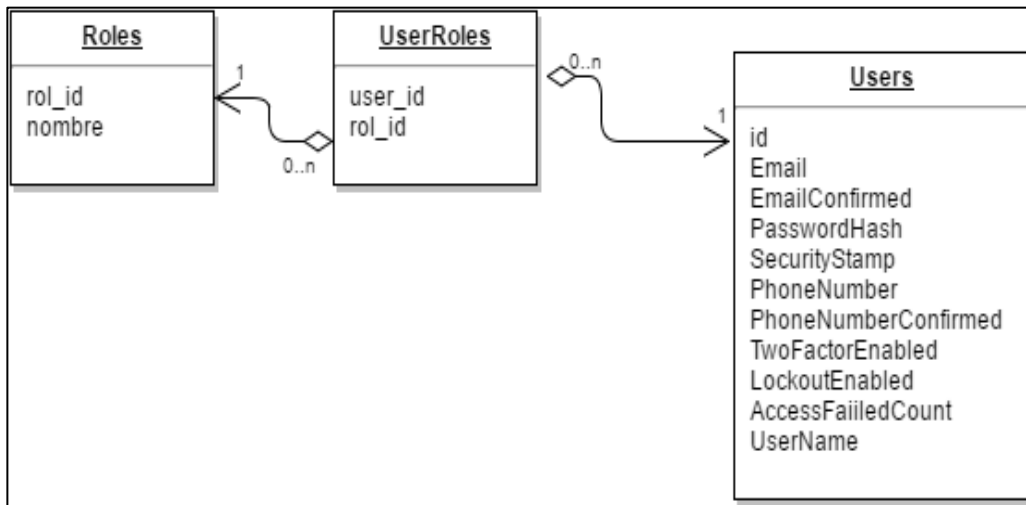


Figura 4.19 Diagrama de modelo de datos de la autenticación.

Diagrama Entidad – Relación

En la Figura 4.20 se encuentra el diagrama entidad – relación de la base de datos, para diseñar el esquema que posteriormente será implementado en el gestor de base de datos SQL Server. Esto ayudará a entender los datos y como se relacionan entre ellos.

En la Figura 4.21 se muestra el Diagrama Entidad – Relación de la tabla de la autenticación. Se omitieron algunas tablas que genera automáticamente el modelo que utiliza ASP.

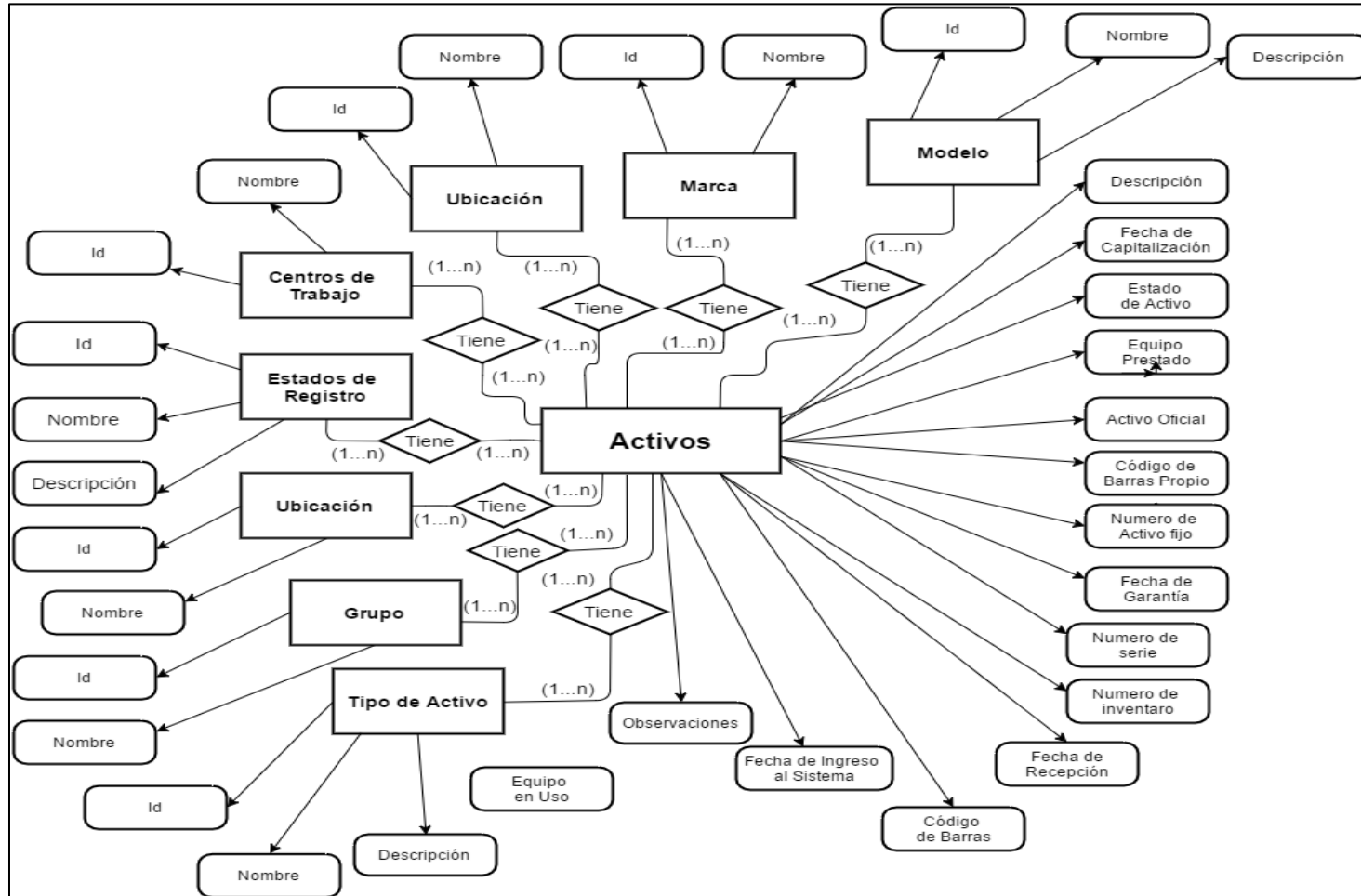


Figura 4.20 Diagrama Entidad – Relación de los Activos.

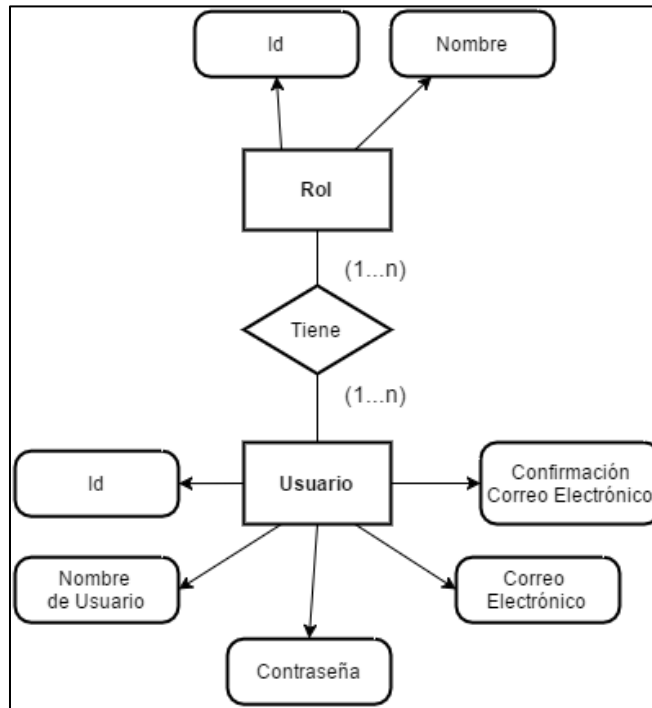


Figura 4.21 Diagrama Entidad – Relación de la autenticación.

Modelo de procesos

Se especifica el flujo de ejecución, es decir, la secuencia de operaciones que pueden ser ejecutadas. De los múltiples diagramas que existen, para describir este proyecto se utilizarán: Mapa de navegación, diagrama de prototipos y modelo UWE.

Mapa de Navegación

En la Figura 4.22 se muestra el mapa de navegación de la aplicación Web.

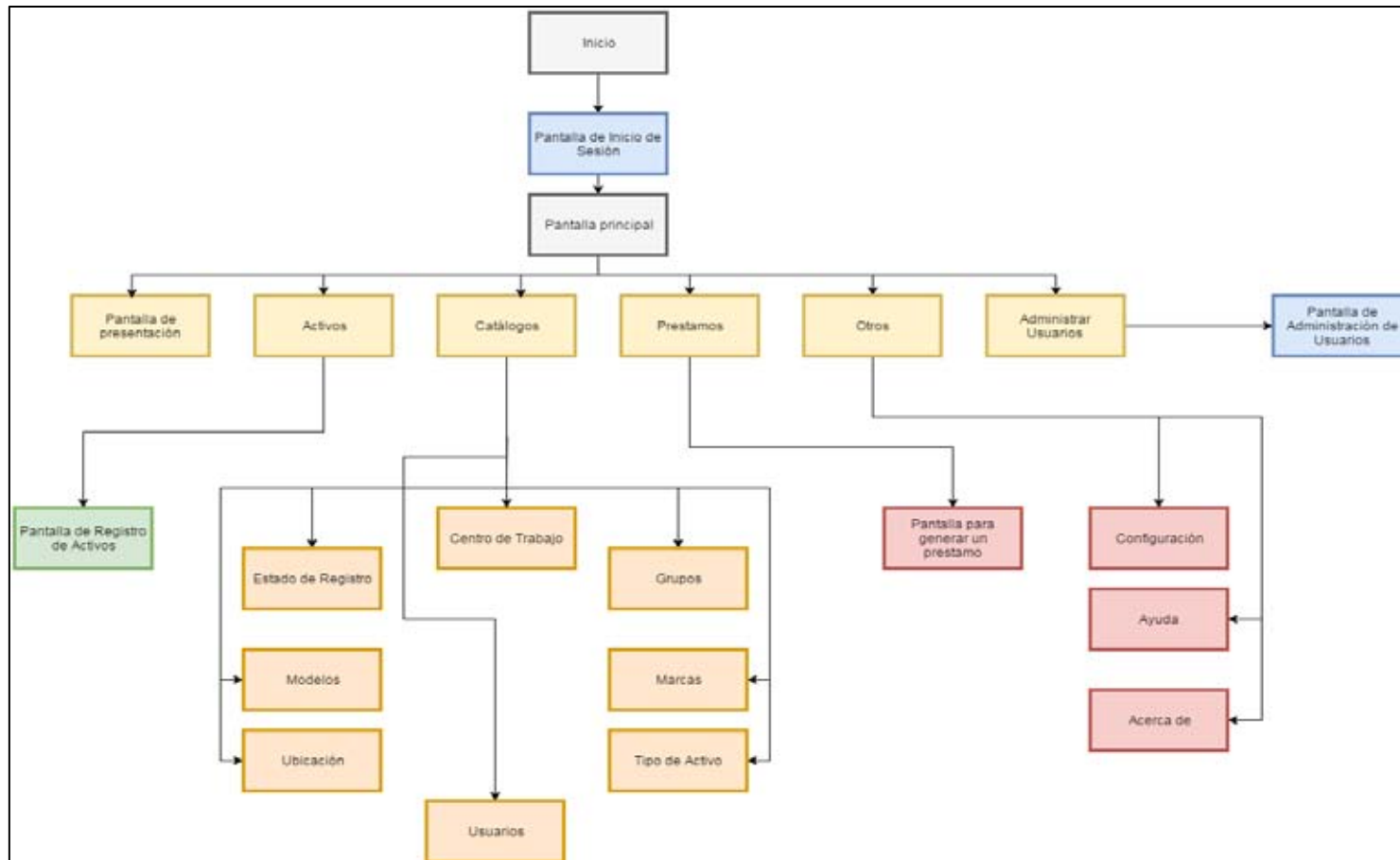


Figura 4.22 Mapa de Sitio de la Aplicación Web.

En la Figura 4.23 se encuentra el mapa de navegación de la aplicación Móvil. Al iniciar sesión muestra una pantalla de presentación que aparte de funcionar como efecto de transición también descarga los catálogos de la base de datos del servidor, almacenándolos en la base de datos interna. Si existe un error, el usuario puede dirigirse a alguna de las siguientes opciones:

- Conectar nuevamente: donde confirma que se encuentra habilitado el adaptador del Wifi del dispositivo móvil y recarga la conexión.
- Cambio de dirección IP: El usuario puede especificar la dirección del servidor al que se va a conectar.

Una vez que es proceso haya sido terminado, se dirigirá al Inicio de Sesión, donde proporcionará credenciales válidas para poder dirigirse a la actividad principal. En ésta existen las siguientes opciones: Cerrar sesión (que dirige a la actividad de Iniciar Sesión), Insertar Registro, Plantillas y Búsqueda de Registro.

La segunda y tercera opción dirigen a la actividad que contiene el formulario de registro y la última opción habilita la cámara del dispositivo móvil para ser utilizada como un escáner de códigos QR y una vez detectado el código, redirigirse al formulario de registro.

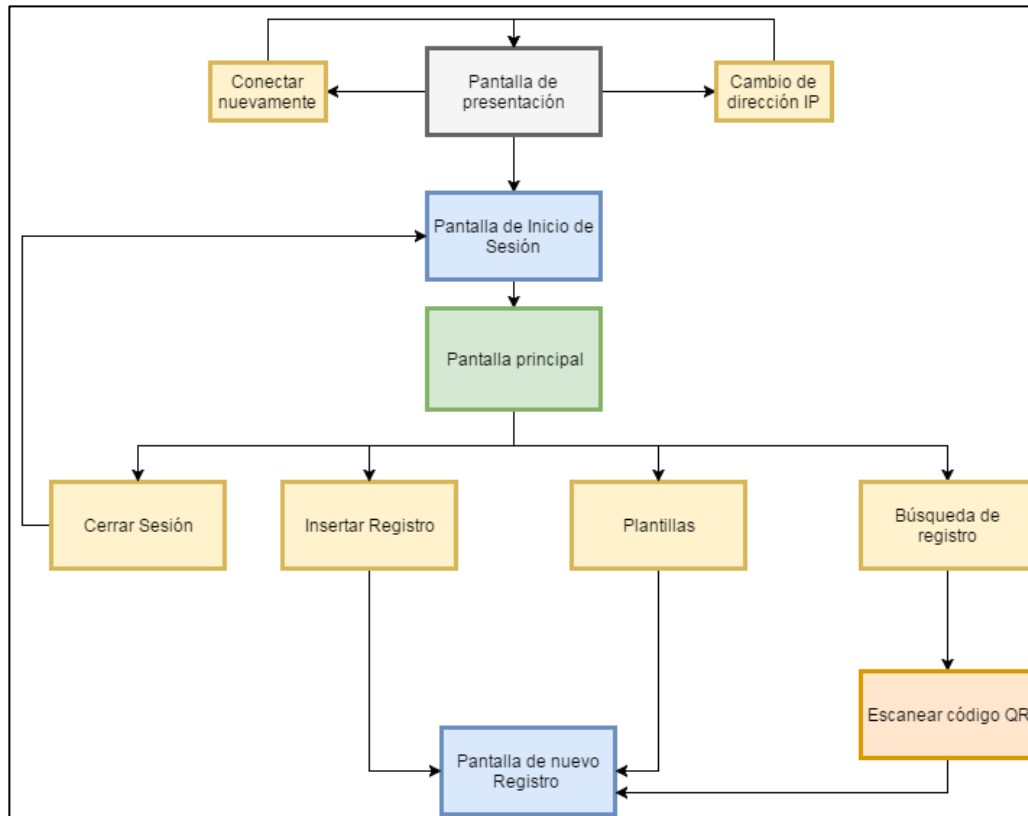


Figura 4.23 Mapa de navegación de la aplicación Móvil.

Diagramas de Prototipos

También conocidos como Wireframes, en estos se visualiza el diseño de las pantallas de las aplicaciones. Los diagramas se hicieron utilizando la versión gratuita del programa “Justinmind”, una aplicación de escritorio para el prototipado de sistemas.

En la Figura 4.24 se muestra el esquema de la pantalla principal orientado al caso en el que un usuario que se encuentra en anonimato en la aplicación Web.

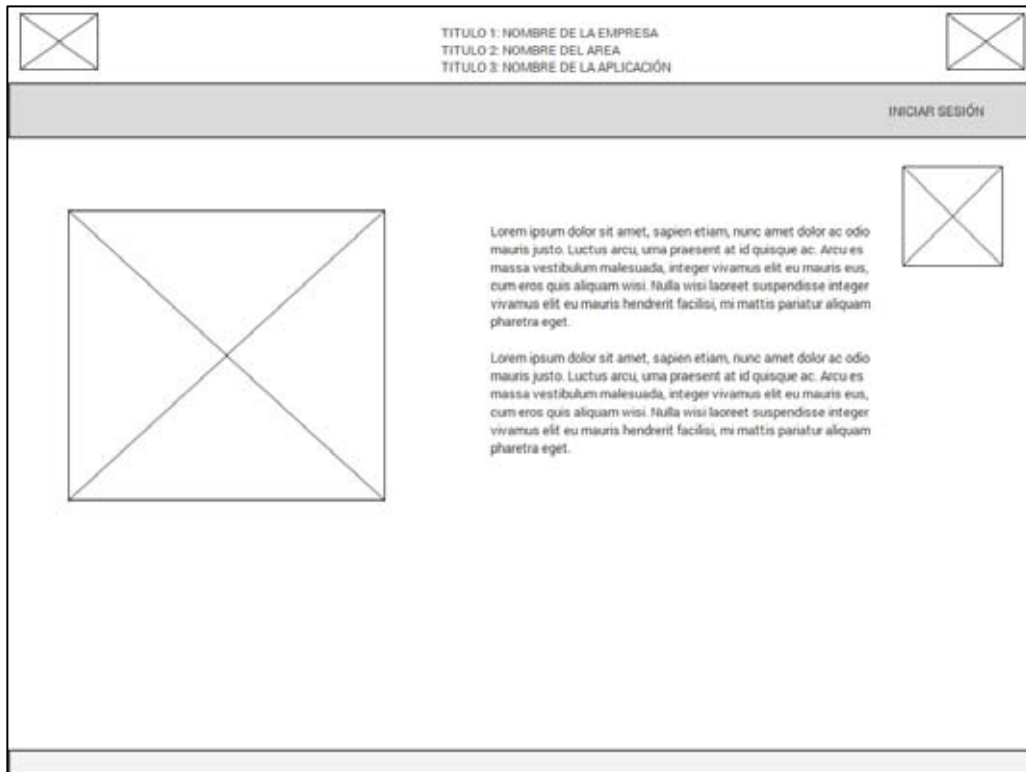


Figura 4.24 Wireframe de la página de Inicio.

La Figura 4.25 muestra el prototipo de la pantalla principal de la aplicación Web una vez que el usuario haya proporcionado las credenciales correctas.

Se destaca que a diferencia del prototipo anterior, este muestra las opciones en el menú y el acceso para los Activos, Pasivos, catálogos (donde se mostrará un menú desplegable) y la opción “Otros” en donde despliega información como Ayuda y Acerca de, en caso de tener la autoridad de administrador, se mostrará la opción para administrar las cuentas de los usuarios.

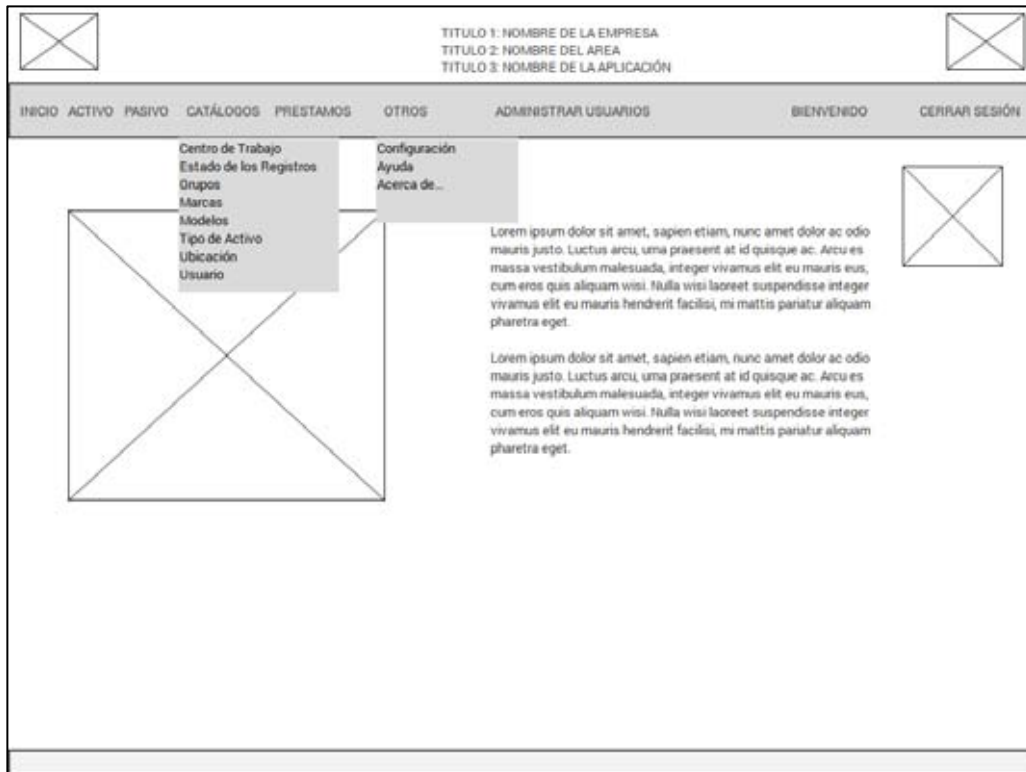


Figura 4.25 Wireframe de la pantalla de inicio para usuarios autenticados.

En la Figura 4.26 se encuentra el prototipo de la página para los catálogos, los cuales tendrán el mismo diseño que se mostró en la Figura 4.25, modificando únicamente los títulos de las tablas donde se despliega la información que existe en la base de datos. En la parte superior se alojarán los botones de acciones a realizar, tales como cuadro de búsqueda, exportar y filtrar.

A pesar de haber cambiado de página, el banner y el pie de página conservarán el mismo diseño. Esto es lo que se le conoce como página Máster.

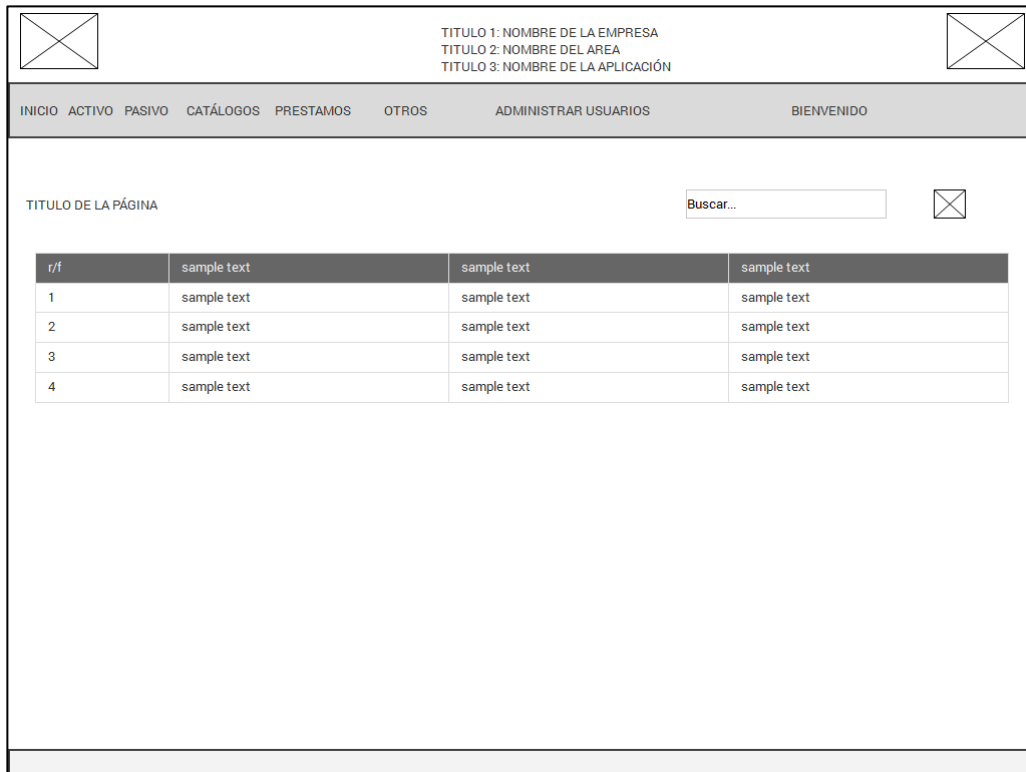


Figura 4.26 Wireframe para los catálogos.

El diseño de la página correspondiente al Activo (ver Figura 4.27) tendrá una columna especial para los botones que se integrarán en la parte superior derecha, cada uno realizará las acciones: exportar como PDF, exportar como XLS, generar plantillas para código QR, seleccionar todos los registros, quitar selección de todos los registros y mostrar columnas, el título estará en la parte superior izquierda al igual que en los catálogos.

En la última columna de la tabla se encuentran los botones para las operaciones que se pueden realizar sobre un registro, las cuales son: Editar, eliminar y duplicar.

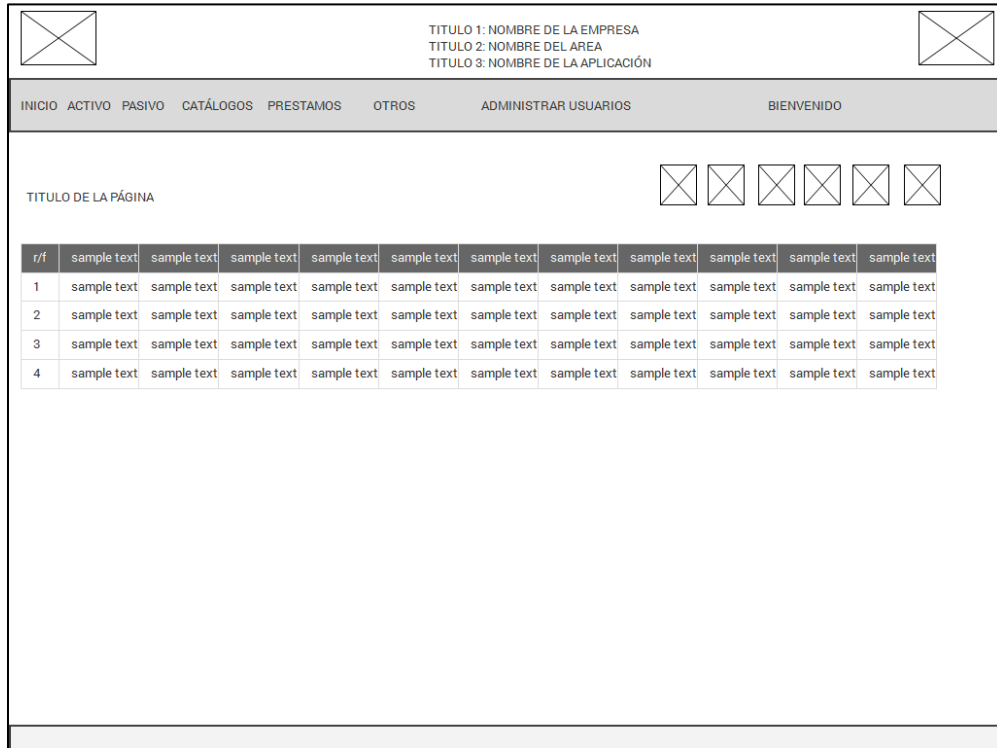


Figura 4.27 Wireframe para la página de Activos.

En la Figura 4.28 se encuentra el prototipo de la página para generar los reportes de Pase de Salida, los elementos estarán justificados simulando como se mostrará en el archivo al ser exportado, la tabla utilizará un diseño minimalista donde se encontrarán los elementos que saldrán del centro de trabajo; La imagen del lado derecho será el icono para eliminar celdas de la tabla y se tendrá un botón debajo de la tabla para agregar un nuevo renglón. Se utilizarán botones de tipo radio (radio buttons) para indicar las condiciones del activo al ser generado el reporte.

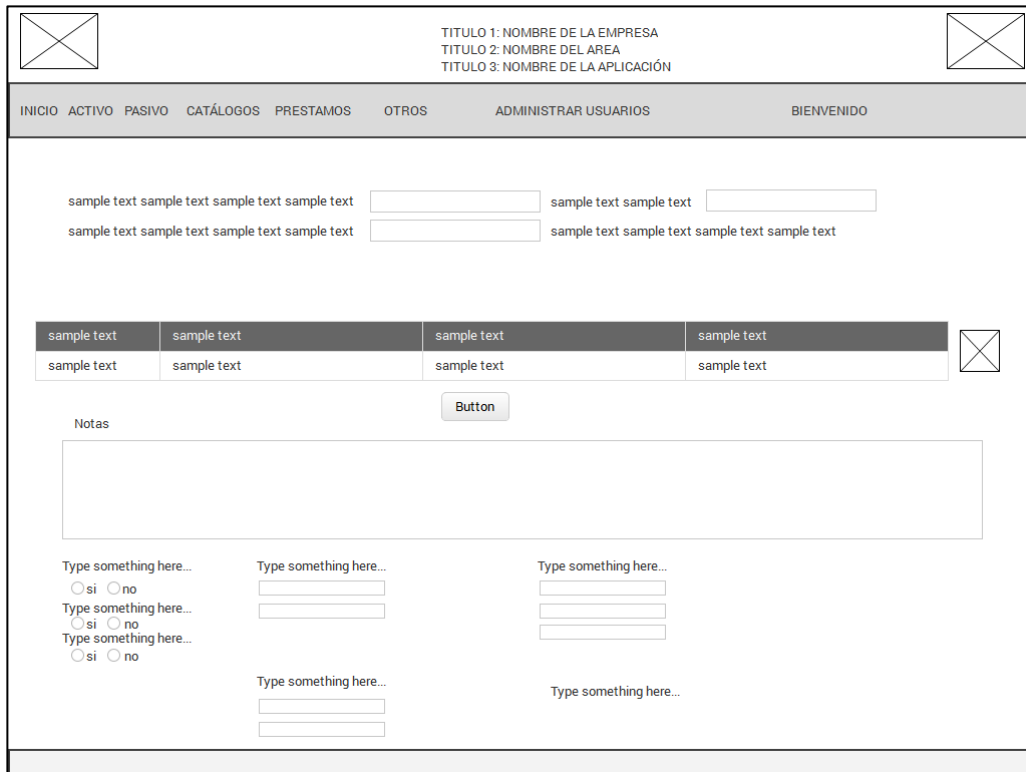


Figura 4.28 Wireframe para el Pase de Salida.

La página para la configuración, es decir, la personalización del encabezado de la página, se maneja con una tabla que contendrá los componentes que puede editar el administrador; El diseño se basa en poder cambiar el nombre de los títulos los cuales, así como el color de estos (que se podrán obtener mediante una paleta de colores que desplegará el código hexadecimal del color seleccionado) y el cambio de logotipos de la empresa.

En la Figura 4.29 se encuentra el diseño del Wireframe de la página de la configuración.

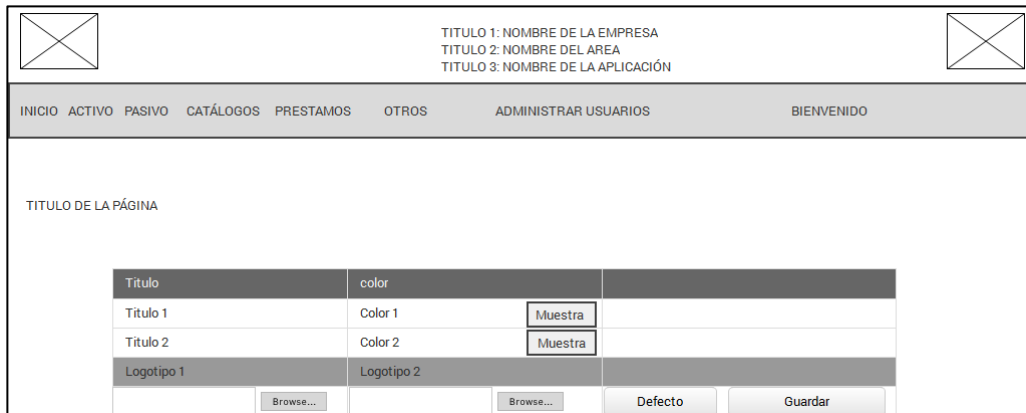


Figura 4.29 Wireframe para la página de la configuración.

Con la Figura 4.29 se concluye el diseño de los Wireframes de la aplicación Web; En la siguiente sección se describirán los diseños para la aplicación Móvil.

El prototipo de la aplicación móvil consta de un diseño simplificado, como se mencionó al principio, basado en los patrones de diseño de Material Design, esto para que el usuario tenga una mejor experiencia al utilizarla.

Al inicio de la aplicación se muestra la pantalla conocida como Splash Screen (ver Figura 4.30), la cual es la portada de presentación de la aplicación. Se tendrán dos opciones representados por dos botones: Uno será utilizado para recargar la conexión con el servidor y el segundo para desplegar una ventana de alerta para cambiar la configuración de la dirección IP del servidor de Base de Datos. Como fondo de esta pantalla se utilizará el logotipo característico de todo el Sistema de Control de Inventarios.

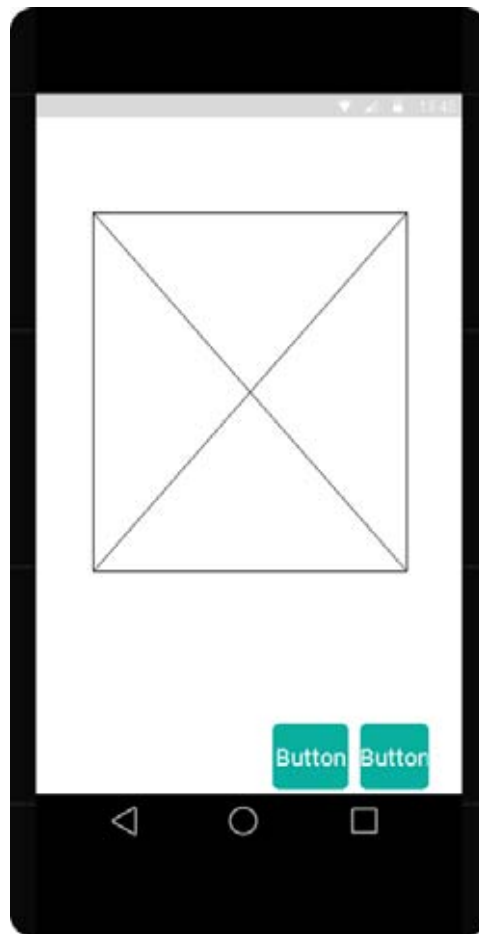


Figura 4.30 Wireframe del Splash Screen.

En la Figura 4.31 se muestra el diseño de la ventana de inicio de sesión que se ejecuta después de la Splash Screen, en donde se tienen dos campos de entrada (usuario y contraseña) y el botón correspondiente para iniciar sesión.

También se agregará una casilla de verificación para recordar las credenciales de acceso de los usuarios.

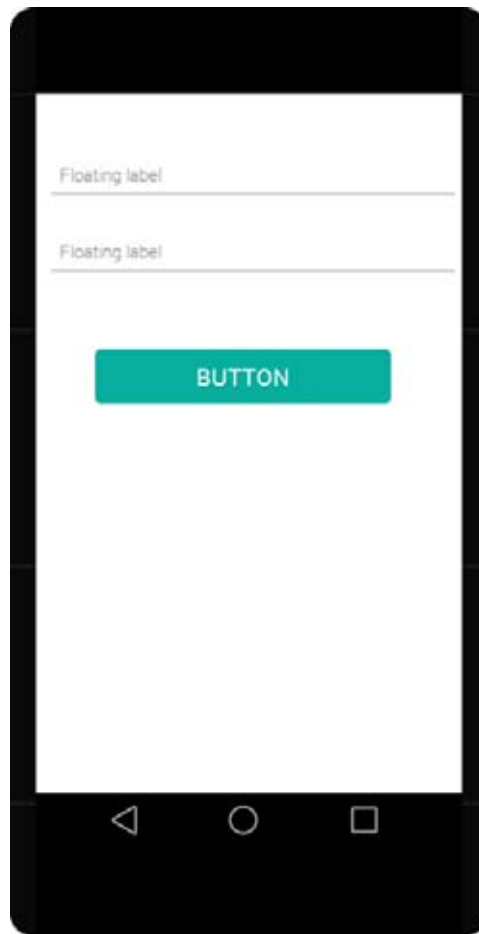


Figura 4.31 Wireframe de iniciar sesión.

En la Figura 4.32 se encuentra el esquema de la pantalla que surge después de la validación de las credenciales de usuario correctamente. Esta es conocida como la pantalla principal, en donde se muestran los títulos, logotipos de la empresa y una barra de acciones en la parte inferior que contiene botones para redirigir a las diferentes operaciones a realizar: Insertar, búsqueda, plantillas, ayuda y cerrar sesión.

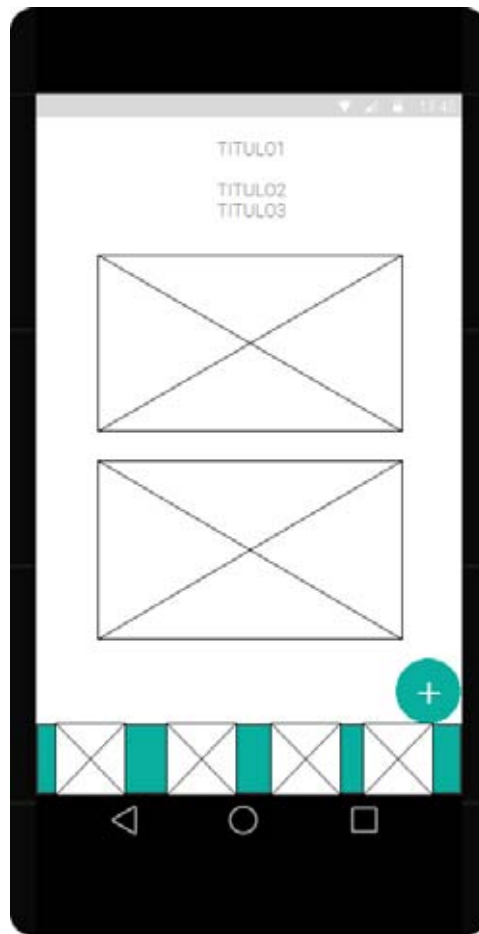


Figura 4.32 Página principal de la aplicación Móvil.

El prototipo de la ventana para la inserción surge al haber seleccionado el primer elemento de la barra de acciones ubicada en la pantalla principal.

En la Figura 4.33 se muestra el diseño del formulario para la inserción de datos, consta de cajas de texto, listas desplegadas y botones de tipo radio para que el proceso sea lo más sencillo posible para el usuario. Este mismo formulario será reutilizado para actualizar los registros



Figura 4.33 Formulario para insertar registros.

Los prototipos de las pantallas presentadas han sido utilizados como guía del diseño de páginas Web y aplicaciones móviles. Estos se implementan en la aplicación final utilizando lenguajes especializados en esquematizar interfaces.

En el caso de las páginas Web se utiliza HTML, CSS y Javascript; Para los Móviles, el lenguaje XAML, plugins y Assets.

Modelo de comunicación

Utilizando los diagramas de comunicación se explica el proceso que sigue determinada acción al ser ejecutada, los elementos que intervienen en esta y el flujo a seguir.

La manera en que se realiza la comunicación de cualquier proceso hacia la base de datos en la aplicación Web es a través del modelo de paso de mensajes de Ajax. Este tipo de modelo de comunicación se le conoce como asíncrono, ya que espera a que el proceso reciba la información para continuar su ejecución.

Como se puede notar en la Figura 4.33 que esquematiza la forma de trabajo de Ajax, esta tecnología se divide en 3 bloques: El proceso del lado del cliente, el medio de transporte y el proceso del lado del servidor. Si se compara con el esquema tradicional de MPI (Interfaz de paso de mensajes) se encuentran cada uno de los elementos principales que lo conforman, en donde se tiene un proceso en el emisor y en el receptor y el modelo de paso de mensajes.

En primer lugar, se realiza una llamada desde el lado del cliente utilizando el lenguaje Javascript. En este proceso se prepara toda la información para que el motor de Javascript envíe la solicitud al servidor. Utilizando un tipo de transporte Web, en este caso HTTP (HyperText Transfer Protocol), se comunica la petición al servidor. En este punto comienzan los procesos del lado del servidor donde un Servicio Web, sistema de base de datos o cualquier otro tipo de procedimiento procesará la solicitud y enviará la respuesta pertinente en formato XML. Dentro del

XML la información puede ir estructurada como un diccionario de datos o como una cadena JSON.

El mismo motor que envía la solicitud es el que espera la respuesta de ésta, enviándola al proceso del lado del cliente utilizando los lenguajes HTML y CSS.

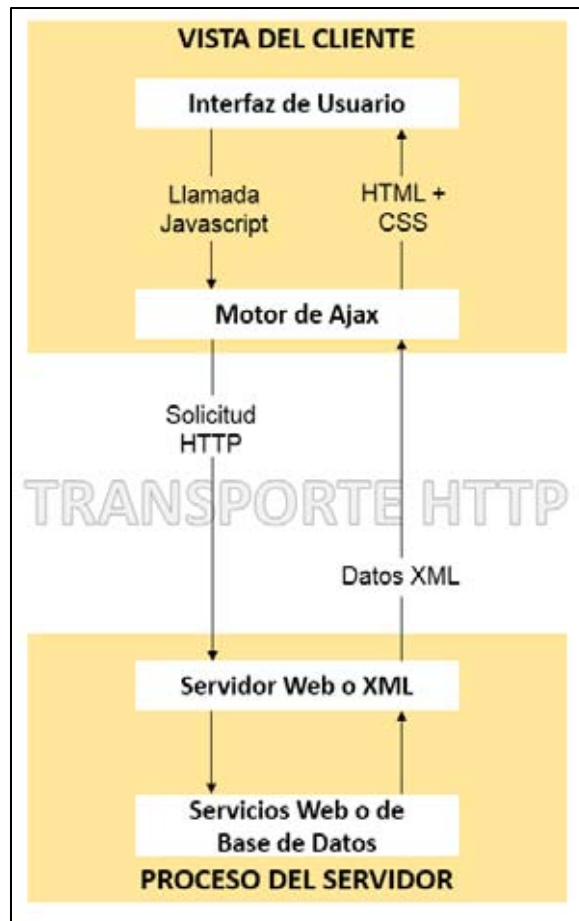


Figura 4.34 Esquema de trabajo de Ajax.

En la Figura 4.35 se esquematiza la forma de trabajo de Ksoap, que es el protocolo utilizado para la comunicación entre la aplicación

Móvil y el servidor Web. Para este caso, las llamadas de datos se realizan a través de clases en Java, enviadas a través del transporte Ksoap utilizando el protocolo HTTP. En el servidor, se procesa la solicitud correspondiente dando como respuesta la información en un archivo XML. El mismo transporte que envía la solicitud espera la respuesta del servicio Web y el resultado lo imprime en pantalla como una cadena de resultados.

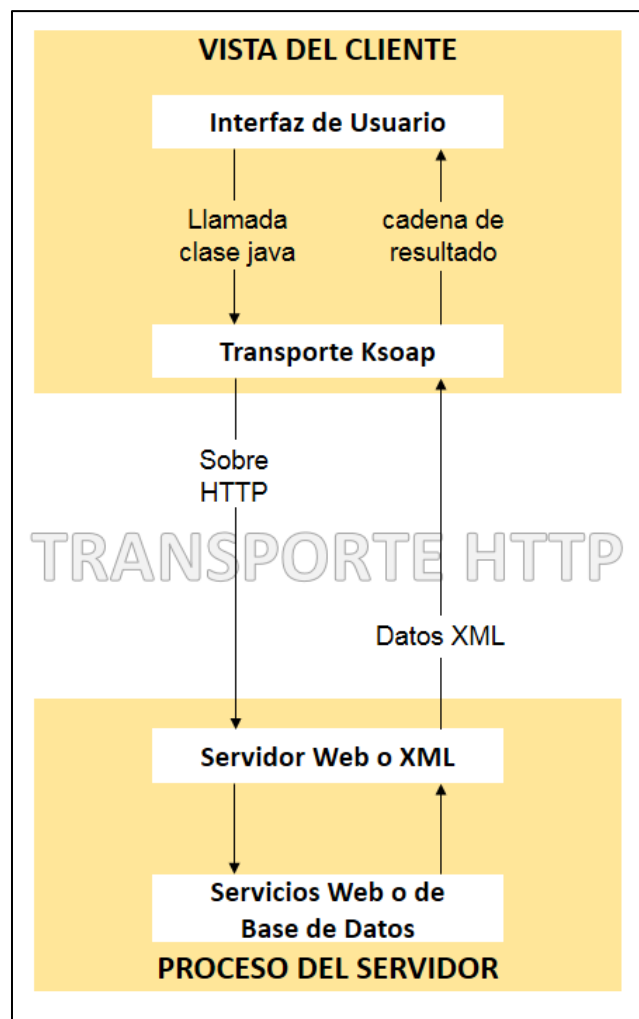


Figura 4.35 Esquema de trabajo de Ksoap.

4.4 Documentación

Durante el desarrollo del software se contempla como una etapa más la documentación, siendo esta de gran importancia para el producto final. Resulta crucial guardar las especificaciones establecidas que fundamentan el funcionamiento del software, así como de los componentes a partir de los que se forma la aplicación final.

Esta etapa se ve reflejada en el sistema en el módulo de Ayuda, donde el usuario puede encontrar una serie de preguntas que usualmente podrían surgir durante la ejecución del sistema.

En esta sección se presentarán los diagramas de caso uso, para representar de manera gráfica el comportamiento final de los diferentes tipos de usuarios operando el sistema.

Para una mejor organización de la información se decide agrupar las acciones en común entre tipos de usuario, de esta forma se forman 2 grupos: Administrador y Superusuario y en otro, Capturista e Invitado.

En la Figura 4.36 se muestran las acciones a realizar por los usuarios que tienen mayores permisos en la aplicación, es decir, el Administrador y el Superusuario.

Comparando las actividades que lleva a cabo cada perfil, se puede notar que en su mayoría coinciden. La diferencia está en que el Administrador es el único actor que puede administrar las credenciales de los usuarios (alta, baja, modificación, asignar rol, reestablecer contraseña) y cambiar los títulos y las imágenes del encabezado de la aplicación.

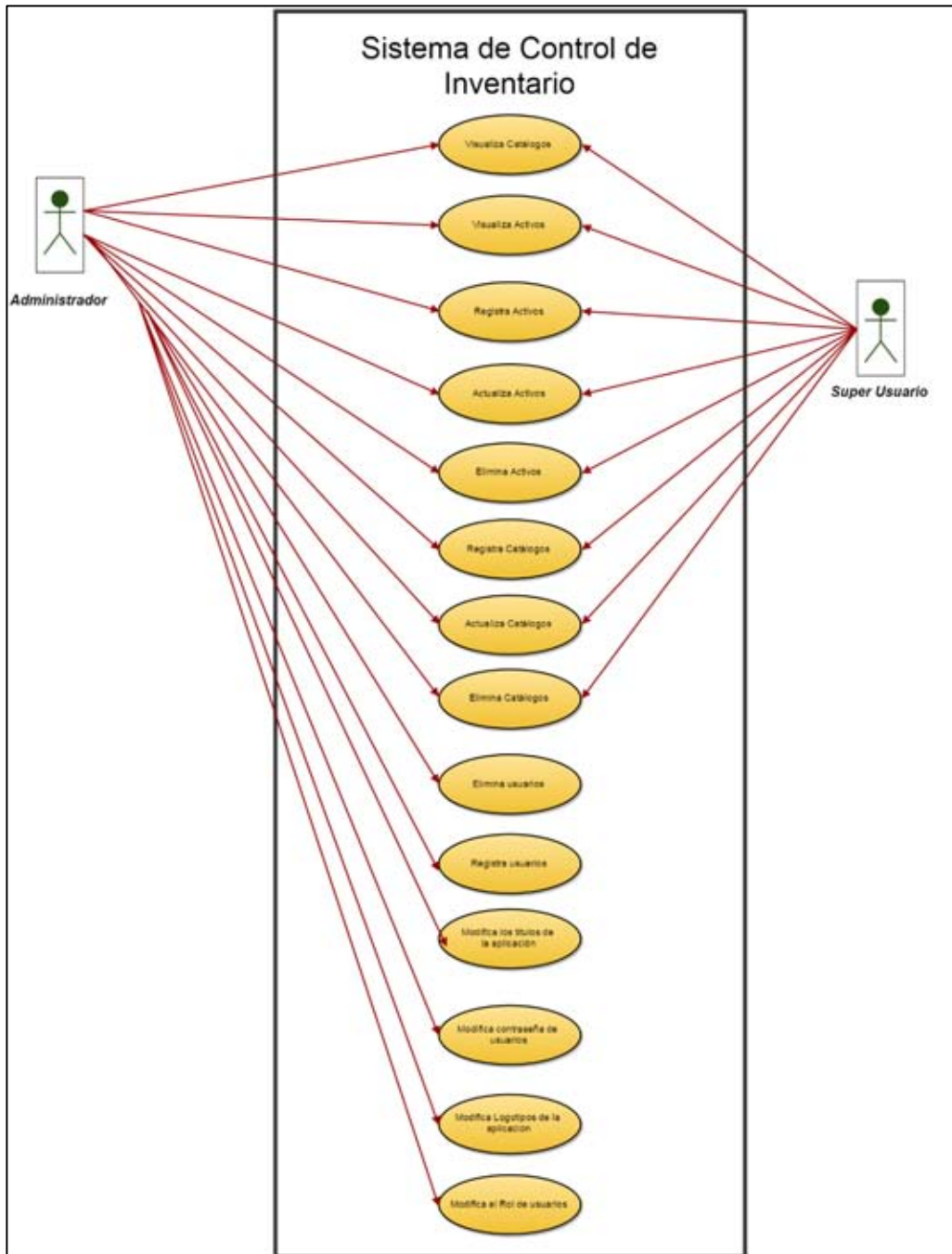


Figura 4.36 Diagrama de caso de uso: Administrador y Superusuario.

En la Figura 4.37 se esquematizan los permisos de los roles: Capturista e Invitado. Las operaciones en común entre ambos son poder visualizar la información de los Activos y los Catálogos. La diferencia es que el Capturista sí puede agregar información a estos módulos, mientras que el Invitado se limita a simplemente ver información.

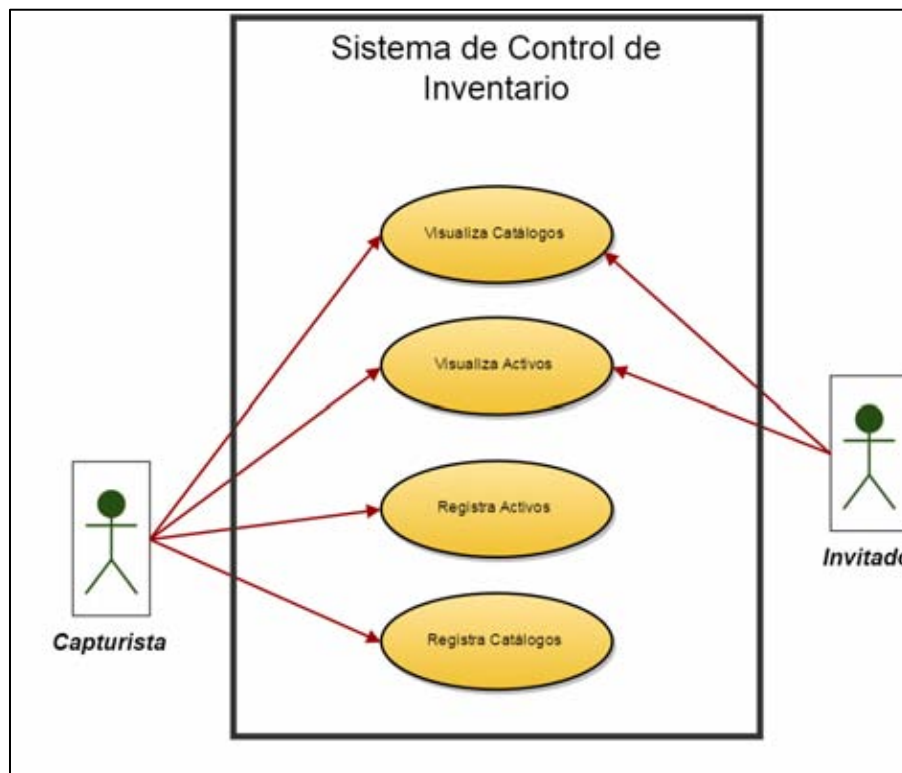


Figura 4.37 Diagrama de casos de uso: Capturista e Invitado.

Una vez comprobados que los permisos de los usuarios están apegados a los requerimientos iniciales del cliente, se continúa hacia la descripción del flujo de procedimientos para realizar las operaciones en cada una de las aplicaciones con las que consta el Sistema de Control de Inventarios.

4.5 Implementación

Esta etapa consiste en la aplicación de los requerimientos iniciales dentro del sistema. Estos se caracterizan por contener algoritmos o diagramas que ejemplifiquen las operaciones a realizar en el sistema.

Se decide utilizar como herramienta de esquematización el diagrama de Actividades para describir las operaciones que realiza todo el sistema.

Como primera instancia se describirán los procesos de la aplicación Web y en segundo punto los procesos realizados por la aplicación Móvil.

Diagrama de procesos para la aplicación Web

En la Figura 4.38 se muestra el primer procedimiento a realizar que es el Alta de Activos. Para poder realizar esta operación es necesario contar con los permisos pertinentes; si es así, se puede realizar el alta del activo. De no existir errores en este procedimiento, se mostrará un mensaje de alta exitosa.

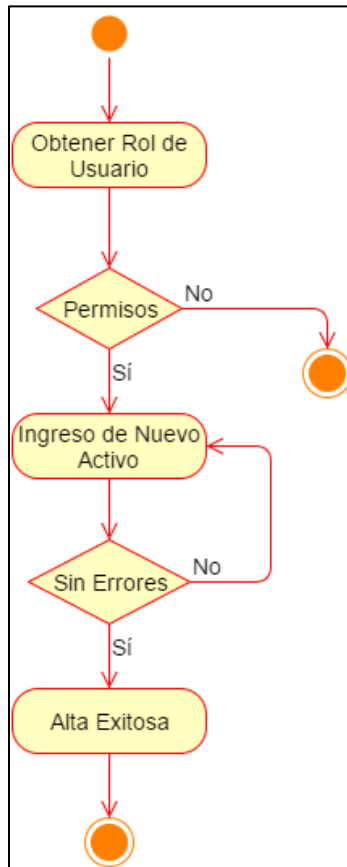


Figura 4.38 Procedimiento para alta de Activo.

En la Figura 4.39 se esquematiza el procedimiento para la baja de Activos. Para poder realizar esta operación es necesario contar con los permisos pertinentes; si es así, se puede realizar la modificación del registro del activo. De no existir errores en este procedimiento, se mostrará un mensaje de modificación exitosa.

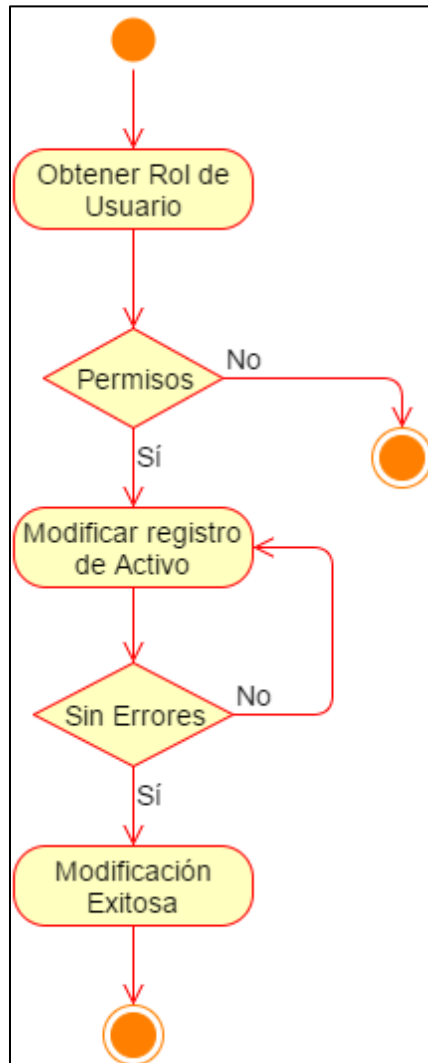


Figura 4.39 Procedimiento para la modificación de Activo.

En la Figura 4.20 se encuentra el procedimiento a seguir para la baja de un registro en Activos. Para realizar este procedimiento, es necesario contar con los permisos pertinentes. Si no existen errores, el proceso finaliza exitosamente.

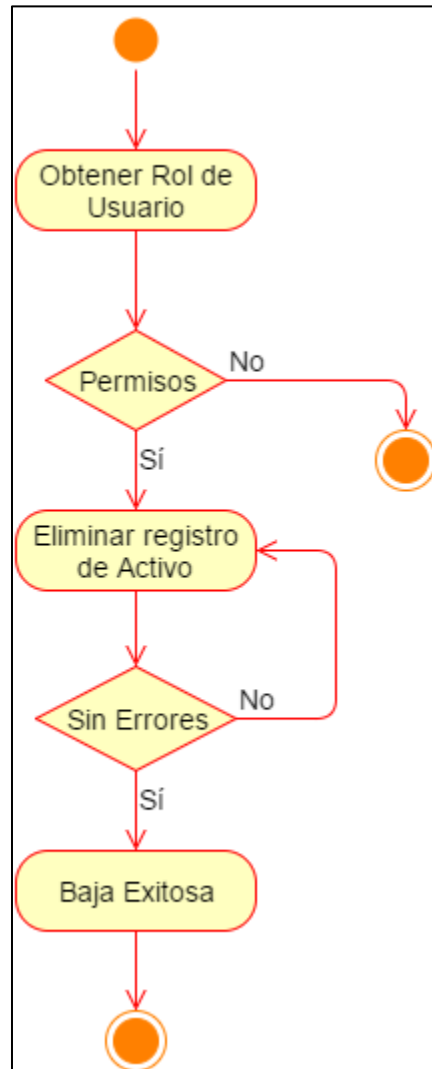


Figura 4.40 Procedimiento para baja de Activo.

En la Figura 4.41 se muestra el flujo a seguir para la búsqueda por filtros. Este consiste en seleccionar los criterios de búsqueda, enviar la información y esperar respuesta con los resultados obtenidos. De no encontrar coincidencias, se termina el proceso.

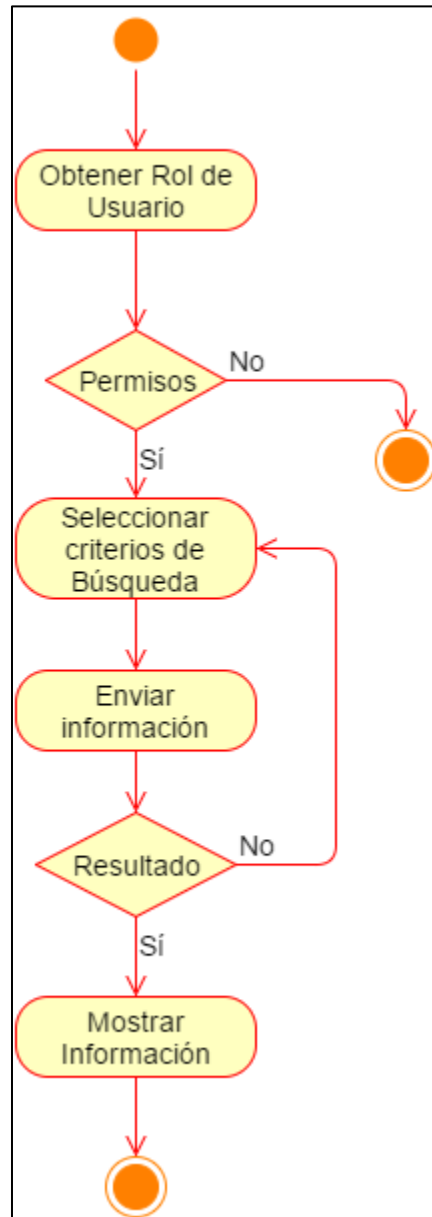


Figura 4.41 Procedimiento para la búsqueda por filtros.

En la Figura 4.42 se muestra el flujo de proceso para realizar un alta de usuario. Este procedimiento sólo puede ser realizado por el

administrador. Al dar de alta un usuario, también se le está asignando un rol. Si no existen errores, la operación es concluida exitosamente.

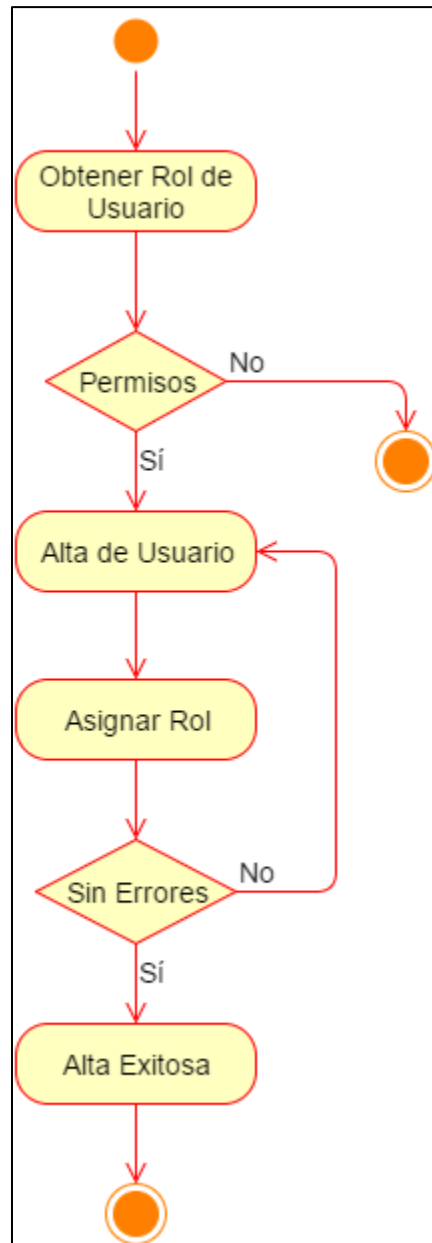


Figura 4.42 Procedimiento para dar de alta un usuario.

En la Figura 4.43 se esquematiza el procedimiento para modificar la información del usuario. En este proceso solamente se puede modificar el nombre de usuario o el correo electrónico. Para el rol y la contraseña se necesita un flujo diferente, explicados en las Figuras 4.44 y 4.45 respectivamente.

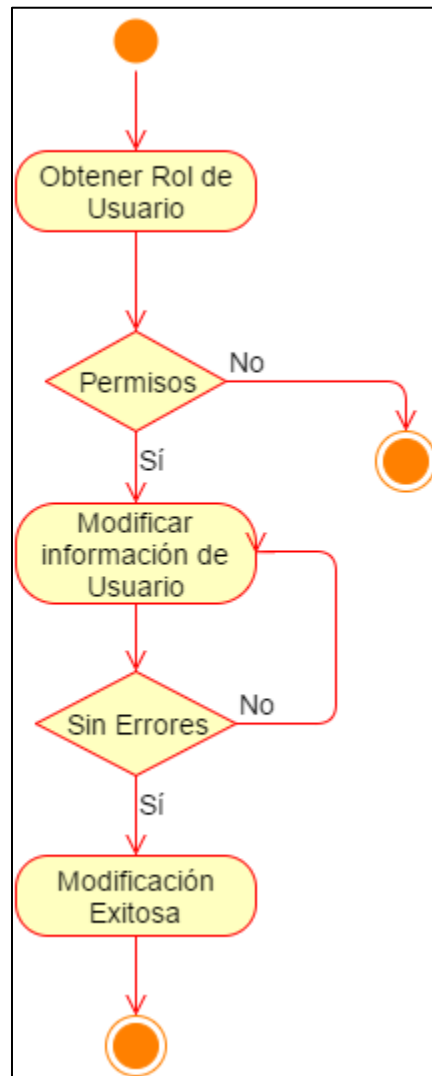


Figura 4.43 Procedimiento para modificación de usuario.

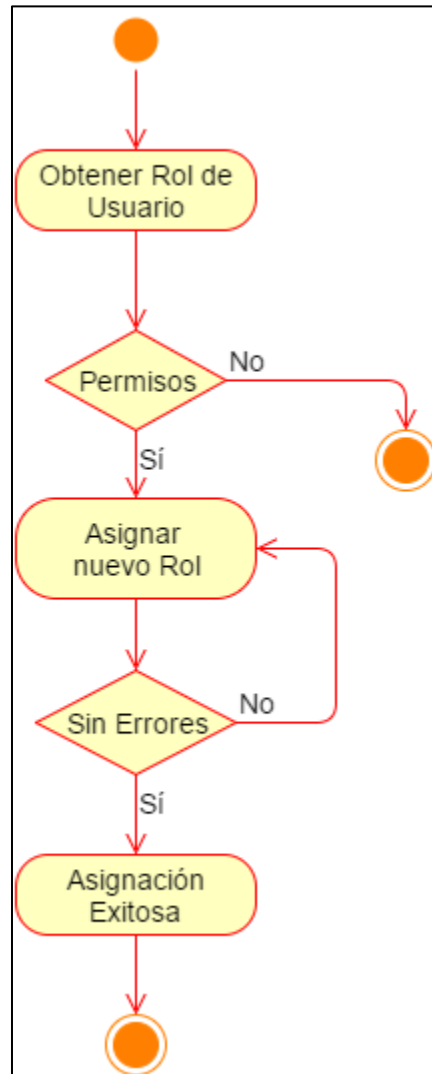


Figura 4.44 Procedimiento para modificar el rol del usuario.

Cabe aclarar que la contraseña del usuario no se puede recuperar, el único proceso a realizar con la contraseña es el restablecimiento, es decir la asignación de una nueva.

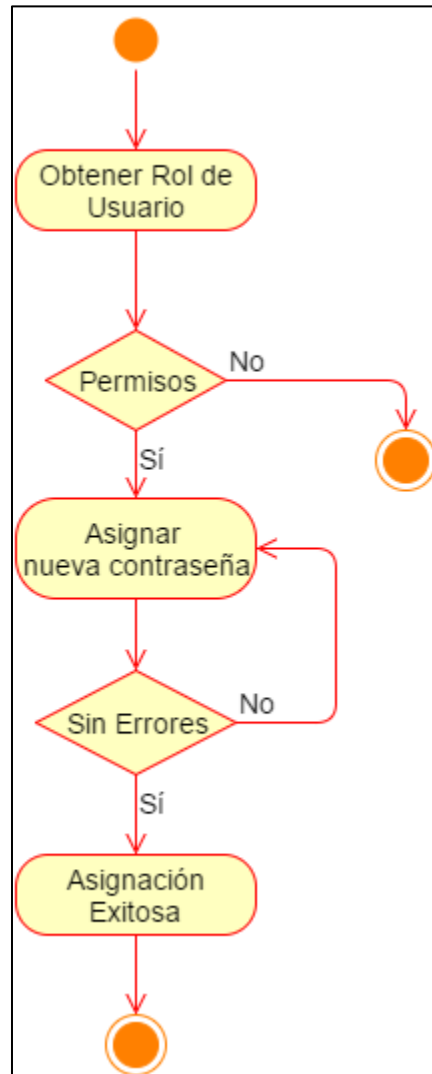


Figura 4.45 Procedimiento para reestablecer contraseña.

En la Figura 4.46 se encuentra el procedimiento para generar el reporte de todos los activos que se encuentran en la base de datos. Este reporte puede ser generado en 2 formatos: Hoja de Cálculo (Excel) o Documento en PDF.

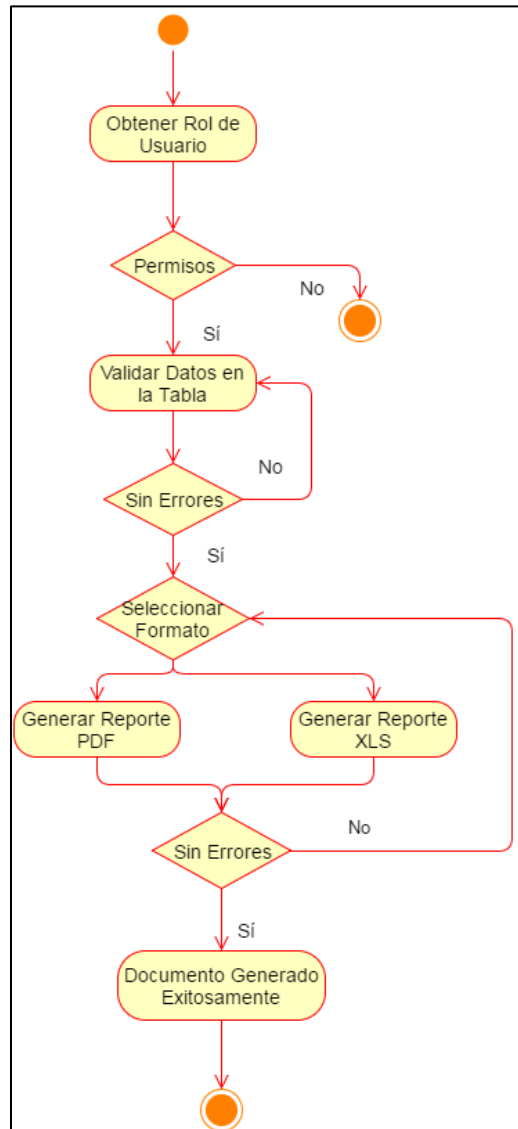


Figura 4.46 Procedimiento para generar reporte de los Activos.

En la Figura 4.47 se esquematiza el procedimiento a seguir para poder generar la plantilla con los códigos QR. Este reporte es importante ya que es el medio por el cual la aplicación Móvil puede comunicarse con la base de datos. La idea de este proceso es que el usuario seleccione

la plantilla de etiquetas con la que cuenta y el sistema el generará la plantilla en formato PDF con los márgenes correspondientes.

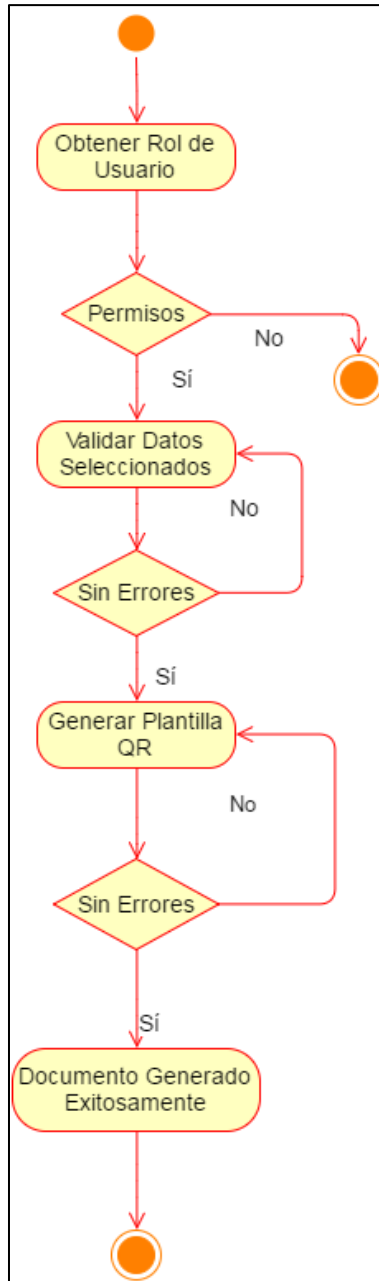


Figura 4.47 Procedimiento para generar plantilla de QR.

En la Figura 4.48 se encuentra el procedimiento a seguir para poder generar un pase de salida. Este pase se utiliza cuando alguno de los Activos saldrá el centro de trabajo ya sea para mantenimiento o en calidad de préstamo a otro centro de trabajo.

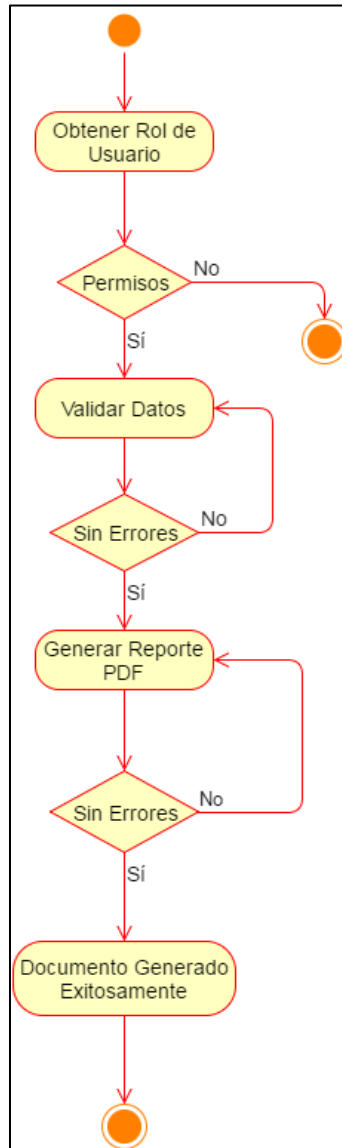


Figura 4.48 Generación de reporte de pase de salida.

Diagrama de procesos de la aplicación Móvil

En la Figura 4.49 se encuentra el procedimiento para iniciar sesión en la aplicación (tanto en Móvil como en Web). En esta se validará que las credenciales sean correctas, devolviendo el rol del usuario y así conocer la vista a la cual se redirigirá.

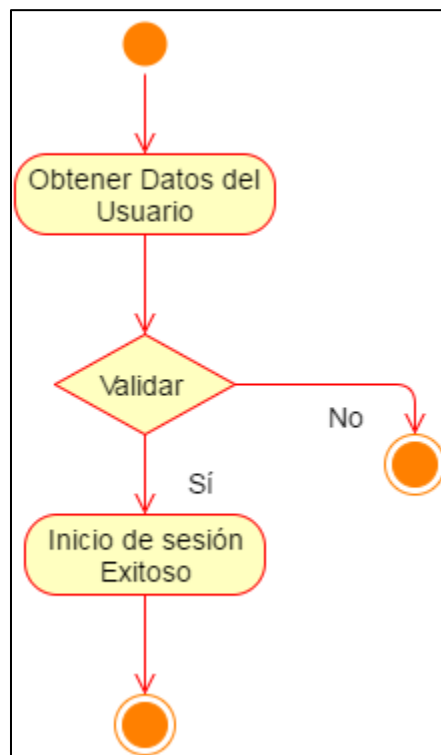


Figura 4.49 Procedimiento para iniciar sesión.

En la Figura 4.50 se encuentra el procedimiento para buscar la información de un Activo a través de la aplicación Móvil. La captura de datos se realizará a través de la cámara del dispositivo móvil, desplegándolo en un formulario.

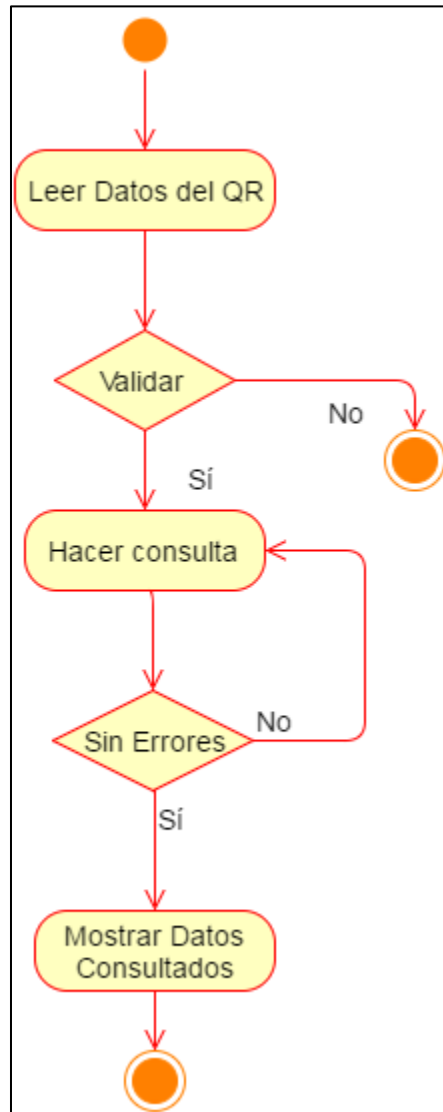


Figura 4.50 Procedimiento para la búsqueda a través del código QR.

Se han especificados todos los procedimientos a realizar a través del sistema con cada una de las aplicaciones o subsistemas que existen. Una vez finalizado esto, se puede dar paso a mostrar los resultados finales de la aplicación.

Este proyecto será implementado en el servidor propio del centro de trabajo, cuyas características de software (las cuales serán tomadas como referencia para mínimos requisitos) son:

Sistema Operativo: Windows Server 2008 R2.

Manejador de Base de Datos: Microsoft SQL Server 2008 R2 Express.

Servidor Web: Internet Information Services (IIS) 7.0.

Versión de .NET Framework: 4.0.

CAPITULO 5:

Resultados y trabajo futuro

CAPÍTULO V. RESULTADOS Y TRABAJO A FUTURO

En este capítulo se mostrarán las pantallas finales de ambos sistemas, el comportamiento final de las aplicaciones trabajando en conjunto, así como las actividades que serán catalogadas como trabajo futuro.

Se ha dividido en las siguientes secciones:

- **Integración de las aplicaciones en el Sistema Distribuido:** En esta sección se presenta el funcionamiento en conjunto de las aplicaciones. Se describe el diseño de los formularios, pantallas de inicio y reportes de los catálogos y tablas principales en la aplicación Web, así como la interfaz de la aplicación Móvil. En este se podrá ver como ambos sistemas se comunican hacia una misma base de datos. Se presentará el ejemplo de cada una de las operaciones básicas sobre el sistema de control de inventario (altas, bajas, duplicados, consultas, modificación y reportes).
- **Trabajo futuro:** En esta sección se presenta las actividades a realizar a fin de mantener la herramienta actualizada y que funcione de manera óptima, es decir, sea fácil y rápida. También se explicará la línea de investigación a seguir dentro de las áreas de la computación.

5.1 Integración de las aplicaciones en el Sistema Distribuido

Ahora, se presenta el producto terminado. Como primer punto se presentará la aplicación Web y después la aplicación Móvil.

Aplicación Web

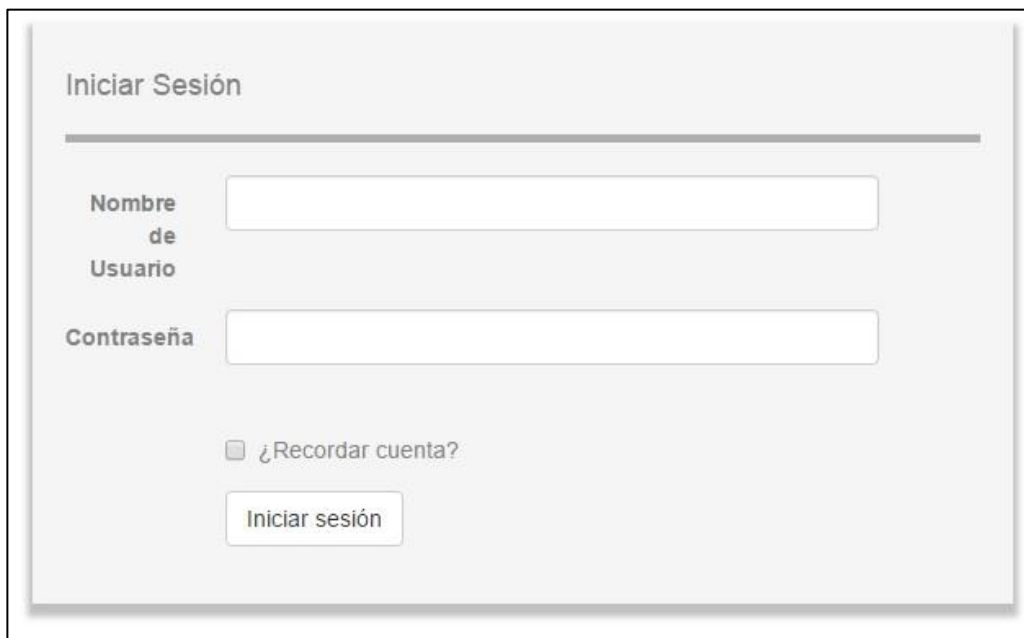
En la Figura 5.1 se encuentra la pantalla de inicio de la aplicación cuando el usuario se encuentra en estado anónimo. Como se puede notar, en el encabezado se encuentran los títulos y logos oficiales de la empresa, seguido de la barra de menú o de navegación en donde sólo se ofrece la opción para iniciar sesión.

Como contenido de la página está el logo oficial de la aplicación, así como una breve descripción del objetivo de la misma.



Figura 5.1 Pantalla de inicio de la aplicación Web.

Cuando el usuario selecciona la opción para iniciar sesión, se mostrará la pantalla de logueo de la Figura 5.2. Tanto el fondo como los encabezados desaparecen, a fin de que solamente se muestren únicamente las opciones para otorgar credenciales.



La imagen muestra una interfaz de usuario para iniciar sesión. El título "Iniciar Sesión" está en la parte superior izquierda. Debajo del título hay una línea horizontal. Hay dos campos de entrada de texto: el primero está etiquetado "Nombre de Usuario" y el segundo "Contraseña". Debajo de los campos hay un checkbox con el texto "¿Recordar cuenta?". En la parte inferior hay un botón que dice "Iniciar sesión".

Figura 5.2 Pantalla de logueo de la aplicación Web.

Al iniciar sesión, el usuario será redirigido de nuevo a la pantalla inicial, la cual presenta cambios en la información a mostrar (Figura 5.3). En la barra de navegación se puede notar que las opciones cambian, según el nivel de permisos que tenga el usuario: Administrador (Figura 5.4), Superusuario (Figura 5.5), Capturista (Figura 5.6) e Invitado (Figura 5.7).

También se agrega el botón para permitir la descarga de la aplicación Móvil a través del navegador Web.



Figura 5.3 Pantalla de inicio con usuario autenticado.

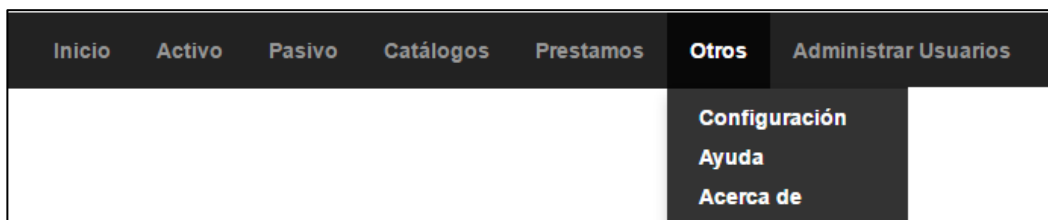


Figura 5.4 Opciones para el tipo de usuario administrador.

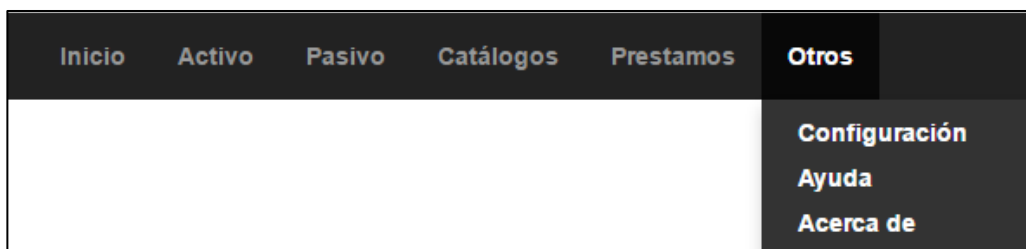


Figura 5.5 Opciones para el tipo de usuario Superusuario.

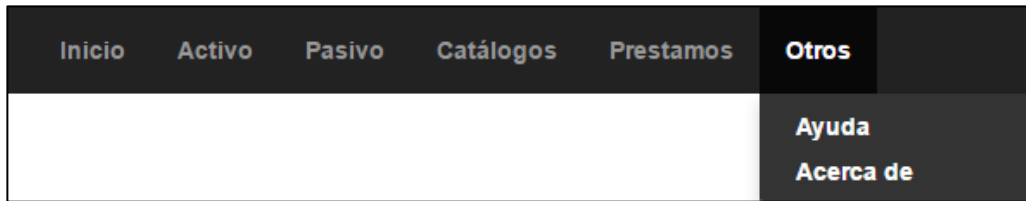


Figura 5.6 Opciones para el tipo de usuario capturista.

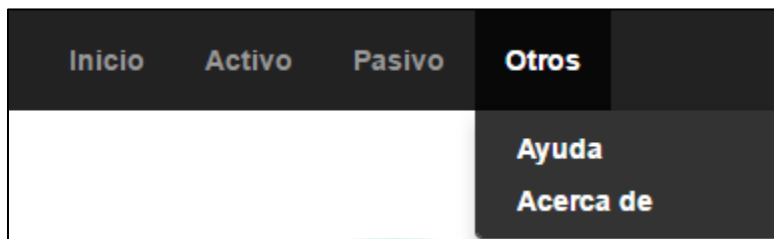


Figura 5.7 Opciones para el tipo de usuario invitado.

Navegando a través de las diferentes opciones que existen en el menú, en primera posición se encuentra el módulo de Activos (Figura 5.8). Como contenido se muestra una tabla con los últimos 20 registros que se encuentren almacenados en la base de datos.

En la cabecera de la tabla está una barra de opciones con las operaciones que se pueden realizar sobre los registros listados a continuación (de izquierda a derecha):

- **Insertar:** Permite agregar un nuevo registro a la base de datos. Esta ventana aparece en forma de dialogo (Figura 5.9).
- **Búsqueda:** A través de filtros que el usuario selecciona, realiza una búsqueda en los registros almacenados (Figura 5.10).

- **Exportar:** El sistema permite realizarlo a través de 2 formatos PDF o Excel (Figura 5.11).
- **QR:** Este botón permite generar los códigos QR de los registros seleccionados de la tabla en formato PDF. Al dar clic, permite elegir a través de una lista de plantillas para etiquetas previamente configuradas (Figura 5.12). El reporte generado se encuentra en la Figura 5.13.
- **Seleccionar todo:** Se puede seleccionar todos los registros que estén visibles en la tabla. Después de este botón se encuentra con el que se puede quitar la selección.
- **Imprimir:** Genera la vista y opciones para imprimir los registros mostrados en la tabla.
- **Mostrar/Ocultar columnas:** Se puede configurar las columnas que se deseen ver en la tabla principal.

Lo que refiere a la tabla, en la primera columna se encuentran las opciones para seleccionar filas, seguido de la información referente a la tabla. En la última columna están las opciones para realizar operaciones sobre los registros como: editar, duplicar y eliminar.

Al final, se coloca un pie de página donde aparece el total de registros mostrados, y cuando se selecciona algún registro, la cantidad de los que han sido seleccionados.

Todo este módulo, tanto la tira de opciones como tabla y los títulos son totalmente responsivos, es decir, que no importa la resolución de la

pantalla o si se esté visualizando a través de un dispositivo móvil, esta se debe de adecuar a los anchos y altos que se tenga.

Descripción	Tipo de Activo	Marca	Ubicación	Número de Activo	Número de Inventario	Activo oficial	Grupo	Opciones
Pantalla LCD	CPU	HP	Apex	8274	8212	S	Atención de Emergencias	
Pantalla LCD	CPU	HP	Apex	8384	8834	S	Comunicaciones	
Pantalla LCD	CPU	HP	Apex	2838	2381	No	Patrimonio de activos fijos	
PAANTALLA LCD	Pantallas	Microsoft	Salaga	82707	1023854	S	Diarios	
Telefono IP	Consola	CISCO	Salaga	4934679	879130	S	Equipos Auxiliares	
Puerta	Banco de Baterías	Microsoft	Area Salagal	111	333	S	SDCA	
Descripcion	Escritorio	SIEMENS	Programación y Equipos	2498	2498	No	Capitalivo	
Telefono IP	Consola	CISCO	Salaga	4934679	88877888	S	Banco de Memor Gestia	
Seton	Escritorio	SIEMENS	Area Salagal	34	21	S	Activo Local	
Laptop	CPU	HP	Oficina Administración	834	134	No	Equipo de Seguridad	
USB Para el rige	Perifoneo	Microsoft	Programación y Equipos	348	78	S	Herramientas	
Banco de rige Jirga	Bancos	Sony	Programación y Equipos - Oficina Programación	12	11	S	Computadores de Escritorio (CPU)	
Monitor nuevo del Supervisor	Pantallas	Sony	Programación y Equipos - Oficina Supervisor	10	23	S	Pantallas	
Maquina para soldar	Equipo para Mantenimiento	EUTECHIC CASTOLIN	Programación y Equipos - Taller de Mantenimiento	3302	33	S	Activo	
Maquina para soldar	Equipo para Mantenimiento	EUTECHIC CASTOLIN	Programación y Equipos - Taller de Mantenimiento	3321	33	S	Activo	

Figura 5.8 Pantalla de Activos.

Agregar Registro ✕

Rellena el siguiente formulario para agregar un nuevo Activo

① Descripción:

① Activo Oficial: Si No

① Estado del Activo: Alta Baja

① Equipo en Uso: Si No

① Equipo Prestado: Si No

① Marca:

① Código de Barras:

① Modelo:

① Fecha de Ingreso:

① Tipo Activo:

① Fecha de Recepción:

① Número Activo:

① Fecha de Garantía:

① Número de Inventario:

① Fecha Capitalización:

① Número de Serie:

① Asignado a:

① Centro de Trabajo:

① Observaciones:

① Ubicación:

① Grupo:

① Estado de Registro:

Figura 5.9 Ventana de dialogo para agregar un registro en Activos.

Buscar Registro
✕

Selecciona los campos por los que desees realizar la búsqueda

Descripción:

Marca:

Modelo:

Tipo Activo:

Número Activo:

Número de Inventario:

Número de Serie:

Centro de Trabajo:

Ubicación:

Grupo:

Estado de Registro:

Activo Oficial: Sí No

Estado del Activo: Alta Baja

Equipo en Uso: Sí No

Equipo Prestado: Sí No

Código de Barras:

Fecha de Ingreso:

Fecha de Recepción:

Fecha de Garantía:

Fecha Capitalización:

Asignado a:

Observaciones:

Mostrar últimos 20
Aceptar
Cancelar

Figura 5.10 Ventana de dialogo para filtrar en Activos.

data:
1 / 1

Activo - Control de Inventarios

Descripción	Tipo de Activo	Marca	Ubicación	Número de Activo	Número de Inventario	Activo oficial	Grupo
Pantalla LCD	CPU	HP	Azotea	53764	9212	Si	Atención de Emergencias
Pantalla LCD	CPU	HP	Azotea	8394	9834	Si	Comunicaciones
Pantalla LCD	CPU	NEC	Azotea	2838	2381	No	Patrimonio de activos fijos
PANTALLA LED	Pantallas	Microsoft	Bodega	837873	10238854	Si	Scaners
Telefono IP	Consola	CISCO	Bodega	45004876	675190	Si	Equipos Auxiliares
Prueba	Banco de Baterias	Microsoft	Area Videowall	111	333	Si	SCADA
Descripcion	Escritorio	SIEMENS	Programación y Equipos	23456	3456	No	Corporativo
Telefono IP	Consola	CISCO	Bodega	45004876	9988776655	Si	Bienes de Menor Cuantía
Switch	Escritorio	SIEMENS	Area Videowall	34	21	Si	Activo Local
Laptop	CPU	HP	Oficina Administración	654	134	No	Equipo de Seguridad
USB Para el inge	Perifericos	Microsoft	Programación y Equipos	345	76	Si	Herramientas
Bocinas del Inge Jorge	Bocinas	Sony	Programación y Equipos - Oficina Programación	12	11	Si	Computadoras de Escritorio (CPU)
Monitor nuevo del Supervisor	Pantallas	Sony	Programación y Equipos - Oficina Supervisor	10	23	Si	Pantallas
Maquina para soldar	Equipo para Mantenimiento	EUTECTIC CASTOLIN	Programación y Equipos - Taller de Mantenimiento	3302	29	Si	Activo
Maquina para soldar	Equipo para Mantenimiento	EUTECTIC CASTOLIN	Programación y Equipos - Taller de Mantenimiento	33021	292	Si	Activo

Figura 5.11 Reporte en PDF generado desde la tabla de Activos.



Figura 5.12 Dialogo para seleccionar la plantilla de etiquetas para QR.



Figura 5.13 Plantilla de QR generada en formato PDF de la tabla Activos.

Siguiendo con la tira de opciones que se encuentra en el menú, el módulo de préstamos genera un documento en formato PDF para que los Activos puedan salir a otro centro de trabajo o a una revisión por mantenimiento o reparación.

El diseño de la vista del formulario lleva ese esquema debido a que se desea que la captura sea idéntica al documento generado y el usuario se sienta familiarizado inmediatamente. En la Figura 5.14 está el diseño de esta página.

Descripción	Marca	Modelo	No. Serie	No. Activo

Figura 5.14 Interfaz para generar el pase de salida de equipos y/o materiales.

Para poder cambiar los encabezados de la aplicación existe el módulo de configuración (Figura 5.15). En esta se encuentran los campos que corresponden para cambiar los títulos, así como sus tamaños y colores de fuente. En la parte inferior están ubicados los controles de carga de imágenes, así poder cambiar los logos que se encuentran en la parte superior.

Todos estos objetos se encuentran contenidos en una tabla responsiva, así poder mantener un diseño uniforme.



Figura 5.15 Pantalla de configuración de la aplicación.

Dentro de las opciones en el menú del Administrador se tiene el de Administrar Usuarios (Figura 5.16). Esta se utiliza para dar de alta usuarios con sus respectivos permisos; también se puede editar permisos, reestablecer contraseña y dar de baja usuarios. El diseño de la tabla se mantiene uniforme al de las páginas anteriores, añadiendo una opción de filtrado en la cabecera de la tabla, esquina superior derecha.



Figura 5.16 Pantalla para la Administración de Usuarios.

Para los catálogos, en el menú se despliega un submenú con todos los catálogos que ocupa la aplicación.

El diseño de todos los catálogos ha sido el mismo con el fin de que el usuario tenga una mejor experiencia al momento de interactuar con la aplicación. En la Figura 5.17 está la captura de pantalla de este módulo, donde se puede notar que se mantiene la cabecera con las opciones disponibles para operaciones con los registros (en este caso, insertar y buscar).

N	Nombre	Opciones
1	HP	[Icons]
2	Microsoft	[Icons]
3	SIEMENS	[Icons]
4	INEC	[Icons]
5	CISCO	[Icons]
6	Sony	[Icons]
8	EUTECTIC CASTOLIN	[Icons]
9	ORIX Depot	[Icons]
10	LG	[Icons]
11	FANUC	[Icons]
12	NEC	[Icons]
13	San Martín	[Icons]

Figura 5.17 Pantalla del catálogo de Marcas.

La aplicación ha sido realizada utilizando los patrones de diseño responsivo. Como evidencia de esto, se presenta el comportamiento del menú al ser utilizado desde un dispositivo móvil en la Figura 5.18.

En el encabezado desaparecen las imágenes, a fin de que se acople todo a la resolución de la pantalla, mostrando únicamente los títulos de la aplicación. Las opciones dentro del menú ahora aparecen en forma desplegable.

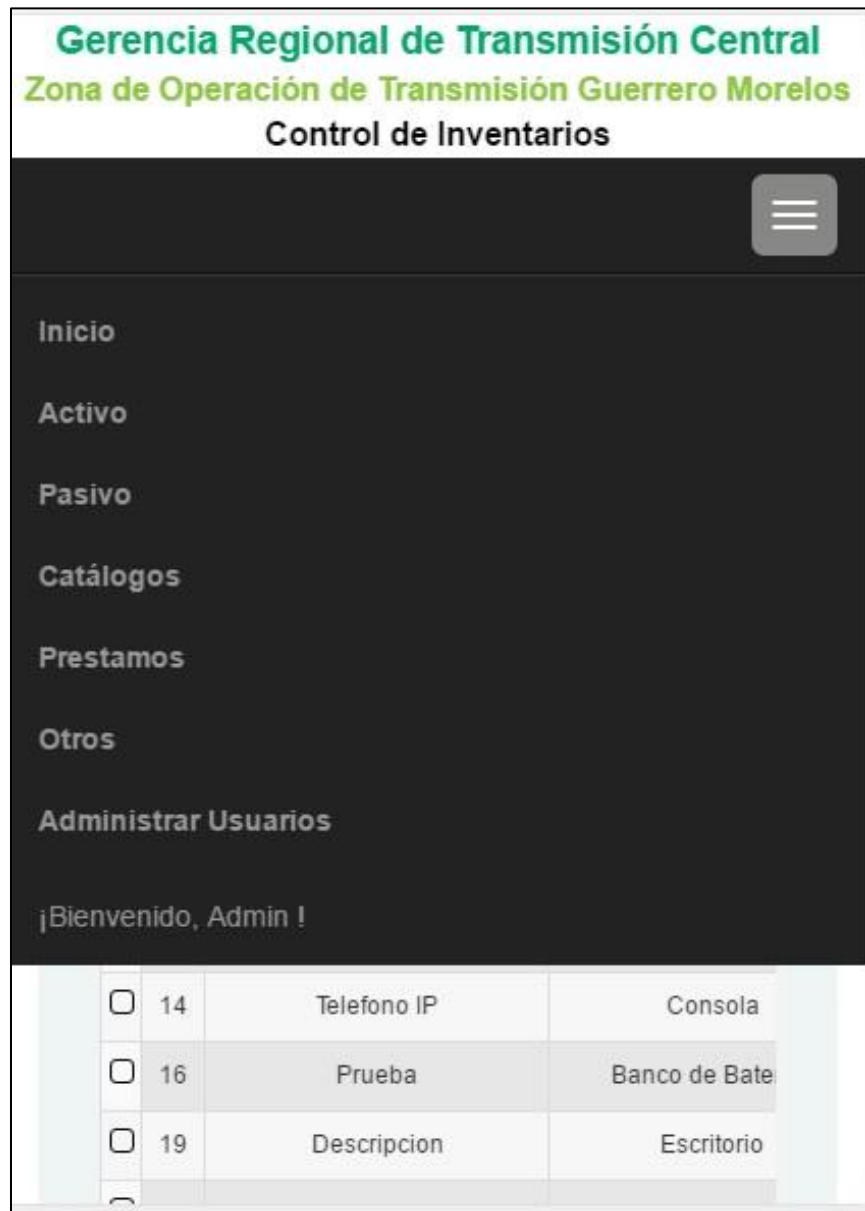


Figura 5.18 Encabezados y menú de la aplicación Web en vista responsiva.

Para el caso de las tablas, éstas se adecuan mostrando únicamente las filas que sean visibles dentro del ancho de la pantalla,

permitiendo al usuario deslizarse a través de las demás columnas. El título de la aplicación aparece en la parte superior de la tabla, seguido de los botones en las opciones para las operaciones con los registros.



Figura 5.19 Vista responsiva de las tablas en la aplicación.

Con esto se demuestra que la aplicación Web funciona correctamente, orientada hacia los requerimientos iniciales del cliente, respetando cada uno de los puntos que espera que la aplicación resuelva, dando paso a que se explique el funcionamiento de la aplicación Móvil.

Aplicación Móvil

La aplicación inicia con el SplashScreen conocida como la pantalla de presentación, se utiliza como ventana de inicio en donde al ejecutar la aplicación se descargarán e insertarán automáticamente los catálogos dentro de la base de datos interna del dispositivo móvil. Ver figura 5.21.

Se muestra un dialogo con una barra de estado que define el progreso de descarga de los catálogos dentro del SplashScreen. Ver figura 5.20.



Figura 5.20 Dialogo de descarga de catálogos.

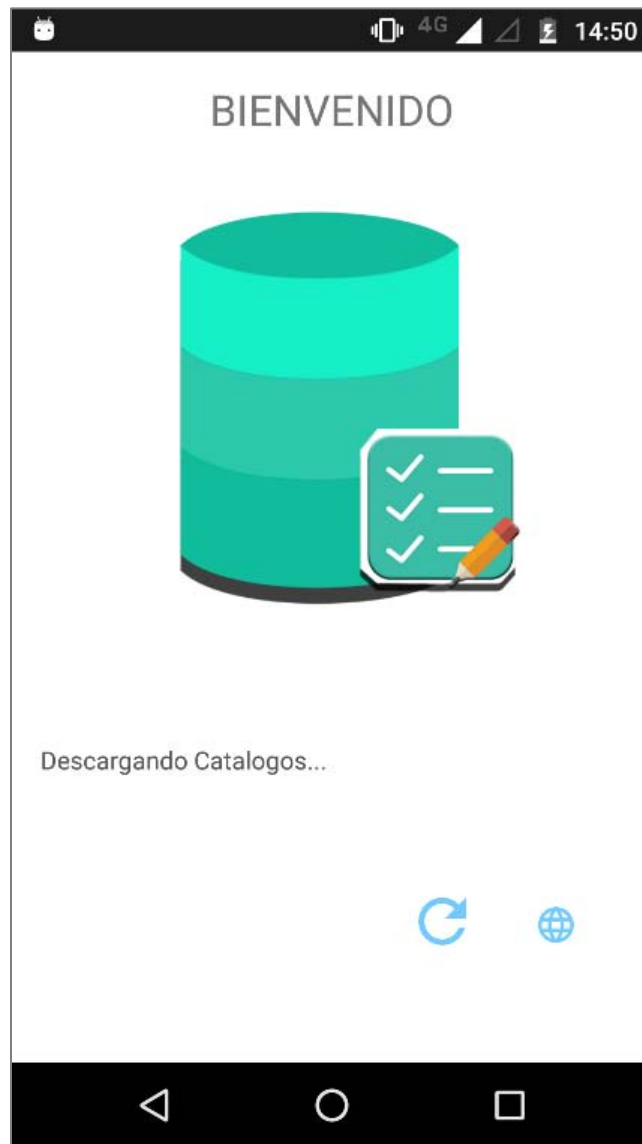


Figura 5.21 SplashScreen de la aplicación Móvil.

Si existe la posibilidad que a conexión se vea interrumpida, existen un botón para reiniciarla; si el servidor cambia de dirección IP, se cuenta con un botón que despliega un dialogo para cambiar la dirección del servidor. Ver figura 5.22.

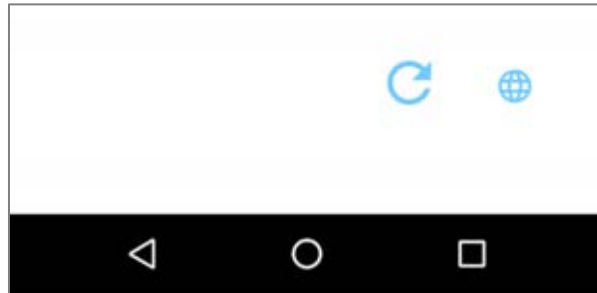


Figura 5.22 Botones en el SplashScreen.

Finalizando la etapa de descarga de información se muestra una pantalla de logueo, la cual se encarga de realizar las validaciones con respecto a los usuarios registrados en la Base de Datos del servidor. La contraseña utiliza un icono que al ser presionado permite que el usuario vea la contraseña escrita, de igual manera posee una casilla de selección para mantener recordado el usuario y la contraseña. En la figura 5.23 se muestra la ventana del logueo.

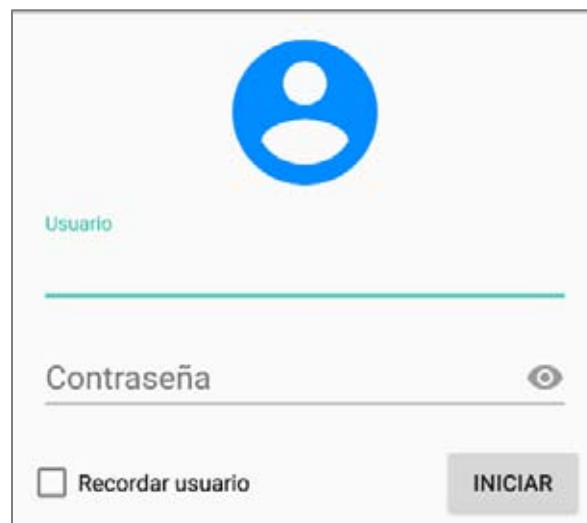


Figura 5.23 Pantalla de Login de la aplicación Móvil.

Al ser exitoso el proceso de logueo, se direccionará a la pantalla principal en donde se muestran los títulos e imágenes con las mismas características de la aplicación web. Ver figura 5.24.

El menú desplegable de la parte inferior contiene cinco opciones: Cerrar sesión, ingresar un nuevo activo, utilizar plantillas para Activo, buscar activo mediante código QR y cerrar menú desplegable



Figura 5.24 Pantalla de inicio de la aplicación Móvil.

En la figura 5.25 se muestra el formulario que contiene todos los datos necesarios para realizar el registro de un activo, solo los usuarios como los capturistas y administradores podrán visualizar esta opción.

Los datos serán enviados hacia el servidor web y procesados por este mismo, si el resultado es exitoso el cambio se verá reflejado en la aplicación móvil y el registro aparecerá en el sistema Web.

The screenshot shows a mobile application interface for adding a new asset. The title bar at the top is dark grey with the text 'Insertar Nuevo Activo' in white. Below the title bar, the form is organized into several sections, each with a horizontal line separator:

- Descripción:** A text input field with a light blue border.
- Marca:** A dropdown menu with 'HP' selected.
- Modelo:** A dropdown menu with 'L2245wg' selected.
- Tipo de Activo:** A dropdown menu with 'Aire Acondicionado' selected.
- Número de Activo:** A text input field.
- Número de inventario:** A text input field.
- Número de Serie:** A text input field.
- Centro de Trabajo:** A dropdown menu with 'ZOTGM' selected.
- Asignado a:** A dropdown menu with 'Alejandro Viveros Mendoza(9DY8K)' selected.
- Ubicación:** A dropdown menu with 'Azotea' selected.

Figura 5.25 Formulario para insertar activos en la aplicación Móvil.

En la figura 5.25 se muestra la ventana para realizar una búsqueda del activo, la aplicación Móvil utiliza su propia cámara como lector de código QR, al detectar un código válido, esta automáticamente realizará la búsqueda y obtendrá los datos que serán modificados, solo los usuarios con el rol de administrador pueden ingresar a esta opción.



Figura 5.26 Escaneo de código QR en la aplicación Móvil.

En cuestiones de rendimiento de la aplicación Web, se obtuvieron los siguientes tiempos de respuesta en unidad de segundos (s) resumidos en la tabla 6.1 (cabe mencionar que estos tiempos fueron proporcionados por la herramienta de Visual Studio).

Tabla 6.1 Tiempos de respuesta en aplicación Web.

PROCESO	TIEMPO (s)
Carga de pantalla inicial.	0.40
Carga de registros de Activos.	1.34
Carga de registro en Catálogos.	3.67
Generación de códigos QR.	4.90
Inserción de datos en Activos.	0.56
Alta de usuarios.	0.50
Validación de credenciales.	0.47

Se tiene como conclusión que los tiempos de respuesta de la aplicación son favorables, cumpliendo el requisito no funcional de que esta herramienta sea efectiva.

Los tiempos de respuesta en comunicación de la aplicación Móvil fueron de igual manera favorables, esto es debido a la infraestructura de redes en la cual está siendo utilizado el sistema que es una red interna, evitando el tráfico de información.

5.2 Trabajo Futuro

- Generar nuevas plantillas para etiquetas de QR, así ofrecer mayor variedad.
- Realizar la aplicación Móvil para distintas plataformas, como los sistemas operativos iOS y Windows Phone.
- Utilizar la arquitectura de software MVC para tener una mejor estructura en el sistema.

CAPITULO 6:

Conclusiones

CAPÍTULO VI. CONCLUSIONES

Se cumple la hipótesis debido a que el implementar el sistema Web para el control de inventario agilizó los tiempos de respuesta cliente – servidor; se mejoró el diseño de la interfaz de usuario, ofreciendo una mayor experiencia a través de componentes intuitivos y una gama de colores apropiada.

Añadiendo el módulo Móvil hubo una mejora en el proceso de registro y consulta de Activos, ya que la forma empírica en la que se realizaba era muy ineficiente. Gracias a esta aplicación, los tiempos se redujeron considerablemente, ya que el alta de equipos es realizada de manera directa, evitando procedimientos burocráticos.

Otro beneficio es que no se necesita de herramientas externas como un lector de código de barras especializado para llevar un mejor control, si no se ocupa el propio dispositivo móvil evitando así costos extras que pueden resultar muy altos, así como ahorros en mantenimiento de dichas herramientas externas.

El uso de esta aplicación es fácil, ya que cualquier usuario que tenga acceso a la red o cuente con un dispositivo móvil con sistema operativo Android, con nociones básicas de conocimientos en computación puede realizar cualquier operación que requiera, incluso controlar el contenido de la misma sin necesidad de tener un especialista en sistemas para reestructurar el código fuente.

En general, utilizar una herramienta de software basada en sistemas distribuidos puede aportar grandes beneficios al área administrativa; no

es necesario estar de frente a un problema de un grado de complejidad alto para poder poner en práctica este tipo de arquitectura de sistemas de información.

REFERENCIAS

REFERENCIAS

- AEvitas. (2013). *Windows Server*. Obtenido de AEvitas: <http://www.aevitas.com.mx/productos/software/microsoft/server.html>
- Alcázar, I., & Álvarez, M. A. (03 de Junio de 2014). *Introducción a Git y GitHub*. Obtenido de Desarrollo web: <http://www.desarrolloweb.com/articulos/introduccion-git-github.html>
- Álvarez, M. Á. (18 de Mayo de 2010). *Primeros pasos con jQuery UI*. Obtenido de Desarrollo Web: <http://www.desarrolloweb.com/articulos/primeros-paso-jquery-ui.html>
- Android. (5 de Octubre de 2016). *Conoce Android Studio*. Obtenido de developer.android: <https://developer.android.com/studio/intro/index.html>
- Anónimo. (Julio de 2009). *Metodología de desarrollo*. Obtenido de Sistemas Distribuidos: <http://sistemadistribuido.blogspot.mx/search/label/unidad%204>
- Anónimo. (2016). *Control de Versiones*. Obtenido de Producingoss: <http://producingoss.com/es/vc.html>
- ANYELGUTI. (29 de Enero de 2016). *Librerías*. Obtenido de Aprende Web: <http://aprende-web.net/librerias/>
- Azaustre, C. (04 de Noviembre de 2015). *LOS 5 PATRONES DEL RESPONSIVE DESIGN CON FLEXBOX*. Obtenido de Carlos Azaustre: <https://carlosazaustre.es/blog/los-5-patrones-del-responsive-design/>
- Baudes Rubio, G. (21 de Febrero de 2002). *Sistemas Distribuidos*. Obtenido de Departament de Matemàtiques i Informàtica. Universitat de les Illes Balears: <http://dmi.uib.es/~bbuades/sistdistr/>
- Bonobo Git Server. (2014). *Bonobo git Server*. Obtenido de <https://bonobogitserver.com/>
- campusMVP. (25 de Agosto de 2015). *Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién?* Obtenido de Campus MVP: <http://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>
- Cantone, D. (2008). *La Biblia del Programador Implementación y Debuging*. Buenos Aires: MP Ediciones.

- Cinvestav. (5 de Octubre de 2016). *Sistemas Distribuidos*. Obtenido de Cinvestav Tamaulipas: http://www.tamps.cinvestav.mx/~vjsosa/clases/sd/sistemas_distribuidos_panorama.pdf
- Ecured. (14 de Septiembre de 2016). *Modelo de Prototipos*. (Ecured) Recuperado el 07 de Septiembre de 2016, de Ecured: https://www.ecured.cu/Modelo_de_Prototipos
- Gestionix. (Octubre de 2016). *Gestionix*. Obtenido de Gestionix: <http://socios.gestionix.com/Register/Images/LogoGestionixB.JPG>
- getbootstrap. (28 de Septiembre de 2016). *Designed for everyone, everywhere*. Obtenido de getbootstrap: <http://getbootstrap.com/>
- GIT. (2009). *Empezando - Acerca del control de versiones*. Obtenido de git: <https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones>
- GIT. (2009). *Fundamentos de Git*. Obtenido de git: <https://git-scm.com/book/es/v1/Empezando-Fundamentos-de-Git>
- Google. (2016). *Material Design para Android*. Obtenido de Android Developers: <https://developer.android.com/design/material/index.html>
- IBM. (2016). *Introducción a SOA y servicios web*. Obtenido de IBM developerWorks: <http://www.ibm.com/developerworks/ssa/webservices/newto/service.html>
- itnnoacion. (Octubre de 2016). *itnnoacion*. Obtenido de itnnoacion: http://itnnoacion.com.mx/gallery_gen/01e9dbfbb2f45aa07b82ee_a6a6c5bb39_490x280.png
- jquery. (28 de Septiembre de 2016). *jQuery write less, do more*. Obtenido de jquery: <https://jquery.com/>
- KSOAP. (18 de Septiembre de 2014). *kSOAP 2*. Obtenido de kObjects.org: <http://kobjects.org/ksoap2/index.html>
- Lawrence Pfleeger, S. (2002). *Ingeniería del Software*. Argentina: Prentice Hall.
- Microsoft. (Noviembre de 2007). *Información general sobre ASP.NET*. Obtenido de Microsoft. Developer Network: [https://msdn.microsoft.com/es-es/library/4w3ex9c2\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/4w3ex9c2(v=vs.100).aspx)
- Microsoft. (2010). *Información general sobre controladores HTTP y módulos HTTP*. Obtenido de Microsoft. Developer Network:

- [https://msdn.microsoft.com/es-es/library/bb398986\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/bb398986(v=vs.100).aspx)
- Microsoft. (2010). *Interoperabilidad entre lenguajes*. Obtenido de Microsoft: [https://msdn.microsoft.com/es-es/library/a2c7tshk\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/a2c7tshk(v=vs.100).aspx)
- Microsoft. (2012). *Introducción a Visual Studio*. Obtenido de Microsoft Developer Network: [https://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx)
- Microsoft. (2014). *Microsoft SQL Server*. Obtenido de Microsoft Developer Network: <https://msdn.microsoft.com/es-mx/library/bb545450.aspx>
- Microsoft. (2015). *C#*. Obtenido de Microsoft: <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>
- Microsoft. (5 de Octubre de 2016). *¿Qué es XAML?* Obtenido de msdn.microsoft: <https://msdn.microsoft.com/es-es/library/cc295302.aspx>
- Microsoft. (26 de Septiembre de 2016). *Información general de Entity Framework*. Obtenido de Microsoft Developer Network.: [https://msdn.microsoft.com/es-es/library/bb399567\(v=vs.110\).aspx](https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx)
- Microsoft. (2016). *Internet Information Services*. Obtenido de Microsoft : <https://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.mspx>
- Microsoft. (Octubre de 2016). *Visual Studio Community*. Obtenido de Microsoft: <https://www.visualstudio.com/es/vs/community/>
- Oracle Corporation. (Octubre de 2016). *¿Cómo puedo empezar a desarrollar programas Java con Java Development Kit (JDK)?* Obtenido de Java: <https://www.java.com/es/download/faq/develop.xml>
- Oracle Corporation. (2016). *¿Qué es la tecnología JAVA y para que la necesito?* Obtenido de Java: https://www.java.com/es/download/faq/whatis_java.xml
- Perez Santillan, E., Nieto Sanchez, H., Flores Iñiguez, E., & Ortiz Gómez, D. A. (Julio de 2009). *UNIDAD 4 METODOLOGÍA PARA EL DESARROLLO DE PROYECTOS EN SISTEMAS DISTRIBUIDOS*. Obtenido de SISTEMAS DISTRIBUIDOS: <http://tesci-381-v.blogspot.mx/2009/07/unidad-4-metodologia-para-el-desarrollo.html>

- QODE. (31 de Octubre de 2012). *¿Qué es una App?* Obtenido de <http://qode.pro/blog/que-es-una-app/>
- Rubira, J. (Julio de 2011). *Solid, cinco principios básicos de diseño de clases.* Obtenido de Genbeta dev: <http://www.genbetadev.com/paradigmas-de-programacion/solid-cinco-principios-basicos-de-diseno-de-clases>
- Salinas, P. (Octubre de 2006). *Modelo de Clases. Departamento de Ciencias de la Computación.* Obtenido de Universidad de Chile: <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>
- sc.ehu. (11 de Octubre de 2016). *1 Introducción a los sistemas distribuidos.* Obtenido de <http://www.sc.ehu.es/acwlaroa/SDI/Apuntes/Cap1.pdf>
- Soluciones empresariales IT. (Octubre de 2016). *AS Invgate.* Obtenido de SEMIT: <http://semit.cl/as-invgate/>
- SQLite. (Octubre de 2016). *About SQLite.* Obtenido de SQLite: <https://sqlite.org/about.html>
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms.* USA: Prentice Hall.
- TICapacitación. (10 de Agosto de 2016). *Plantillas de Visual Studio.* Obtenido de TICapacitación: www.ticapacitacion.com
- Torres, R. (29 de Enero de 2014). *C#: Creando archivos PDF con iTextSharp.* Obtenido de Desarrolladores: <http://desarrolladores.me/2014/01/c-creando-archivos-pdf-con-itextsharp/>
- Tower, J. (2015). *Undocking Parts of the Team Explorer in Visual Studio.* Obtenido de Jonathan Tower's Blog: <http://jtower.com/blog/undocking-parts-of-team-explorer-in-visual-studio>
- w3schools. (28 de Septiembre de 2016). *AJAX Introduction.* Obtenido de w3schools: http://www.w3schools.com/ajax/ajax_intro.asp
- w3Schools. (28 de Septiembre de 2016). *CSS Introduction.* Obtenido de w3schools: http://www.w3schools.com/css/css_intro.asp
- w3schools. (28 de Septiembre de 2016). *HTML Introduction.* Obtenido de w3schools: http://www.w3schools.com/html/html_intro.asp
- w3schools. (28 de Septiembre de 2016). *HTML Responsive Web Design.* Obtenido de w3schools: http://www.w3schools.com/html/html_responsive.asp

w3schools. (28 de Septiembre de 2016). *JavaScript Tutorial*. Obtenido de w3schools: <http://www.w3schools.com/js/>

ZOTGM. (Agosto de 2016). *Pagina prrincipal*. Obtenido de Zona de Operación de Transmisión Guerrero Morelos: 10.25.117.65

ÍNDICE DE FIGURAS	Página
Figura 1.1 Fachada de la ZOTGM.	15
Figura 1.2 Ubicación de la ZOTGM.	21
Figura 1.3 Organigrama de la ZOTGM.	24
Figura 1.4 Logotipo de Gestionix.	26
Figura 1.5 Ejemplo de vista de catálogos de Gestionix.	27
Figura 1.6 Logotipo de Assets.	28
Figura 1.7 Página principal de Assets.....	29
Figura 1.8 Logotipo de Bind erp.....	30
Figura 1.9 Ejemplo de pantalla de registro en Bind erp.	31
Figura 2.1 Página Principal del Sistema Actual.	36
Figura 2.2 Formulario de búsqueda en Activos	37
Figura 2.3 Formulario registro en Activos.	38
Figura 2.4 Tabla de resultados de la búsqueda en Activos.	39
Figura 2.5 Opciones de personalización de la Tabla de Activos.....	40
Figura 2.6 Interfaz gráfica del Catálogo de Usuarios.	41
Figura 2.7 Configuración de la Interfaz Gráfica.....	42
Figura 2.8 Configuración de la conexión a la Base de Datos.	42
Figura 2.9 Estructura de caso particular de la Arquitectura Cliente – Servidor.	48
Figura 2.10 Fases de la Metodología para Sistemas Distribuidos.	57
Figura 3.1 Logotipo de Visual Studio.	61
Figura 3.2 Tabla comparativa de productos de Visual Studio.....	62
Figura 3.3 Logotipo de Visual Studio Community 2015.	63
Figura 3.4 Logotipo de IIS.....	64
Figura 3.5 Logotipo de Windows Server.	65
Figura 3.6 Logotipo de CSS3.....	66

Figura 3.7 Logotipo de HTML5.	67
Figura 3.8 Logotipo de Bootstrap.....	68
Figura 3.9 Esquema del patrón Mostly Fluid.....	70
Figura 3.10 Logotipo de JavaScript.	71
Figura 3.11 Logotipo de JQuery.....	72
Figura 3.12 Logotipo jQuery UI.....	72
Figura 3.13 Logotipo de JQuery.....	73
Figura 3.14 Logotipo ASP.net.....	74
Figura 3.15 Logotipo del lenguaje C#.	75
Figura 3.16 Logotipo de Entity Framework.	76
Figura 3.17 Logotipo de Microsoft SQL Server.....	77
Figura 3.18 Logo de iTextSharp.....	79
Figura 3.19 Logotipo ZXing.....	80
Figura 3.20 Logotipo de Android Studio.....	82
Figura 3.21 Logotipo de Java.....	83
Figura 3.22 Logotipo del lenguaje XAML.....	84
Figura 3.23 Ejemplo de pantallas creadas con Material Design.	87
Figura 3.24 Logotipo de SQLite.....	88
Figura 3.25 Funcionamiento de kSOAP.....	90
Figura 3.26 Control de versiones local.....	91
Figura 3.27 Control de versiones centralizadas.....	92
Figura 3.28 Control de versiones distribuido.....	94
Figura 3.29 Logotipo del control de versiones git.....	95
Figura 3.30 Diagrama de flujo de información en Git.....	96
Figura 3.31 Logotipo del servidor Bonobo Git server.....	97
Figura 3.32 Esquema de un sistema distribuido en modelo 4 capas.	98
Figura 4.1 Especificación de operaciones en cada sistema.	103

Figura 4.2 Diagrama de Contexto.	104
Figura 4.3 Implementación del sistema distribuido.	106
Figura 4.4 Asset de los iconos de la aplicación Web.	113
Figura 4.5 Asset de los iconos de la aplicación Móvil.	114
Figura 4.6 Ejemplo de estructura de una clase.	115
Figura 4.7 Diagrama de clases padre.	116
Figura 4.8 Estructura de una clase en Visual Studio.	119
Figura 4.9. Diagrama de clases de contenido.	120
Figura 4.10 Diagrama de clases de catálogos.	121
Figura 4.11 Diagrama de clases de los controladores.	122
Figura 4.12 Diagrama de clases de los servicios Web.	123
Figura 4.13 Diagrama de clases de las entidades.	124
Figura 4.14 Diagrama de clases de los modelos.	124
Figura 4.15 Diagrama de clases de AppData.	125
Figura 4.16 Diagrama de clase para generar códigos QR.	126
Figura 4.17 Diagrama de clases Aplicación Móvil.	127
Figura 4.18 Modelo de datos del proyecto.	129
Figura 4.19 Diagrama de modelo de datos de la autenticación.	130
Figura 4.20 Diagrama Entidad – Relación de los Activos.	131
Figura 4.21 Diagrama Entidad – Relación de la autenticación.	132
Figura 4.22 Mapa de Sitio de la Aplicación Web.	133
Figura 4.23 Mapa de navegación de la aplicación Móvil.	135
Figura 4.24 Wireframe de la página de Inicio.	136
Figura 4.25 Wireframe de la pantalla de inicio para usuarios autenticados.	137
Figura 4.26 Wireframe para los catálogos.	138
Figura 4.27 Wireframe para la página de Activos.	139

Figura 4.28 Wireframe para el Pase de Salida.	140
Figura 4.29 Wireframe para la página de la configuración.....	141
Figura 4.30 Wireframe del Splash Screen.	142
Figura 4.31 Wireframe de iniciar sesión.....	143
Figura 4.32 Página principal de la aplicación Móvil.	144
Figura 4.33 Formulario para insertar registros.	145
Figura 4.34 Esquema de trabajo de Ajax.	147
Figura 4.35 Esquema de trabajo de Ksoap.....	148
Figura 4.36 Diagrama de caso de uso: Administrador y Superusuario.	150
Figura 4.37 Diagrama de casos de uso: Capturista e Invitado.	151
Figura 4.38 Procedimiento para alta de Activo.	153
Figura 4.39 Procedimiento para la modificación de Activo.	154
Figura 4.40 Procedimiento para baja de Activo.	155
Figura 4.41 Procedimiento para la búsqueda por filtros.....	156
Figura 4.42 Procedimiento para dar de alta un usuario.	157
Figura 4.43 Procedimiento para modificación de usuario.	158
Figura 4.44 Procedimiento para modificar el rol del usuario.	159
Figura 4.45 Procedimiento para reestablecer contraseña.	160
Figura 4.46 Procedimiento para generar reporte de los Activos.	161
Figura 4.47 Procedimiento para generar plantilla de QR.	162
Figura 4.48 Generación de reporte de pase de salida.	163
Figura 4.49 Procedimiento para iniciar sesión.	164
Figura 4.50 Procedimiento para la búsqueda a través del código QR.	165
Figura 5.1 Pantalla de inicio de la aplicación Web.....	169
Figura 5.2 Pantalla de logueo de la aplicación Web.	170

Figura 5.3 Pantalla de inicio con usuario autenticado.....	171
Figura 5.4 Opciones para el tipo de usuario administrador.....	171
Figura 5.5 Opciones para el tipo de usuario Superusuario.	171
Figura 5.6 Opciones para el tipo de usuario capturista.....	172
Figura 5.7 Opciones para el tipo de usuario invitado.....	172
Figura 5.8 Pantalla de Activos.	174
Figura 5.9 Ventana de dialogo para agregar un registro en Activos. ..	174
Figura 5.10 Ventana de dialogo para filtrar en Activos.	175
Figura 5.11 Reporte en PDF generado desde la tabla de Activos.	175
Figura 5.12 Dialogo para seleccionar la plantilla de etiquetas para QR.	176
Figura 5.13 Plantilla de QR generada en formato PDF de la tabla Activos.	176
Figura 5.14 Interfaz para generar el pase de salida de equipos y/o materiales.....	177
Figura 5.15 Pantalla de configuración de la aplicación.....	178
Figura 5.16 Pantalla para la Administración de Usuarios.	178
Figura 5.17 Pantalla del catálogo de Marcas.....	179
Figura 5.18 Encabezados y menú de la aplicación Web en vista responsiva.....	180
Figura 5.19 Vista responsiva de las tablas en la aplicación.....	181
Figura 5.20 Dialogo de descarga de catálogos.....	182
Figura 5.21 SplashScreen de la aplicación Móvil.....	183
Figura 5.22 Botones en el SplashScreen.....	184
Figura 5.23 Pantalla de Login de la aplicación Móvil.	184
Figura 5.24 Pantalla de inicio de la aplicación Móvil.....	185
Figura 5.25 Formulario para insertar activos en la aplicación Móvil....	186

Figura 5.26 Escaneo de código QR en la aplicación Móvil. 187

ÍNDICE DE TABLAS	Página
Tabla 1.1 Objetivos e Indicadores de la ZOTGM.....	23
Tabla 2.1 Comparación de las etapas de desarrollo entre las metodologías seleccionadas.....	45
Tabla 4.1 Convenciones de nomenclaturas.....	107
Tabla 4.2 Estilos de codificación.....	109
Tabla 4.3 Paleta de colores del proyecto.....	112
Tabla 4.4 Descripción de las clases padre	117
Tabla 4.5 Descripción de las clases de la Aplicación Móvil.	127
Tabla 6.1 Tiempos de respuesta en aplicación Web.	188
Tabla B.1 Diccionario de datos del Activo.....	210
Tabla C.1 Diccionario de datos de la Autenticación.....	215

ANEXOS

ANEXO A. Cuestionario de especificaciones del sistema.

1. ¿Considera que la Aplicación Web que utiliza para el Control de Inventario necesita ser actualizado?

Sí, definitivamente.

2. ¿Qué procedimientos realizados actualmente necesitan ser mejorados?

Registro de los elementos, plantillas, seguridad y los filtros.

3. ¿Qué funciones nuevas se implementarán para el sistema?

Las plantillas y poder duplicar registros.

4. ¿El sistema requiere la generación de reportes? ¿En qué formato?

En formato PDF y Excel para los resultados de la búsqueda por filtros y un listado general de todos los registros.

5. ¿Cuáles son los criterios por los que se generarán estos reportes?

Depende de las necesidades del usuario; deben de existir todos los filtros posibles para garantizar versatilidad.

6. En temas de seguridad, ¿Con cuántos usuarios contará la aplicación y cuáles restricciones tendrán?

Administrador: Podrá realizar todas las operaciones sobre la información del Activo y los catálogos, así como dar de alta o baja usuarios y asignar

roles de usuario. También podrá cambiar la configuración de colores del sistema.

Superusuario: Sólo podrá realizar todas las operaciones sobre la información del Activo y los catálogos, así como cambiar la configuración de colores del sistema.

Invitado: Solamente podrá ver información del Activo.

Capturista: Sólo podrá agregar datos al Activo y los catálogos.

7. ¿Requiere que el sistema contenga apartados para poder modificar títulos de la aplicación y logos oficiales?

Sí.

8. ¿Desea que la aplicación pueda ser utilizada a través de diferentes dispositivos electrónicos como laptops, tabletas o celulares?

Sí, me gustaría.

9. ¿Qué información almacenará dentro del módulo Activos?

Descripción del activo, tipo, modelo, marca, número de serie, número de inventario, número de activo fijo, fecha de ingreso al sistema, fecha de capitalización, activo oficial, estado del activo, ubicación, centro de trabajo, usuario al que pertenece el activo.

10. ¿Qué información se almacenará dentro del módulo Pasivo?

Se tiene que corroborar con la información contable (verlo con el superintendente del área o el personal administrativo).

11. ¿Qué información se almacenará dentro del módulo Software Instalado?

Tener un registro de licencias y software, así como la información acerca del equipo en donde se encuentra instalado. Nota: sólo el administrador puede ver la clave de la licencia de los productos.

12. ¿Cuáles catálogos son importantes almacenar en el sistema?

Usuarios, modelos, marcas, tipo de activo, estado del registro, ubicación, centros de trabajo y grupos.

13. Comentarios u observaciones del cliente.

Agregar un módulo para el pase de salida para cuando tenga que salir un equipo a un centro de trabajo o mantenimiento. Se debe proporcionar una propuesta de diseño del documento a partir del ya existente.

ANEXO B. Diccionario de datos del Activo

En la Tabla B.1 se encuentra el diccionario de datos de la base de datos de Activos.

Tabla B.1 Diccionario de datos del Activo.

Objeto	Nombre	Tipo de Dato	Descripción
ENTIDAD	CTipoActivo	—	Tipos que existen para clasificar los activos ingresados al Sistema.
ATRIBUTO	tipo_activo_id	entero	Llave primaria de la tabla CTIPOACTIVO.
ATRIBUTO	Nombre	nvarchar(50)	Nombre del tipo de Activo.
ATRIBUTO	Descripción	nvarchar(300)	Descripción referente al tipo de activo.
ENTIDAD	CMarca	—	Marcas más comunes en los activos.
ATRIBUTO	marca_id	entero	Llave primaria de la tabla CMARCA.
ATRIBUTO	Nombre	nvarchar(50)	Nombre de la marca.
ENTIDAD	CModelo	—	Registros de los modelos de los activos.
ATRIBUTO	modelo_id	entero	Llave primaria de la tabla CMODELO.
ATRIBUTO	Nombre	nvarchar(50)	Caracteres que corresponden al modelo.
ATRIBUTO	Descripción	nvarchar(300)	Descripción referente al modelo.
ENTIDAD	CUbicacion	—	Ubicaciones oficiales dentro del centro de trabajo.

Objeto	Nombre	Tipo de Dato	Descripción
ATRIBUTO	ubicacion_id	entero	Llave primaria de la tabla CUBICACION.
ATRIBUTO	Nombre	nvarchar(50)	Nombre oficial de la ubicación.
ENTIDAD	CEstadoRegistro	—	Estado que puede tener un registro cuando es ingresado al sistema.
ATRIBUTO	estado_registro_id	entero	Llave primaria de la tabla CESTADOREGISTRO.
ATRIBUTO	Nombre	nvarchar(30)	Nombre atribuido al estado del registro.
ENTIDAD	CUusuario	—	Usuarios dados de alta en el centro de trabajo.
ATRIBUTO	usuario_id	entero	Llave primaria de la tabla CUSUARIO.
ATRIBUTO	Nombre	nvarchar(30)	Nombre de pila del usuario.
ATRIBUTO	apellido_paterno	nvarchar(30)	Apellido paterno del usuario.
ATRIBUTO	apellido_materno	nvarchar(30)	Apellido materno del usuario.
ATRIBUTO	Rpe	nchar(5)	Identificador interno para los empleados del centro de trabajo.
ATRIBUTO	Habilitado	nchar(20)	Identificar el estado actual del empleado.
ENTIDAD	CCentroTrabajo	—	Centros de trabajo afiliados a la misma institución.
ATRIBUTO	centro_trabajo_id	entero	Llave primaria de la tabla CCENTROTRABAJO.
ATRIBUTO	Nombre	nvarchar(50)	Nombre oficial del Centro de Trabajo.

Objeto	Nombre	Tipo de Dato	Descripción
ATRIBUTO	activo_id	entero	Llave primaria de la tabla ACTIVO.
ATRIBUTO	Descripción	nvarchar(300)	Descripción del Activo.
ATRIBUTO	tipo_activo_id	entero	Llave foránea referencia a la tabla CTIPOACTIVO.
ATRIBUTO	marca_id	entero	Llave foránea referencia a la tabla CMARCA.
ATRIBUTO	modelo_id	entero	Llave foránea referencia a la tabla CMODELO.
ATRIBUTO	ubicación_id	entero	Llave foránea referencia a la tabla CUBICACION.
ATRIBUTO	estado_registro_id	entero	Llave foránea referencia a la tabla CESTADOREGISTRO.
ATRIBUTO	numero_activo	entero	Identificador de los activos.
ATRIBUTO	numero_inventario	entero	Identificador para el inventario.

Objeto	Nombre	Tipo de Dato	Descripción
ATRIBUTO	usuario_id	entero	Llave foránea referencia a la tabla CUSUARIO.
ATRIBUTO	activo_oficial	bit	Reconocer entre un registro oficial o no.
ATRIBUTO	numero_serie	nvarchar(50)	Seriación del bien a registrar.
ATRIBUTO	centro_trabajo_id	entero	Llave foránea referencia a la tabla CCENTROTRABAJO.
ATRIBUTO	codigo_barra	nvarchar(30)	Código de barras comercial incluido en el producto.
ATRIBUTO	codigo_barra_propio	nvarchar(30)	Código de barras generado por el sistema.
ATRIBUTO	fecha_ingreso_sistema	nvarchar(12)	Fecha en que el activo se da de alta en el sistema.
ATRIBUTO	fecha_recepcion	nvarchar(12)	Fecha en que el activo se recibió en el centro de trabajo.

Objeto	Nombre	Tipo de Dato	Descripción
ATRIBUTO	fecha_capitalizacion	nvarchar(12)	Fecha de capitalización del activo.
ATRIBUTO	fecha_garantia	nvarchar(12)	Fecha límite para completar el registro.
ATRIBUTO	estado_activo	nvarchar(5)	Estado en el que se encuentra el bien (alta o baja).
ATRIBUTO	equipo_uso	bit	Equipo está siendo utilizado.
ATRIBUTO	equipo_prestado	Bit	Equipo se encuentra en otro centro de trabajo en calidad de préstamo.
ATRIBUTO	observaciones	nvarchar(500)	Comentarios y/o observaciones del activo.

ANEXO C. Diccionario de datos de la Autenticación

En la Tabla C.1 se encuentra el diccionario de datos de la base de datos de la autenticación.

Tabla C.1 Diccionario de datos de la Autenticación.

Objeto	Nombre	Tipo de Dato	Descripción
ENTIDAD	Roles	—	Roles existentes para los Usuarios.
ATRIBUTO	rol_id	nvarchar(128)	Llave primaria de la tabla Roles.
ATRIBUTO	nombre	nvarchar(256)	Nombre del tipo de Activo.
ENTIDAD	Users	—	Usuarios dados de alta para login del sistema.
ATRIBUTO	Id	nvarchar(128)	Llave primaria de la tabla Users.
ATRIBUTO	Email	nvarchar(256)	Correo Electrónico del usuario.
ATRIBUTO	EmailConfirmed	bit	Confirma cuenta a través del correo electrónico.

Objeto	Nombre	Tipo de Dato	Descripción
ATRIBUTO	SecurityStamp	nvarchar(256)	Sello distintivo de la cuenta.
ATRIBUTO	PasswordHash	nvarchar(MAX)	Contraseña del usuario encriptada.
ATRIBUTO	PhoneNumber	nvarchar(MAX)	Número de teléfono del usuario.
ATRIBUTO	PhoneNumberConfirmed	Entero	Confirma cuenta a través del número telefónico.
ATRIBUTO	TwoFactorEnabled	bit	Método para autenticar usuarios.
ATRIBUTO	AccessFailedCount	int	Número de fallas al tratar de ingresar a la cuenta.
ATRIBUTO	UserName	nvarchar(256)	Nombre de usuario o alias.
ATRIBUTO	rol_id	Nvarchar(128)	Llave foránea referente a la tabla Roles