

Aplicación móvil y sistema de información como plataformas de desarrollo para la gestión de inventarios de la universidad de Cundinamarca seccional Ubaté (Udecinv) caso de estudio programa ingeniería de sistemas

Nestor Ivan Castellanos Laiton y Manuel Stiven Rodríguez Lamprea

Programa de Ingeniería de Sistemas, Universidad de Cundinamarca

Trabajo de grado

Ana Lucia Hurtado Mesa

Abril 19, 2021

Dedicatoria

Dedico este trabajo de grado en primer lugar a Dios, que me ha permitido seguir adelante en este proceso dándome las capacidades intelectuales y mentales para afrontar cada reto que se me presentó a lo largo del desarrollo de este proyecto. A mis padres por brindarme su apoyo incondicional, la motivación y tener la paciencia para acompañarme en cada una de mis metas propuestas, ellos siempre están dándome sus consejos y son la guía para tomar el mejor camino en la toma de las decisiones más importantes para mi vida. A mis cuatro hermanos por brindarme sus consejos y estar siempre cuando los necesito. Finalmente dedico este trabajo de grado a mi compañero y amigo Stiven Rodríguez, por su compromiso con el proyecto y con quien conformamos un equipo de trabajo que nos permitió culminar este proceso de la mejor manera.

Nestor Ivan Castellanos Laiton

Dedico este proyecto de grado a Dios todo poderoso por haberme permitido llegar al final de mi primer proceso académico profesional como Ingeniero de sistemas, de igual manera a mis padres por todo el acompañamiento durante estos cinco años de carrera donde en cada momento me brindaron todo su apoyo y acompañamiento, también a mis hermanos, familiares y amigos que motivaron a seguir adelante en mis estudios, y sin lugar a duda dedico este proyecto a mi compañero Nestor Castellanos por su dedicación, compromiso y motivación tanto en el proyecto como en las asignaturas, culmino esta etapa sabiendo que no fue fácil, pero de ello he aprendido aceptar aquellos errores que no podía aceptar fácilmente, aprendí tanto de esto que a pesar de que esos errores que se me dificultaban ya sean difíciles o sencillos me hacían seguir avanzado más, pero esto no me detenía, me hacía sentir más orgulloso de lo que puedo hacer y enseñarle a muchas personas.

Manuel Stiven Rodríguez Lamprea

Agradecimientos

Queremos agradecer a la Universidad de Cundinamarca por habernos permitido adquirir los conocimientos y habilidades necesarias para poder formarnos como profesionales, dándonos la oportunidad de poder aplicarlos durante todo el proceso de nuestra formación académica hasta el día de hoy, donde estamos presentando nuestro proyecto de grado con la ayuda de nuestra directora Ana Lucia Hurtado Mesa, a quien le agradecemos por su compromiso y apoyo durante todo este proceso. Agradecemos de igual manera a la docente Yeny Liliana Casas Méndez por habernos dado la idea principal de nuestro proyecto, así como a nuestros docentes y compañeros que estuvieron acompañándonos durante los últimos cinco años y lograron hacer de este proceso una experiencia inolvidable y de mucho aprendizaje para la vida. Finalmente, y no menos importante queremos agradecer a nuestros padres y familia que siempre creyeron en nuestras capacidades y han puesto de su parte para lograr este gran objetivo propuesto.

Índice

1.	Introducción	1
2.	Resumen.....	3
3.	Definición del problema	5
3.1.	Descripción de la situación problemática.....	5
3.2.	Formulación del problema	7
4.	Justificación	8
5.	Objetivos	10
5.1.	Objetivo general	10
5.2.	Objetivos específicos.....	10
6.	Alcances y limitaciones	11
7.	Marco de referencia	15
7.1.	Estado del arte	15
7.2.	Marco teórico	23
7.3.	Marco conceptual	26
7.4.	Marco legal.....	31
8.	Marco metodológico	33
8.1.	Aspectos metodológicos de desarrollo.....	34
9.	Cronograma.....	37
10.	Ingeniería del proyecto	44
10.1.	Planeación.....	44
10.1.1.	Requerimientos funcionales	44
10.1.2.	Requerimientos no funcionales	46
10.1.3.	Historias de Usuario	48
10.1.4.	Velocidad del proyecto.....	58
10.1.5.	División de iteraciones	59
10.1.6.	Plan de Entregas	60
10.1.7.	Reunión inicial de iteración.....	61
10.1.8.	Roles XP.....	62
10.2.	Diseño.....	64
10.2.1.	Simplicidad en el diseño.....	64
10.2.2.	Metáfora del sistema	64
10.2.3.	Tarjetas CRC	75
10.2.4.	Caso de uso sitio web	81

10.2.5.	Caso de uso aplicación móvil	83
10.2.6.	Diagrama de clase sistema multiplataforma.....	104
10.2.7.	Base de datos	105
10.3.	Codificación.....	110
10.3.1.	Integraciones frecuentes	110
10.3.2.	Codificación iteración 1	111
10.3.3.	Codificación iteración 2	135
10.3.4.	Codificación iteración 3	146
10.4.	Pruebas.....	158
10.4.1.	Pruebas unitarias.....	158
10.4.2.	Pruebas de aceptación	162
10.4.3.	Detección y documentación de errores.....	168
10.4.4.	Manuales y documentación	170
11.	Resultados	171
12.	Conclusiones	172
13.	Referencias bibliográficas.....	174
14.	Anexos	178

Índice de tablas

Tabla 1 Requerimientos funcionales	44
Tabla 2 Requerimientos no funcionales	46
Tabla 3 Historias de usuario.....	48
Tabla 4 Formato historias de usuario	49
Tabla 5 Historia de usuario sitio web ingreso al sistema.	49
Tabla 6 Historia de usuario sitio web entrega de inventario.	50
Tabla 7 Historia de usuario sitio web búsqueda de elementos.....	50
Tabla 8 Historia de usuario sitio web opciones de elementos.....	51
Tabla 9 Historia de usuario sitio web creación de código de barras	51
Tabla 10 Historia de usuario sitio web encargados.....	51
Tabla 11 Historia de usuario sitio web asignación de inventarios.	51
Tabla 12 Historia de usuario sitio web solicitudes usuario dependencia.	52
Tabla 13 Historia de usuario sitio web respuesta solicitudes.....	53
Tabla 14 Historia de usuario sitio web reportes.....	53
Tabla 15 Historia de usuario sitio web notificaciones.	54
Tabla 16 Historia de usuario aplicativo móvil ingreso a la aplicación	54
Tabla 17 Historia de usuario aplicativo móvil entrega de inventario	55
Tabla 18 Historia de usuario aplicativo móvil búsqueda de elementos	55

Tabla 19 Historia de usuario aplicativo móvil opciones de elementos.	56
Tabla 20 Historia de usuario aplicativo móvil asignación de inventarios.....	56
Tabla 21 Historia de usuario aplicativo móvil solicitudes usuario dependencia.	57
Tabla 22 Historia de usuario aplicativo móvil respuesta solicitudes.	57
Tabla 23 Historia de usuario aplicativo móvil notificaciones.....	57
Tabla 24 Velocidad del proyecto.	58
Tabla 25 Iteración 01.....	59
Tabla 26 Iteración 02.....	60
Tabla 27 Iteración 03.....	60
Tabla 28 Plan de entregas.	61
Tabla 29 Reunión inicial de iteración	61
Tabla 30 Roles XP.	63
Tabla 31 Tarjetas CRC Login.	76
Tabla 32 Tarjetas CRC entregas de inventario.	76
Tabla 33 Tarjetas CRC búsqueda de elementos.....	76
Tabla 34 Tarjetas CRC opciones de elementos.....	77
Tabla 35 Tarjetas CRC creación de código de barras.	77
Tabla 36 Tarjetas CRC encargados.....	77
Tabla 37 Tarjetas CRC asignación de inventarios	77
Tabla 38 Tarjetas CRC solicitudes usuario dependencia.	78

Tabla 39 Tarjetas CRC respuesta solicitud.	78
Tabla 40 Tarjetas CRC reportes.	78
Tabla 41 Tarjetas CRC notificaciones.	79
Tabla 42 Lista de casos de uso web usuario administrador.	79
Tabla 43 Lista de casos de uso web usuario dependencia.....	80
Tabla 44 Lista de casos de uso móvil usuario administrador.....	80
Tabla 45 Lista de casos de uso móvil usuario dependencia.	80
Tabla 46 Actor sitio web administrador.....	81
Tabla 47 Actor sitio web usuario dependencia.	82
Tabla 48 Actor aplicativo móvil administrador	84
Tabla 49 Actor aplicativo móvil usuario dependencia.....	84
Tabla 50 Caso de uso web: Login.	85
Tabla 51 Caso de uso web: Encargados.	86
Tabla 52 Caso de uso web: Búsquedas.	87
Tabla 53 Caso de uso aplicativo móvil: Login.....	88
Tabla 54 Caso de uso aplicativo móvil: Búsquedas.	89
Tabla 55 Caso de uso aplicativo móvil: Opciones elementos.....	90
Tabla 56 Caso de uso web: Inventarios.....	91
Tabla 57 Caso de uso web: Opciones elementos.	92
Tabla 58 Caso de uso web: Generar código de barras	93

Tabla 59 Caso de uso web: Entrega de inventarios.....	94
Tabla 60 Caso de uso web: Reportes.	95
Tabla 61 Caso de uso aplicativo móvil: entregas de inventario.....	96
Tabla 62 Caso de uso aplicativo móvil: Asignación de inventarios.	97
Tabla 63 Caso de uso web: Solicitudes administrador.....	98
Tabla 64 Caso de uso web: Solicitudes usuario dependencia.	99
Tabla 65 Caso de uso web: Notificaciones.	100
Tabla 66 Caso de uso aplicativo móvil: Solicitudes administrador.	101
Tabla 67 Caso de uso aplicativo móvil: Solicitudes usuario dependencia.....	101
Tabla 68 Caso de uso aplicativo móvil: Notificaciones.....	102
Tabla 69 Tabla sesiones.	107
Tabla 70 Tabla checklist.	108
Tabla 71 Tabla elementos.	108
Tabla 72 Tabla encargados.....	109
Tabla 73 Tabla inventarios.....	109
Tabla 74 Tabla solicitudes.....	109
Tabla 75 Pruebas de compatibilidad.	162
Tabla 76 Detección y documentación de errores.	168

Índice de figuras

Figura 1 Estadística sobre sistema operativo en teléfonos móvil	11
Figura 2 <i>Ciclo de vida de metodología XP</i>	35
Figura 3 Cronograma de actividades sistema multiplataforma.....	38
Figura 4 Cronograma de actividades sistema multiplataforma Fase 1.	39
Figura 5 Cronograma de actividades sistema multiplataforma Fase 2.	40
Figura 6 Cronograma de actividades sistema multiplataforma iteración 1.....	41
Figura 7 Cronograma de actividades sistema multiplataforma iteración 2.....	42
Figura 8 Cronograma de actividades sistema multiplataforma iteración 3.....	43
Figura 9 Arquitectura de cliente servidor.	65
Figura 10 Arquitectura del sistema multiplataforma.	66
Figura 11 Bootstrap 4.....	67
Figura 12 CCS3.....	68
Figura 13 Html5.....	68
Figura 14 JavaScript.....	69
Figura 15 Git.....	69
Figura 16 nodejs.....	70
Figura 17 Express js.....	70
Figura 18 Heroku.	71
Figura 19 Mysql.....	71

Figura 20 Android Studio.	72
Figura 21 Rest Api.	72
Figura 22 php7.	73
Figura 23 Google Play.	73
Figura 24 Gama de colores	74
Figura 25 <i>Logotipo</i>	75
Figura 26 Diagrama de caso de uso sitio web.....	81
Figura 27 Diagrama de caso de uso aplicación móvil.	83
Figura 28 Diagrama de secuencia Login sitio web	86
Figura 29 Diagrama de secuencia encargados sitio web.....	87
Figura 30 Diagrama de secuencia búsquedas sitio web.	88
Figura 31 Diagrama de secuencia Login aplicación móvil.....	89
Figura 32 Diagrama de secuencia búsquedas aplicación móvil.....	90
Figura 33 Diagrama de secuencia opciones elementos móvil.	91
Figura 34 Diagrama de secuencia inventario sitio web.	92
Figura 35 Diagrama de secuencia elementos sitio web.	93
Figura 36 Diagrama de secuencia generar código de barras sitio web.	94
Figura 37 Diagrama de secuencia entregas sitio web	95
Figura 38 Diagrama de secuencia reportes sitio web.....	96
Figura 39 Diagrama de secuencia entregas aplicación móvil.....	97

Figura 40	Diagrama de secuencia asignación de inventarios aplicación móvil.....	97
Figura 41	Diagrama de secuencia solicitudes administrador sitio web	98
Figura 42	Diagrama de secuencia solicitudes usuario dependencia sitio web.....	100
Figura 43	Diagrama de secuencia notificaciones sitio web.	101
Figura 44	Diagrama de secuencia solicitudes aplicación móvil.	102
Figura 45	Diagrama de secuencia notificaciones aplicación móvil.	103
Figura 46	Diagrama de clase sistema multiplataforma.	104
Figura 47	Modelo entidad relación base de datos.....	105
Figura 48	Modelo relacional de base de datos	106
Figura 49	Login sitio web.	111
Figura 50	Código de validación de usuarios sitio web.	112
Figura 51	Código conexión a la base de datos del sistema multiplataforma.	113
Figura 52	Menú principal de usuario administrador sitio web.	113
Figura 53	Menú principal de usuario dependencia sitio web.....	114
Figura 54	Código de lenguaje estructurado de html.	115
Figura 55	Ventana de búsquedas usuario administrador sitio web.....	115
Figura 56	Ventana de búsquedas usuario dependencia sitio web.	116
Figura 57	Algoritmo en JavaScript para filtros en sitio web.	117
Figura 58	Ventana para ver información de elementos en sitio web.	117
Figura 59	Ventana para modificar elementos.	118

Figura 60 Ventana para agregar nuevo elemento.....	118
Figura 61 Ventana para actualizar foto del elemento	119
Figura 62 Código para realizar consultas a la base de datos.....	120
Figura 63 <i>Ventana encargados sitio web</i>	121
Figura 64 Ventana de ingreso aplicación móvil.....	122
Figura 65 Ventana advertencia de conexión a internet.	123
Figura 66 Ventana de información sobre el aplicativo móvil.....	124
Figura 67 Código de validación de usuarios en móvil.....	125
Figura 68 Ventana de menú lateral aplicación móvil.	125
Figura 69 Ventana de menú lateral aplicación móvil.	126
Figura 71 Ventana búsquedas de inventarios aplicación móvil.....	128
Figura 72 Ventanas de búsqueda por placa aplicación móvil.....	129
Figura 73 Código de funcionamiento de lector de código de barra aplicación móvil. ..	130
Figura 74 Ventana para tomar foto en el aplicativo móvil	131
Figura 75 Ventana de opción de elementos aplicación móvil.	133
Figura 76 Ventana de añadir nuevo elemento aplicación móvil.....	134
Figura 77 Ventana de actualizar información del elemento aplicación móvil.....	135
Figura 78 Ventana de entrega de inventarios sitio web.	135
Figura 79 Ventana de chequeo sitio web.	136
Figura 80 Ventana de chequeo sitio web.	137

Figura 81 Código para generar código de barras.	137
Figura 82 Ventana trasladar elemento sitio web.	138
Figura 83. <i>Ventana de reportes sitio web.</i>	139
Figura 84 Ventana de entregas de inventario aplicación móvil.	140
Figura 85 Ventana de crear checklist aplicación móvil.	141
Figura 86 Ventana de checklist aplicación móvil.	142
Figura 87 Ventana de opción de inventario aplicación móvil.	143
Figura 88 Ventana de cambio de encargado por inventario en aplicación móvil.	144
Figura 89 Ventana de trasladar elemento de inventario aplicación móvil.	145
Figura 90 Ventana de crear solicitud sitio web.	146
Figura 91 Ventana de solicitudes pendientes sitio web.	147
Figura 92 Ventana de crear solicitudes sitio web.	148
Figura 93 Ventana de crear solicitudes sitio web.	148
Figura 94 Ventana solicitudes aplicación móvil.	149
Figura 95 Ventana crear nueva solicitudes aplicación móvil.	150
Figura 96 Ventana solicitudes realizadas usuario dependencia aplicación móvil.	151
Figura 97 Ventana solicitudes usuario administrador aplicación móvil.	151
Figura 98 Notificaciones aplicación móvil.	152
Figura 99 Estructura de archivos sitio web.	153
Figura 100 Estructura de archivos sitio web.	154

Figura 101 Estructura de archivos en el servidor.....	155
Figura 102 Rutas de navegación usuario administrador sitio web.....	156
Figura 103 Rutas de navegación usuario dependencia sitio web.....	157
Figura 104 Validación de usuario sitio web.	159
Figura 105 Test de velocidad de sitio web.....	160
Figura 106 pruebas de concurrencia sitio web.....	160
Figura 107 Reuniones por video conferencia.	163
Figura 108 Reuniones por video conferencia avances del proyecto.....	164
Figura 109 Pruebas de aceptación sitio web con Elva Triana encargada de almacén ...	165
Figura 110 Pruebas de aceptación aplicación móvil con Elva Triana encargada de almacén	166
Figura 111 Pruebas de aceptación, encuesta verificación de módulos sistema multiplataforma.	166
Figura 112 Pruebas de aceptación rol usuario dependencia, Ingeniería Ana Milena Soto Encargada de oficina de sistemas y tecnología.	168

Introducción

Las tecnologías de la información y comunicación han revolucionado el sector empresarial y se ha convertido en un reto implementar soluciones tecnológicas que permitan optimizar procedimientos y automatizar diferentes tareas en las áreas administrativas y operacionales. Una gran solución para esto ha sido la creación de los sistemas multiplataforma, los cuales facilitan el acceso a la información, brindan la capacidad de estar disponibles en cualquier dispositivo tecnológico y reducen tiempo y costos económicos para las empresas.

Para el caso de la Universidad de Cundinamarca existe un proceso relacionado a la gestión de inventarios, el cual es fundamental dentro de la institución debido a que por este se lleva el registro y seguimiento de los inventarios activos físicos que hacen parte de su patrimonio. Por tal motivo es de vital importancia para la Universidad de Cundinamarca seccional Ubaté, contar con un software que le permita la gestión de inventarios y optimice este proceso por medio de aplicación móvil y sitio web, donde se sincronice la información en tiempo real, se agilice la entrega y verificación por medio de la etiqueta de código de barras de cada uno de los elementos y se genere reportes del estado de los elementos del inventario con sus diferentes características y notificaciones sobre solicitudes de cambios en el inventario.

Teniendo en cuenta lo anterior se decide desarrollar un sistema para inventarios (Udecinv) que se encuentre disponible en dos plataformas informáticas, **aplicación para dispositivos móviles y sitio web**, las cuales estarán sincronizadas para mantener la información actualizada en cualquier dispositivo que se use, así mismo tendrán la capacidad de hacer más fácil la entrega y verificación del inventario con tan solo escanear el código de barras del elemento, validando su existencia en este. Por consiguiente, Udecinv permitirá optimizar la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté, logrando implementar fortalecer procesos

administrativos, para el desarrollo del proyecto se realiza el proceso de implementación para el programa Ingeniería de sistemas como caso de estudio y con el fin de hacer todas las pruebas necesarias dentro del sistema.

Resumen

Las entidades públicas y privadas, están siendo incentivadas de manera oportuna a la utilización e implementación de tecnologías avanzadas en sus procesos administrativos y operativos. Es por eso que la creación de sistemas digitalizados que permitan optimizar procesos y automatizar diferentes tareas, es una tendencia actual. Por lo tanto, este documento contempla la implementación de la metodología de desarrollo XP para la creación del sistema multiplataforma en sitio y aplicación móvil, donde se evidencia todo el proceso de optimización que se realiza en la gestión de inventarios, esto dado gracias al uso de la cámara como de lector de código de barras en los teléfonos móviles Android para reducir el tiempo de los checklist en las entregas de los inventarios.

De igual manera se logra una sincronización de la información entre las dos plataformas, esto facilitando la generación de reportes al finalizar el chequeo de los diferentes elementos, a su vez mediante el uso de notificaciones se puede llevar un seguimiento del estado de uso y ubicación física de los elementos que requieran intervención por parte del administrador.

Abstract

Public and private entities are being encouraged in a timely manner to use and implement advanced technologies in their administrative and operational processes. That is why the creation of digitized systems that allow to optimize processes and automate different tasks, is a current trend. Therefore, this document contemplates the implementation of the XP development methodology for the creation of the cross-platform system on site and mobile application, where the entire optimization process that is carried out in inventory management is evidenced, this

given thanks to the use of the camera as a barcode reader on Android mobile phones to reduce checklist time in inventory deliveries.

Likewise, a synchronization of the information between the two platforms is achieved, this facilitating the generation of reports at the end of the check of the different elements, In turn, the use of notifications can be used to track the state of use and physical location of items that require intervention by the administrator.

Palabras clave: Sistema multiplataforma, Código de Barras, Optimización, Aplicación móvil, Reportes, Sitio web, Android, Sincronización, Inventario físico, Checklist, Notificaciones.

Definición del problema

3.1. Descripción de la situación problemática

La Universidad de Cundinamarca seccional Ubaté cuenta con un gran inventario dividido en las diferentes carreras y dependencias administrativas, en el caso particular del programa de ingeniería de sistemas tiene asignados 465 elementos del inventario total, los cuales se encuentran distribuidos en las diferentes áreas del programa como: cubículos de profesores, oficina de investigación, oficina del programa, laboratorio de cisco, sala de arquitectura TI y laboratorio de física. Es importante aclarar que el proceso de entregas de los inventarios en la Seccional Ubaté es realizado por el funcionario encargado del almacén, el cual tiene un registro de los artículos en un archivo Excel donde se encuentra la placa del elemento, la descripción, la persona encargada, la dependencia y su precio.

El control y manejo de inventario en la Universidad de Cundinamarca actualmente se realiza mediante una etiqueta que contiene un código de barras que se usa para diferenciarlo de otros artículos por ser un código único. El procedimiento al final e inicio de cada año es el proceso de entregas donde se valida el estado y ubicación de cada uno de ellos.

Por lo tanto, para el checklist se evidencian varios problemas que hacen esta actividad complicada y se ocupe mucho tiempo, puesto que para la entrega de los elementos del inventario se realiza un proceso manual en hojas impresas con toda la información del elemento, para el cual un monitor del programa o el encargado acompañado de la persona de almacén hacen una ruta por cada inventario, verificando los elementos con su respectivo código que se encuentra en ellos.

Teniendo en cuenta la cantidad de elementos del inventario también se encontró que para el proceso de entregas de los inventarios se requiere de 1 hora y 30 minutos en promedio, gastando demasiado tiempo en este proceso, así mismo gracias a una encuesta realizada a diferentes encargados de inventarios (ver anexo 1), se puede evidenciar que se tienen una perspectiva de un 60% de favorabilidad de cómo se está llevando este proceso, esto permitiendo dar una oportunidad para aumentar este índice.

En consecuencia, se realiza un doble trabajo debido a que primero se efectúa el checklist de forma manual y después se pasa a una verificación con el archivo Excel, además en el proceso de chequeo se encuentran los artículos del inventario que tienen una etiqueta con el código de barras y debido al tiempo de uso, se ha borrado, deteriorado o perdido. Por lo tanto, se debe hacer una validación de que estos elementos se encuentren activos en el inventario.

En la dirección del programa ingeniería sistemas se ha tenido un gran inconveniente con algunos elementos que se encuentran en estado de deterioro, ya que en las listas no se tiene en cuenta ese aspecto del elemento, lo cual hace que el inventario del programa tenga artículos que ya se deben dar de baja, pero aún se encuentran asignados a pesar de ser obsoletos y no cuentan con un debido seguimiento de su estado.

3.2. Formulación del problema

¿Cómo mejorar el proceso de gestión de inventarios en la Universidad de Cundinamarca Seccional Ubaté por medio de una aplicación móvil y sistema de información caso de estudio programa Ingeniería de sistemas?

Justificación

La gestión de inventarios es de gran importancia en la Universidad de Cundinamarca seccional Ubaté, ya que hace parte de los procesos administrativos que contribuyen en la organización de los recursos físicos presentes en la seccional, por tal motivo la creación de una aplicación móvil y sitio web como plataformas de desarrollo son necesarios para que este proceso se realice de una manera óptima. Es importante destacar que el uso de herramientas tecnológicas aporta un gran sin número de beneficios para agilizar procesos cotidianos y es común ver la implementación de estas en muchas organizaciones públicas y privadas.

En este caso se beneficia, en primer lugar, el funcionario de la dependencia de almacén de la seccional Ubaté quien es la encargada de control de inventarios y en segundo lugar a los jefes de dependencia en este caso la coordinación del programa de ingeniería de sistemas quien es la encargada de los inventarios asignados como objeto de estudio. Solucionando diferentes problemáticas que se evidencian al momento de realizar la entrega de dichos inventarios. El sitio web y aplicación móvil sirve para realizar registros de elementos nuevos, verificar actuales, realizar búsquedas por sus atributos, además de poder verificar el estado, ubicación de cada uno de ellos, generar reportes y código de barras de acuerdo a la norma NTC 3842.

Puede agregarse que, al implementar este sistema multiplataforma, la seccional Ubaté tendrá varios beneficios en sus procesos administrativos, ya que se realizó una encuesta de viabilidad del proyecto donde se evidencio en un 100% la favorabilidad del proyecto por parte del personal que ha sido participe de las entregas de los elementos (ver anexo 1), así mismo el sistema aportaría una mejor eficacia en la gestión de inventarios, reducción de tiempo en las entregas de los elementos, organización y seguridad de la información, agilidad en la creación

de reportes acerca del inventario y facilidad de encontrar los elementos en sus respectivos lugares de almacenamiento.

Objetivos

5.1. Objetivo general

Desarrollar una aplicación móvil y sistema de información para la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté programa ingeniería de sistemas.

5.2. Objetivos específicos

- Identificar los requerimientos para la construcción del sistema multiplataforma, con la persona encargada de los inventarios y de almacén de la Universidad de Cundinamarca seccional Ubaté.
- Definir la arquitectura del sistema multiplataforma teniendo en cuenta la información recopilada con el cliente y el diseño de modelado de base de datos y tarjetas CRC.
- Construir el sistema multiplataforma de acuerdo a la arquitectura que se definió en un entorno web y móvil para Android que facilite el acceso a la información del inventario de la universidad de Cundinamarca seccional Ubaté.
- Evaluar los posibles errores del sistema multiplataforma mediante la implementación de pruebas del sistema y de aceptación, para evitar posibles inconvenientes en el proceso. Así mismo, documentar las fallas encontradas en el desarrollo y dar solución efectiva a estas.

Alcances y limitaciones

El sistema multiplataforma está diseñado para que todas las dependencias de la universidad de Cundinamarca seccional Ubaté pueden hacer uso de este, por medio de un usuario y contraseña que será asignada por el encargado de almacén que tiene el rol de administrador, en primera instancia Udecinv trabaja con la población objetivo del programa de ingeniería de sistemas de la universidad de Cundinamarca seccional Ubaté.

Se cuenta con dos roles, donde el primer rol es administrador y el segundo rol es para el usuario dependencia (Funcionario que tengan a su cargo elementos del inventario), hay que mencionar que estos roles facilitan el control de la información de los inventarios que se manejan en la seccional, así mismo cada rol cuenta con diferentes módulos en el sitio web y aplicativo móvil respectivamente.

El sistema está desarrollado en dos plataformas informáticas las cuales son sitio web y aplicación móvil con sistema operativo Android, debido a que los diferentes funcionarios que hacen parte del proceso de inventarios cuentan con dispositivos móviles con esta tecnología como se puede mostrar en el siguiente gráfico:

Figura 1

Estadística sobre sistema operativo en teléfonos móvil



Nota. El grafico representa el sistema operativo de los teléfonos móviles de las personas que han sido participe en el proceso de chequeo de inventarios. Fuente: Autores del proyecto

Cada plataforma tiene en sus diferentes módulos los roles mencionados anteriormente y diferentes funcionalidades distribuidos de la siguiente manera:

Sistema Multiplataforma en sitio web rol administrador: Este rol está diseñado para el funcionario encargado de almacén de la Universidad de Cundinamarca seccional Ubaté, donde tiene opciones como:

- Crear usuario y contraseña para el rol usuario dependencia.
- Entrega de los elementos del inventario: Este se realizará por medio de un lector código barras externo o un campo manual para digitar la placa.
- Búsquedas: Estas son realizadas con diferentes filtros como: Nombre del elemento, placa, lugar físico, elementos chequeados y no chequeados.
- Visualización de la imagen de cada elemento del inventario en las búsquedas.
- Opciones: Añadir, actualizar y dar de baja, cuenta con una opción de cambio de estado (Bueno, Regular, Mantenimiento y Malo), Crear código de barras para los elementos que ya no lo tienen y agregar imagen a cada elemento.
- Añadir y actualizar encargados.
- Asignación de inventarios por encargado.
- Generar Reportes: Sobre la entrega de inventarios.
- Notificaciones: Estas son generadas por solicitudes dadas por el usuario dependencia.
- Ingreso al sistema mediante manejo de sesiones seguro, con correo institucional y contraseña.

Sistema Multiplataforma en sitio web rol usuario dependencia: Este rol está diseñado para los funcionarios que tengan a cargo inventarios de las diferentes dependencias de la universidad, donde cuenta con diferentes opciones como:

- Búsquedas: Estas podrán ser realizadas con diferentes filtros como: Nombre del elemento, placa, dependencia y lugar físico.
- Visualización de la imagen de cada elemento del inventario en las búsquedas.
- Solicitud: solicitar a la persona encargada de almacén realizar un cambio al estado de algún elemento como su estado (Bueno, Regular, Mantenimiento y Malo) o actualización de lugar físico del elemento.
- Opciones: Crear código de barras para elementos que no cuenten con este.
- Ingreso al sistema mediante manejo de sesiones seguro, con correo institucional y contraseña.

Sistema Multiplataforma en aplicativo móvil rol administrador:

- Entrega de los elementos del inventario: este se realiza por medio de la cámara del dispositivo móvil, que sustituye la función de lector de código de barras y a su vez tiene un campo para digitar la placa de manera manual.
- Búsquedas: Estas pueden ser realizadas con diferentes filtros como: Nombre del elemento, placa, inventario, lugar físico, elementos chequeados y no chequeados.
- Opciones: Añadir, Actualizar y dar de baja, cambio de estado (Bueno, Regular, Mantenimiento y Malo).
- Notificaciones: Estas son generadas por solicitudes dadas por el usuario dependencia.

- Ingreso a la App mediante manejo de sesiones seguro, con correo institucional y contraseña.

Sistema Multiplataforma en aplicativo móvil rol usuario dependencia:

- Búsquedas: Estas pueden ser realizadas con diferentes filtros como: Nombre del elemento, placa, inventario y lugar físico.
- Solicitud: Solicitar a la persona encargada realizar un cambio al estado de un elemento a (Bueno, Regular, Mantenimiento y Malo), Actualización de lugar físico del elemento.
- Ingreso a la App mediante manejo de sesiones seguro, con usuario y contraseña.

El sistema multiplataforma cuenta con sincronización en tiempo real, evidenciando las distintas actualizaciones que se realizan, ya sea en el aplicativo móvil o en el sitio web y en cada uno de los usuarios (roles) que lo estén usando.

Para el estudio de este proyecto se tuvo en cuenta la línea de investigación que se fundamenta con base a las líneas que proporciona la facultad de ingeniería de la universidad de Cundinamarca, la cual es software, sistemas emergentes y nuevas tecnologías. Además de lo anterior se busca reducir el impacto ambiental en una forma específica la cual es reducir el uso de papel en el desarrollo del control de inventarios, digitalizando cada una de estas actividades.

Marco de referencia

Este marco de referencia está compuesto por estado del arte, teorías y conceptos que son base fundamental para la investigación del proyecto de grado, de igual forma buscan dar un enfoque y profundizan acerca de la temática de gestión de inventarios y sistemas multiplataforma los cuales están integrados para dar una solución efectiva a la problemática planteada anteriormente.

7.1. Estado del arte

En la actualidad hay desarrollos que tienen como funcionalidad la gestión de inventarios e involucran diferentes herramientas que hacen que las empresas tengan mayor control de sus procesos administrativos en los cuales se encuentra integrado la gestión de inventarios. Para que un proyecto sea más viable en un entorno tecnológico se debe hacer un análisis de proyectos similares que puedan tener un enfoque similar respecto a la temática elegida. Es por esto que las siguientes referencias se han tenido en cuenta para el proceso de análisis de la información, donde servirán para poder tomar decisiones acerca de las funcionalidades y plus que se le puedan agregar a el desarrollo del proyecto con el objetivo de que este sobresalga de los demás.

Reseña del software disponible en Colombia para la gestión de inventarios en cadenas de abastecimiento: (Gutiérrez & Patricia Jaramillo, 2009). La gestión de los diferentes tipos de inventario a lo largo de una cadena de abastecimiento es un problema complejo que, en la mayoría de los casos de la industria colombiana e internacional, se aborda mediante la

implementación de herramientas de software. Sin embargo, la decisión de implementar dichas herramientas está en muchos casos limitada por la capacidad financiera para invertir en ellas y por el desconocimiento de la oferta de software disponible en Colombia. Este artículo presenta una reseña del software disponible en Colombia para la gestión de inventarios con el fin de brindar a las empresas que se enfrentan a la decisión de implementar un software, una herramienta de soporte que indique cuál es la oferta en Colombia y cuáles son las características de los sistemas disponibles. Para esto se presenta primero el estado del arte, en el cual se identifica el resultado de la revisión de la literatura y el uso de las herramientas de software en la industria colombiana. Seguidamente se presenta la caracterización de los productos de software disponibles en el país. Dicha caracterización permite identificar la configuración general de los productos, los módulos de gestión con los que cuenta, las opciones de integración con otras herramientas informáticas, y la información general de precios e instalación. Finalmente se presentan las conclusiones del estudio y las futuras oportunidades de investigación.

Desarrollo del Sistema de Información de Gestión de Inventario en BPRTIK

KEMKOMINFO Jakarta: (Fetrina et al., 2017). El Instituto de Capacitación e Investigación para Tecnología de la Información y la Comunicación (BPRTIK) es una institución dependiente del Ministerio de Comunicaciones y Tecnología de la Información (KEMKOMINFO). Dado que esta institución administra sus inventarios mediante el uso de una hoja de cálculo para que los datos no estén sincronizados correctamente y sean propensos a la duplicación de datos. Los informes de inventario, como los informes del proceso de mantenimiento, también se realizan manualmente y se registran en documentos que no se han organizado en una sola base de datos, lo que hace que esos informes sean vulnerables a la pérdida o corrupción de datos. Además, el

proceso de asignación y monitoreo de la tarea todavía se realiza manualmente mediante el uso de una nota o incluso verbalmente, lo que luego lleva a los informes indocumentados. En este estudio, los datos fueron recolectados por entrevista, observación y estudio de literatura. El desarrollo rápido de aplicaciones (RAD) y el enfoque orientado a objetos utilizando el lenguaje de modelado unificado (UML) se utilizaron como métodos de diseño y desarrollo del sistema, respectivamente. Los resultados de este estudio son el sistema de información de gestión de inventario, que puede respaldar y gestionar los procesos del inventario, como el proceso de control y monitoreo, mantenimiento, asignación e informes.

Mystock Inventory Manager: (Mystock, 2018). Es una aplicación diseñada específicamente para usuarios de dispositivos móviles y tabletas. Sirve como una herramienta poderosa para que las personas, las pequeñas y medianas empresas registren y administren los movimientos de existencias y puedan calcular los costos fácilmente con una inversión mínima. Esta aplicación también se puede utilizar para organizaciones más grandes para la gestión de inventario de múltiples ubicaciones utilizando un solo dispositivo. Además, esta aplicación es ideal para monitorear el inventario de ubicaciones remotas e intercambiar datos entre dispositivos móviles y servidores de datos. Aunque es fácil de usar, esta aplicación rica en funciones con una interfaz visual agradable puede ser la mejor solución móvil para satisfacer todos sus requisitos de inventario.

Sistema de Control de Inventario en Entorno Web y Dispositivos Móviles con Sistema Operativo Android para la Empresa Rodamientos Bower: (Vázquez, 2016). Este proyecto tiene como objetivo principal mejorar la forma de cómo se maneja en inventario dentro de la

empresa Rodamientos Bower, mediante el uso de lenguajes de programación libres y para dispositivos móviles, así como el diseño y desarrollo de una página web se busca brindar las herramientas necesarias para que la empresa pueda controlar todos los ítems que se encuentran en su bodega de forma rápida y manteniendo el mayor control posible. No solo es un proyecto de desarrollo de habilidades tecnológicas sino sirve para un desarrollo de habilidades de negociación y comunicación entre el desarrollador y el cliente (usuario final del sistema), así como también para mejorar la forma como se emplean metodologías de desarrollo y arquitecturas de software. Todo lo mencionado anteriormente hace de este proyecto una gran experiencia, muy enriquecedora tanto para las personas encargadas de desarrollarlo como cada uno de los implicados de la empresa.

Sistema de Control de Inventario Mediante Códigos QR para el Grupo de Investigación Teletecno: (Lilia, 2017). En el siguiente artículo se describe el desarrollo e implementación de un sistema de control de inventario mediante códigos QR, el cual será empleado por el grupo de investigación TELETECNO de la Universidad Distrital Francisco José de Caldas en la Facultad Tecnológica; con el fin de facilitar el acceso a la información de los instrumentos que se encuentran allí, de esta manera los estudiantes podrán hacer uso de ellos, ya sea en un trabajo de investigación o en el desarrollo de un proyecto.

Invetdroid: App para Realización de Inventarios en Dispositivos Android: (Vázquez Rubén Ordóñez, 2018). App destinada a pequeñas y medianas empresas con el propósito de poder realizar inventarios, destacando por la posibilidad de realizar capturas de códigos de barras mediante la cámara del dispositivo móvil. Los datos de las capturas se almacenan en un fichero

con formato XLS el cual puede ser enviado por email. Recordar que el formato XLS es utilizado por la mayoría de empresas del mundo con lo que asegura la practicidad a múltiples.

Implementada bajo el sistema operativo Android por su alta cuota de mercado y bajo coste de desarrollo e implementación. Disminuyendo la inversión de la empresa para acometer inventarios.

Un Modelo de Gestión de Inventario en Comercio Móvil: (Zheng & Meng, 2010). En el entorno de comercio móvil, la gestión de inventario evoluciona a gestión de inventario móvil. Este último es diferente de la gestión de inventario tradicional en muchos aspectos. En este documento, se desarrolla un modelo práctico integrado para mejorar la gestión de inventario en el comercio móvil. Los puntos clave para desarrollar este modelo práctico de gestión de inventario incluyen determinar la cantidad de puntos de pedido, determinar la cantidad de pedido económico y determinar el stock de seguridad. En la última sección, se examinó la aplicabilidad de este modelo, lo que demostró que este modelo hizo que la gestión de inventario móvil sea una rutina.

Sistema de Información de Control de Inventarios del Almacén del ITS: (López et al., 2011). Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto, proyecto está dirigido a hacer mejoras en el Instituto Tecnológico de Saltillo (ITS), en el Departamento de Recursos Materiales y Servicios, en el área de Almacén. En base a las Tecnologías de Información se desarrolló un sistema de información, el cual se encuentra en la etapa de implementación en un proceso de pruebas y correcciones, como un soporte de apoyo administrativo que permita llevar el control de inventarios del almacén del ITS con la finalidad

de ofrecer rapidez y seguridad en el manejo del inventario. Actualmente se trabaja de forma manual con formatos en Word y Excel, es necesario emigrar a otros entornos que permitan automatizar las tareas propias del almacén. Dentro de los resultados que se propondrán del proyecto, es obtener consultas e informes requeridos por los usuarios, un seguimiento y control de los materiales que entran y salen del almacén a los diferentes departamentos del ITS. Lo anterior permitirá a los responsables de la institución establecer los mecanismos requeridos para dar un seguimiento adecuado y cumplir con los objetivos del programa de gestión de calidad bajo la normativa de ISO 9001:2000.

Sistema de Inventario con Código de Barras: (Flores, 2018). A continuación, se explicara sobre el contenido bajo una estructura como lo es planteamiento del problema, objetivos, hipótesis, objeto del estudio, metodología, desarrollo, resultado y conclusiones, toda esta estructura tendrá que ver con un sistema de inventario con código de barras que ayudará a la empresa INIFAP a llevar un mejor control ya que dicha empresa realiza el inventario de manera manual y es por ello que se decidió realizar este proyecto para la actualización de tecnología en la empresa. Se mencionará como es que se fue desarrollando el proyecto para alcanzar el resultado deseado además que se den recomendaciones para que en un futuro esto se pueda mejorar.

Desarrollo e Implementación de un Sistema de Control de Inventario de Equipos de Audio y Video para el Canal de Televisión de la Universidad Católica de Santiago de Guayaquil: (Rivera Guevara & Gabriela Estefanía, 2018). El canal de televisión de la Universidad Católica de Santiago de Guayaquil dispone de activos de audio y video cuya

adecuada gestión es fundamental para el correcto desempeño de sus actividades. La gestión del inventario y el control de los equipos y de su disponibilidad representa por ende una importante tarea para el personal encargado de este. El presente proyecto de titulación busca el desarrollo e implementación de una aplicación web que le permita al UCSGTV llevar un control eficiente del inventario de sus equipos de audio y video y que logre optimizar los procesos en este involucrados. Con tal fin, se utilizó un enfoque de investigación de tipo cualitativo, realizando entrevistas al personal del canal, que permitieron el levantamiento de información relevante sobre los procesos que intervienen en la gestión del inventario y los requerimientos de la aplicación. Una vez implementado el sistema, se logró optimizar la entrada y salida de equipos de audio y video, generar reportes de estado, solicitudes de egreso, controlar el stock del inventario y mantener un registro detallado del movimiento de estos activos.

Análisis, Diseño e Implementación de Un Sistema de Gestión de Inventarios de Activos de Ti Para los Laboratorios de la Carrera de Ingeniería en Ciencias de la Computación de la Universidad Politécnica Salesiana Sede Quito Campus Sur: (Oña L., 2013). El tema de investigación se desarrolló en el Hospital Teodoro Maldonado Carbo- IESS Guayaquil, con el objeto de conocer la realidad en la que se encuentra en el manejo de los desechos hospitalarios infecciosos (segregación, manipulación, transporte y almacenamiento final). Para cumplir con el mismo se empleó entrevistas, encuestas, observaciones directas y revisión de documentos. Durante el estudio de campo se identificaron las falencias de los procedimientos de segregación, transporte, manipulación y tratamiento de residuos hospitalarios; además de analizar el comportamiento del capital humano que interviene en la gestión de los desechos. En base a los resultados se propone un Plan de Mejora que le permita al Centro de

Salud tener herramientas para optimizar su desempeño en la gestión de manejo de desechos infecciosos y bioseguridad y a la vez cumplir con la Auditoría que realizará el Ministerio de Salud Pública del Ecuador para extender una acreditación que le permitirá tramitar su permiso de funcionamiento.

Los proyectos referenciados tienen enfoques similares a el sistema multiplataforma propuesto, la gran diferencia respecto a este proyecto es que ninguno está desarrollado en dos plataformas como son web y móvil, permitiendo darle una mayor ventaja al software que se quiere desarrollar porque aumentaría su capacidad de uso, gracias a su fácil accesibilidad que puede ser desde un dispositivo móvil o una computadora y a su innovación. Ahora bien, los sistemas anteriores son proyectos que no han sido implementados en Colombia, por consiguiente, se abriría una gran oportunidad para implementarlo en la Universidad de Cundinamarca seccional Ubaté.

También cabe comparar que estos sistemas implementaron metodologías de desarrollo que usan fases similares a las de la metodología xp, estas fueron: modelo incremental, scrum y modelo en cascada. De modo que son fundamentales para el proceso de desarrollo con la metodología elegida, porque se pueden aplicar diferentes conceptos teóricos y mejorar los desaciertos que ellos pudieron tener, para que se puedan evitar en la ejecución del proyecto.

7.2. Marco teórico

Para la creación del sistema de información web y aplicativo Udecinv se tienen en cuenta diferentes teorías acerca de inventarios, sistemas móviles y multiplataforma e implementación de código de barras, los cuales tienen gran importancia para el desarrollo del sistema.

Teoría de Inventarios

(Cátedra: Producción I Capítulo 9. Teoría de Inventarios., 2008.). Los inventarios prevalecen en el mundo de los negocios. Mantener inventarios es necesario para las compañías que tratan con productos físicos, como fabricantes, distribuidores y comerciantes. Por ejemplo, los fabricantes necesitan inventarios de materiales requeridos para la manufactura de productos. También deben almacenar productos terminados en espera de ser enviados. De manera similar, tanto los distribuidores como las tiendas deben mantener inventarios de bienes disponibles cuando los consumidores los necesiten. Reducir los costos de almacenamiento evitando inventarios innecesariamente grandes puede mejorar la competitividad de cualquier empresa.

Esta teoría es muy importante ya que contempla cómo funcionan los inventarios a nivel de empresas donde cabe recalcar que el inventario de la universidad de Cundinamarca funciona de una manera similar donde se deben tener en cuenta los elementos que ya no hacen parte de este inventario con el fin de que el sistema siempre este actualizado y listo para cualquier modificación que se requiera en el inventario físico de tal forma que toda la información sea verídica.

Origen del Código de Barras

(Códigos de barras, 2018). La necesidad de clasificar, inventariar y codificar las mercaderías existe desde que existe el comercio. Los artesanos ponían sus marcas particulares en la mercancía, y se llevaban inventarios y registros de compra-venta. Todos los sistemas de codificación evolucionaron a través del tiempo, y en la actualidad se utiliza el código de barras, muy usado en autoservicios y tiendas similares. El código de barras utilizado en los autoservicios es un estándar internacional y único, además de que no es exclusivo de un solo sector. De manera simplificada, el código de barras es una serie de líneas paralelas y espacios de diferente grosor; el ancho de las líneas y de los espacios determina el dato codificado en el código. El código de barras no contiene información, sólo identifica el producto. El código de barras es una herramienta que facilita el proceso de gestión de inventarios ya que este almacena la información unitaria de cada elemento y permite acceder a la información más rápidamente.

Importancia de las Apps en las Organizaciones

(La importancia de las apps móviles en las empresas - Economía TIC, n.d.). En la última década, las apps móviles se han posicionado como unas de las herramientas más eficaces para las empresas, especialmente en el campo del e-commerce. Y no es para menos, pues las aplicaciones facilitan la vida de los usuarios, promueven la interconectividad y mejoran la experiencia en la adquisición de productos y servicios. Según un estudio realizado por la compañía Flurry Analytics, el usuario promedio pasa alrededor de 5 horas diarias en su teléfono y más de la mitad de este tiempo lo dedica a las apps. En el año 2017 los ingresos a través de apps generaron un total de 76 mil millones de dólares. En definitiva, el uso de las apps cambió la dinámica del mercado para siempre. Por ello, la implementación de estos sistemas en

los modelos de negocio es un pilar fundamental para el crecimiento de toda empresa. Esta teoría profundiza en la importancia que están teniendo las aplicaciones móviles en el mundo, lo cual hace que este proyecto aumente su viabilidad y pueda ser de mayor uso en el proceso de gestión de inventarios.

Sistemas Multiplataforma

(Desarrollo de aplicaciones multiplataforma: web-apps vs. nativas, 2018). En pleno siglo XXI vivimos rodeados de dispositivos, y éstos viven de las aplicaciones de las que disponga su sistema operativo. Si hace unas décadas se solía decir “no existes hasta que sales por la tele”, hoy en día esa presencia obligada se refiere a los dispositivos móviles. Cualquier organización debe tener su aplicación dedicada. Pese a que la miríada de dispositivos se ha reducido en los últimos tiempos (tras la caída de Symbian, Windows Phone, BlackBerry, Firefox OS, Tizen...), la necesidad de estar presente en los que siguen vigentes es más acuciante que nunca. Android, Kindle Fire, iOS, Xbox, Nintendo Switch, PlayStation, tablets... Cada plataforma es un mundo y los rápidamente cambiantes sistemas operativos conforman un ecosistema digital de lo más enrevesado. De ahí surge como requisito el desarrollo de aplicaciones multiplataforma. Finalmente, los sistemas multiplataforma son muy importantes en la actualidad porque brindan muchas ventajas de acceso a la información y ahorro de tiempo y costos por eso este sistema de inventarios integra estas ventajas gracias a su desarrollo multiplataforma.

7.3. Marco conceptual

Este marco contempla las definiciones e información relacionada con la temática seleccionada para el desarrollo del proyecto, permitiendo contextualizar la información implementada con el fin de que esta sea de fácil comprensión y entendible para lograr una comprensión más adecuada de los resultados.

Aplicación Móvil

Las aplicaciones móviles son programas diseñados para ser ejecutados en teléfonos, tabletas y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades. (“Definición y cómo funcionan las aplicaciones móviles,” 2020.). Estas permiten a su vez facilitar el acceso a la información de gestión de inventarios por su capacidad de adaptarse a cualquier dispositivo móvil y permitiendo a los usuarios utilizar sus periféricos para realizar la entrega de los inventarios por medio de la cámara.

Código de Barras

Es un conjunto de líneas paralelas verticales de distintos anchos y espacios que sirve para codificar y comparar información de un artículo, que al ser leído por un escáner proporciona datos relacionados con las características del producto como nombre, tipo, ubicación, precio, entre otros. (“¿Qué es el Código de Barras? – Visión LOGYCA,” n.d.). El código de barras es una parte fundamental en el sistema multiplataforma porque es el que permite realizar la entrega

de los inventarios de una manera más ágil, ahorrando tiempo de la recopilación de las placas de los elementos.

Metodología XP

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos. Extreme Programming se centra en potenciar las relaciones interpersonales del equipo de desarrollo como clave del éxito mediante el trabajo en equipo, el aprendizaje continuo y el buen clima de trabajo. Esta metodología pone el énfasis en la retroalimentación continua entre cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes. (“Metodología XP Programación Extrema (Metodología ágil) - Diego Calvo,” n.d.). Esta metodología se adapta a el objetivo que se quiere realizar con el aplicativo multiplataforma, ya que permite tener una comunicación con el cliente más seguido y los resultados del proyecto se pueden reflejar más rápido por ser una metodología ágil.

Inventario

El inventario representa la existencia de bienes almacenados destinados a realizar una operación, sea de compra, alquiler, venta, uso o transformación. Debe aparecer, contablemente, dentro del activo como un activo circulante. (“¿Qué es inventario? Tipos, utilidad, contabilización y valuación - GestioPolis,” n.d.). Tener en cuenta un concepto tan importante como este es fundamental para que la ejecución del proyecto siempre está basada en cómo funcionan los inventarios.

Base de Datos

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrarla y utilizarla fácilmente. (“¿Qué son las bases de datos?” n.d.). Estas permiten también la comunicación entre diferentes plataformas donde hacen el rol de servidor y estas plataformas el cliente, teniendo siempre la información organizada y disponible para estos puedan hacer las consultas necesarias.

Multiplataforma

Multiplataforma es la capacidad de un software de soportar múltiples plataformas informáticas. Esto significa que el software que es multiplataforma tiene la característica de funcionar de forma similar en distintas plataformas. (“Definición de Multiplataforma (informática),” n.d.). La gran ventaja de los sistemas multiplataforma son su capacidad de sincronizar la información en los diferentes sistemas operativos donde se estén usando, dando la capacidad de que para la gestión de inventarios se faciliten muchos procesos y funcionalidades.

TIC

Las tecnologías de información y comunicación, mayormente conocidas como “TIC”, son aquellas cuya base se centra en los campos de la informática, la microelectrónica y las telecomunicaciones, para dar paso a la creación de nuevas formas de comunicación. Se trata de un conjunto de herramientas o recursos de tipo tecnológico y comunicacional, que sirven para facilitar la emisión, acceso y tratamiento de la información mediante códigos variados que pueden corresponder a textos, imágenes, sonidos, entre otros. (“¿Qué son las Tics o Tecnologías

de la Información y la Comunicación?,” n.d.). Las TIC actualmente están presentes en cualquier proceso donde se implemente la tecnología, al respecto conviene decir que donde se realice la gestión de inventarios se implementa estas en su sistema.

Aplicación Web

Una aplicación web es una aplicación o herramienta informática accesible desde cualquier navegador, bien sea a través de internet (lo habitual) o bien a través de una red local. A través del navegador se puede acceder a toda la funcionalidad y tener cualquiera de las soluciones enumeradas en el punto anterior. (“¿Qué es una aplicación Web? – Blog Neosoft Sistemas,” n.d.). Las aplicaciones web también son sistemas que se renderizan en un navegador y permiten su acceso a la información desde cualquier servidor web que tenga conexión a internet, ofreciendo ventajas para los usuarios que se encuentren a gran distancia, brindándoles la capacidad de interactuar con los datos e información ingresada a el sistema, de acuerdo a lo anterior se garantiza que tanto administrador como usuarios de las diferentes dependencias puedan llevar un control más cercano de los inventarios.

Plataformas de Desarrollo

Una plataforma de desarrollo de código bajo o sin código es un entorno de desarrollo visual integrado que permite a los desarrolladores ciudadanos arrastrar y soltar componentes de aplicaciones, conectarlos y crear una aplicación móvil o web. Este enfoque modular permite a los desarrolladores profesionales crear aplicaciones más rápidamente al liberarlos de la necesidad de escribir línea por línea de código. Las plataformas de código bajo y sin código también permiten a los analistas de negocios, administradores de oficinas, propietarios de pequeñas empresas y

otras personas que no son desarrolladores de software crear y probar aplicaciones, porque liberan a los creadores de aplicaciones de tener que saber algo sobre lenguajes de programación tradicionales, código de máquinas o el trabajo de desarrollo que implica el construir los componentes configurables de la plataforma. (“¿Qué es ¿Qué son las plataformas de desarrollo de código bajo y sin código?? - Definición en WhatIs.com,”).

7.4. Marco legal

Este proyecto se fundamenta en lo establecido en las normas y leyes propuestas por el estado que garantizan la protección de la información de los usuarios que brindan información privada y datos relevantes los cuales hacen parte del sistema, una de estas es la Ley Estatutaria 1581 de 2012, para los temas y disposiciones generales de protección de datos personales. La Ley de Protección de Datos abarca el derecho que tienen los ciudadanos a conocer, actualizar y rectificar información recopilada sobre ellos en bases de datos o archivos registrados por entidades públicas o privadas. Es necesario recalcar que su cobertura se limita al estado colombiano o territorios extranjeros siempre y cuando se posean acuerdos o tratados de carácter internacional que dictaminen su validez; sin embargo, no aplicará principalmente en: bases de datos o archivos usados en un ámbito personal o doméstico, bases de datos con información periodística y bases de datos para asuntos de seguridad y defensa. (“Normograma del Ministerio de Tecnologías de la Información y las Comunicaciones. (LEY_1581_2012, 2020).

En consecuencia, el sistema multiplataforma deberá brindar seguridad a los datos que se registren en la base de datos de este, teniendo en cuenta que esta información solo será para uso de la persona que utilice el aplicativo con el fin de mantener la privacidad de sus datos, en efecto por ningún motivo podrá ser publicada a otras plataformas en línea o medios de transferencia de información.

Es importante también destacar en este apartado la protección de los derechos de autor del código y los diferentes contenidos que hacen parte del sistema multiplataforma, los cuales fueron realizados por los creadores del proyecto, por lo tanto, se toman como referencia las leyes colombianas que respaldan los derechos de autor. Una de estas leyes es la ley 23 del 28 de enero

de 1982, la cual es la principal reguladora de derechos de autor en Colombia y estipula la protección de obras literarias, artísticas y científicas, También protege esta ley a los intérpretes o ejecutantes, a los productores de fonogramas y a los organismos de radiodifusión, en sus derechos conexos a los del autor.(*Ley 23 de 1982 Nivel Nacional*, Artículo 1.).

Ley 565 del 2000 donde se establece el tratado de la OMPI -Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor, adoptado en Ginebra, el 20 de diciembre de mil 1996 y el cual en su artículo 4 declara lo siguiente: “Los programas de ordenador están protegidos como obras literarias en el marco de lo dispuesto en el Artículo 2 del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión.” (*C-1183-00 Corte Constitucional de Colombia*, pag. 2).

Ley 48 de 1975, por medio de la cual se autoriza la adhesión de Colombia a la convención universal sobre derecho de autor y se aprueba la convención Internacional sobre la Protección de los Artistas Intérpretes o Ejecutantes, los Productores de Fonogramas y los Organismos de Radiodifusión, hecha en Roma el 26 de octubre de 1961. Donde se establece que los Estados contratantes, animados por el deseo de asegurar en todos los países la protección del derecho de autor sobre las obras literarias, científicas y artísticas, convencidos de que un régimen de protección de los derechos de autor adecuado a todas las naciones y formulado en una convención universal, que se una a los sistemas internacionales vigentes sin afectarlos, contribuirá a asegurar el respeto de los derechos de la personalidad humana y a favorecer el desarrollo de las letras, las ciencias y las artes, persuadidos de que tal régimen universal de protección de los derechos de los autores facilitará la difusión de las obras del espíritu y una mejor comprensión internacional. (Ley 48 de 1975, Pág. 1.).

Marco metodológico

Para la realización del sistema multiplataforma se implementa la investigación aplicada, ya que se aplica y usan todos los conocimientos adquiridos durante la carrera de pregrado ingeniería de sistemas, así mismo esta investigación brinda la solución de problemáticas mediante teorías que han sido investigadas para posteriormente llevarlas a la práctica.

De igual forma se tendrán en cuenta diferentes fases que van articuladas con la metodología de desarrollo que se está aplicando, cumpliendo con cada uno de los objetivos propuestos anteriormente. Se verifica que cada uno de los procesos se cumpla de acuerdo a las fechas establecidas en el cronograma para que se pueda dar como resultado un proyecto de software que solucione la problemática actual, para esto se tienen en cuenta los siguientes pasos:

En primer lugar, se realiza el levantamiento de los requerimientos con la persona encargada de almacén y de inventarios de la Universidad de Cundinamarca seccional Ubaté, para tener claro cómo funciona el proceso de inventarios actualmente y establecer la problemática y sus posibles soluciones tecnológicas para posteriormente aplicar las siguientes fases.

Luego se analizó esta información para definir una arquitectura que será base para el desarrollo del sistema multiplataforma, contando con las historias de usuario, clases UML, el modelado de la base de datos y los diferentes diagramas de estado, esto con el fin de preparar toda la estructura de una manera adecuada para un desarrollo eficaz, además de tener en cuenta la interfaz que se implementa en las diferentes plataformas informáticas.

El siguiente proceso es el desarrollo del sistema multiplataforma para web y móvil que permita gestionar el manejo de inventarios de la universidad de Cundinamarca seccional Ubaté, implementando las diferentes herramientas para el desarrollo, el cual está articulado por tres

partes importantes las cuales son: el diseño, la lógica y la base de datos, cada una de estas con un respectivo lenguaje o motor de bases de datos y entornos que permitan facilitar su desarrollo.

Finalmente, se evalúan los posibles errores que el sistema tenga después del desarrollo, para tener como resultado final un software listo para ser usado por el usuario encargado del almacén de la Universidad de Cundinamarca seccional Ubaté y la coordinación del programa ingeniería de sistemas como rol usuario dependencia.

8.1. Aspectos metodológicos de desarrollo

La metodología de desarrollo que se está implementando en la creación de este proyecto es la metodología XP (Programación Extrema), la cual proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Además, esta metodología es aplicable con resultados positivos a proyectos de mediana y pequeña envergadura, donde los grupos de trabajo no superan 20 personas. (CyTA, 2006.).

Esta metodología cuenta con 4 valores que son fundamentales para crear un buen ambiente de trabajo entre los involucrados en el proyecto para cumplir con las expectativas del cliente.

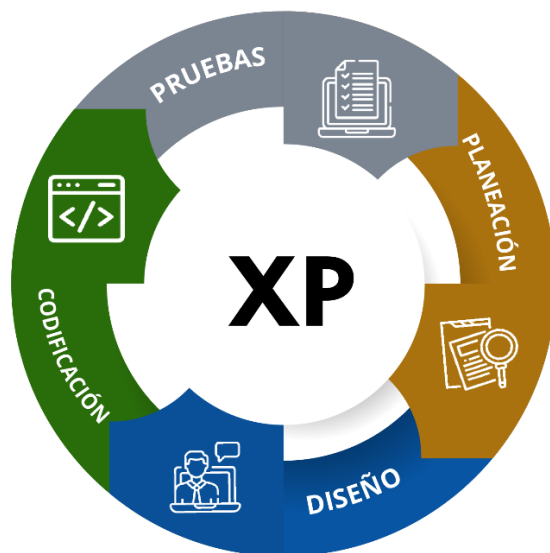
- La comunicación. Es muy importante que exista un ambiente de colaboración y comunicación al interior del equipo de desarrollo, así como en la interacción de éste con el cliente. En XP la interacción con el cliente es tan estrecha, que es considerado parte del equipo de desarrollo.
- La simplicidad. Este valor se aplica en todos los aspectos de la programación extrema. Desde diseños muy sencillos donde lo más relevante es la funcionalidad necesaria que requiere el cliente, hasta la simplificación del código mediante la refactorización del mismo.

- La retroalimentación. Se presenta desde el comienzo del proyecto, ayuda a encaminarlo y darle forma. Ésta se presenta en los dos sentidos, por parte del equipo de trabajo hacia el cliente, con el fin de brindarle información sobre la evolución del sistema, y desde el cliente hacia el equipo en los aportes a la construcción del proyecto.
- El coraje. El equipo de desarrollo debe estar preparado para enfrentarse a los continuos cambios que se presentarán en el transcurso de la actividad. Cada integrante debe tener el valor de exponer los problemas o dudas que halle en la realización del proyecto. Aún con estas variaciones, las jornadas de trabajo deben proporcionar el máximo rendimiento. (LUIS MIGUEL & ELENA, 2007).

La metodología implementada se descompone en cuatro fases secuenciales:

Figura 2

Ciclo de vida de metodología XP.



Fuente: Autores del proyecto, basado en Metodología XP.

- Planeación: En este punto se empieza a interactuar con la persona encargada del almacén de la universidad de Cundinamarca seccional Ubaté y los integrantes del proyecto para

identificar los requerimientos del sistema, Al mismo tiempo se establecen la cantidad y duración de cada iteración y se concretan las historias de usuario, el plan de entregas y las reuniones continuas, de esta manera se cumple con la primera fase de la metodología de desarrollo. (ver anexo 3).

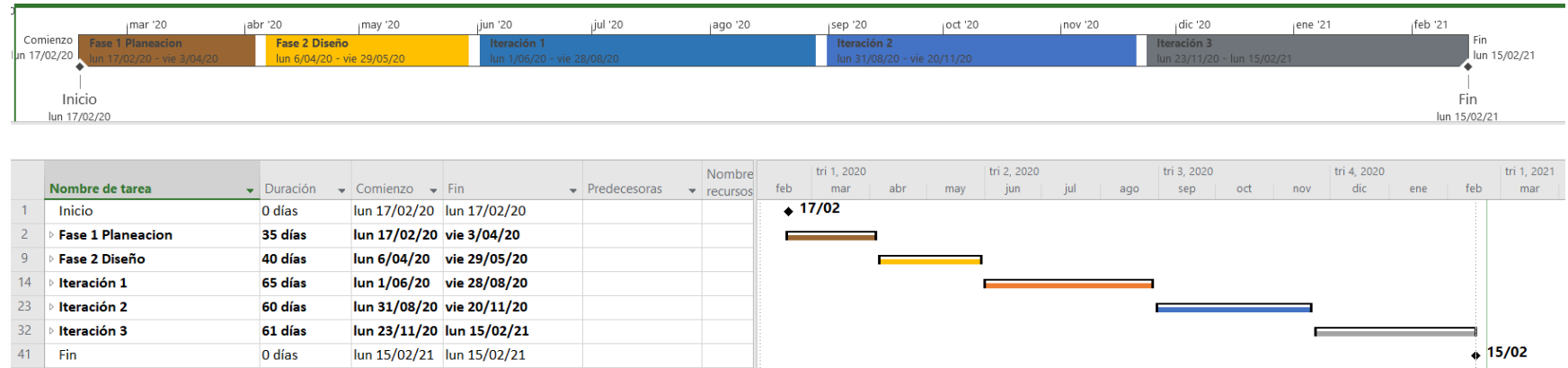
- **Diseño:** Toda la información recopilada en la fase anterior es sometida a un proceso de análisis y se estudian las funcionalidades a realizar respecto a la gestión de inventarios, a su vez se realizan diseños de los modelos los cuales son: casos de uso, diagramas de UML y modelos relacionales. Para cumplir con las historias de usuario asignadas para esta fase.
- **Codificación:** Es un proceso que se realiza de forma paralela con el diseño y el cual está sujeto a varias observaciones, se realizan una serie de pruebas unitarias a cada una de las historias que se incluirán en la entrega. Esta fase se realiza la programación en parejas con el objeto de crear un código para una historia de usuario, el desarrollo será realizado de forma simultánea para dispositivos móviles y entorno web, con el objetivo de poder llevar una similitud en cuanto a el diseño y funcionalidad en ambas plataformas, evitando consecuencias negativas en la sincronización de la información.
- **Pruebas:** XP enfatiza mucho los aspectos relacionados con las pruebas, clasificándolas en diferentes tipos y funcionalidades específicas, indicando quién, cuándo y cómo deben ser implementadas y ejecutadas. Además, se realizan a el proyecto pruebas unitarias, de aceptación y cuando se encuentra algún error, puede agregarse que de igual forma que en la codificación estas pruebas serán realizadas en paralelo en ambas plataformas, verificando que la información y los procesos estén funcionando de igual forma y de la manera adecuada. (LUIS MIGUEL & ELENA, 2007). (ver anexo 2).

Cronograma

El cronograma que se realizó para el sistema multiplataforma, está basado en la metodología XP y cada una de sus 4 fases que se implementaron durante el desarrollo del proyecto, donde se especifican las tareas a realizar en cada una de estas fases con sus fechas respectivas teniendo en cuenta las iteraciones que se requieren en la metodología.

Figura 3

Cronograma de actividades sistema multiplataforma.

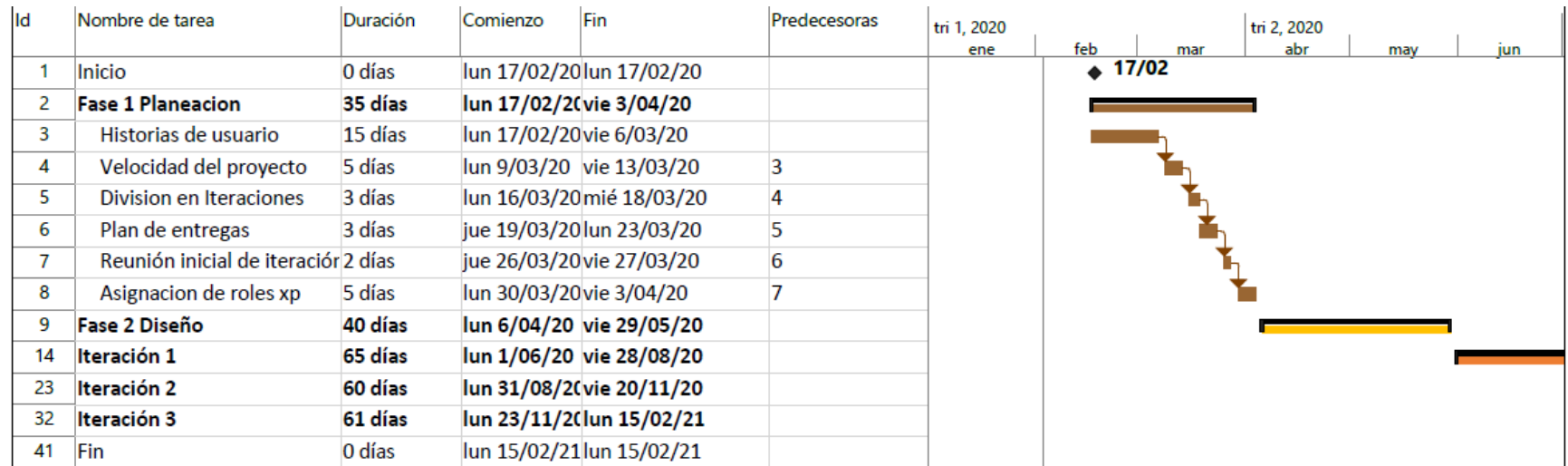


Nota: El cronograma se realizó en base a la metodología XP. Fuente: Autores del proyecto.

Figura 4

Cronograma de actividades sistema multiplataforma Fase 1.

Fase 1 planeación

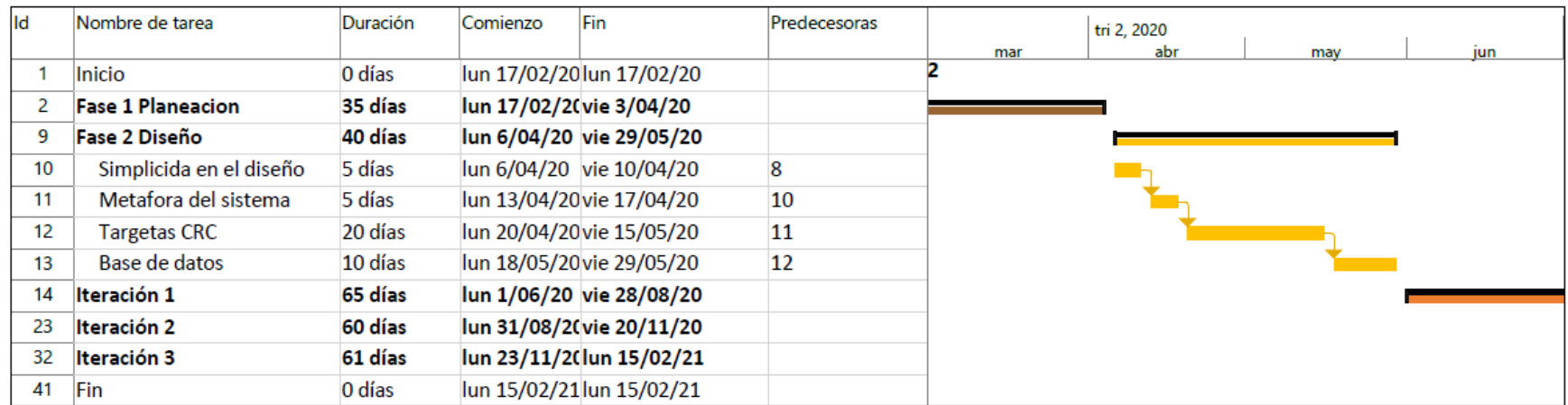


Fuente: Autores del proyecto.

Figura 5

Cronograma de actividades sistema multiplataforma Fase 2.

Fase 2 Diseño



Fuente: Autores del proyecto.

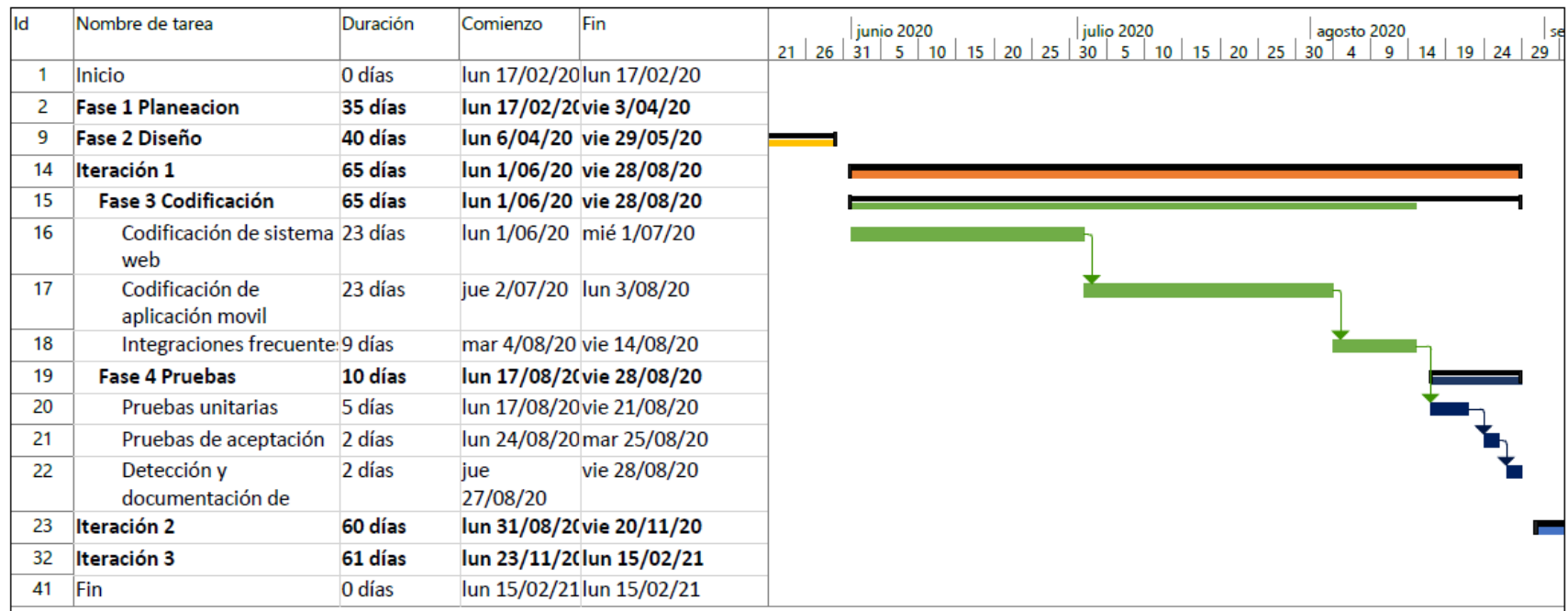
Figura 6

Cronograma de actividades sistema multiplataforma iteración 1.

Iteración 1

Fase 3 codificación

Fase 4 Pruebas



Fuente: Autores del proyecto.

Ingeniería del proyecto

10.1. Planeación

Esta fase de la metodología XP contempla la etapa inicial del proyecto, donde se empieza a interactuar con el cliente y todo el grupo de desarrolladores con el fin de dar solución a la problemática propuesta. Así mismo identificar los requerimientos del sistema, diseño de las historias de usuario, planes de entrega, asignación de roles y ajustes del proyecto. Esto permitirá establecer las herramientas y procedimientos que serán implementados durante la ejecución del proyecto.

Se recopiló toda la información necesaria para la realización del proyecto por medio de un levantamiento de requerimientos con el cliente donde se pudieron clasificar estos por: requerimientos funcionales y no funcionales que son:

10.1.1. Requerimientos funcionales

Tabla 1

Requerimientos funcionales.

Requerimiento	Descripción
RF-01	Se tendrán 2 roles que se clasificaron en dos usuarios, el primero administrador y el segundo el usuario dependencia.
RF-02	El usuario administrador será el usuario principal y podrá crear usuarios dependencia, en el sitio web.
RF-03	El usuario administrador podrá crear y modificar inventarios, encargados, elementos y usuarios dependencia.
RF-04	El administrador podrá crear nuevos checklist para la entrega de inventarios.

RF-05	El rol administrador deberá tener el rol de entregas de inventario para realizar el proceso de entrega inicial y final de los diferentes inventarios que cuenta cada dependencia.
RF-06	En la ventana principal del administrador se podrá visualizar los inventarios, encargados y usuarios dependencia.
RF-07	Para la creación de usuarios de dependencia se requerirá información como nombres, apellidos, email, dependencia y contraseñas.
RF-08	En el módulo administrador se podrá asignar inventarios a los diferentes encargados y elementos a los diferentes inventarios que cuenten con más de un elemento.
RF-09	En el módulo inventarios del rol administrador se podrá visualizar el estado de los elementos de todos los inventarios, clasificados de la siguiente manera (Bueno, Malo, Regular, Mantenimiento).
RF-10	El usuario administrador podrá generar y ver los reportes en pdf de los checklist finalizados para su posterior uso en informes.
RF-11	El usuario administrador podrá actualizar las fotos de los elementos desde el sitio web y teléfono móvil.
RF-12	El administrador podrá resolver o eliminar solicitudes realizadas por el usuario dependencia, para el cambio de estado de uso o ubicación del elemento.
RF-13	El administrador podrá ver las notificaciones de las solicitudes pendientes por resolver.
RF-14	El usuario administrador podrá dar de baja a un elemento modificando su estado a inactivo en la base de datos, permitiendo que este no aparezca en los inventarios.
RF-15	El usuario dependencia podrá ver sus inventarios asignados en su página de inicio.
RF-16	El usuario dependencia podrá ver toda la información de sus elementos en sus diferentes inventarios asignados, así mismo la foto de cada uno de los elementos.
RF-17	El usuario dependencia podrá generar solicitudes de los elementos que necesite hacerle una modificación de cambio de lugar físico o estado de uso de los elementos.

RF-18	El usuario dependencia podrá ver las solicitudes que aun el administrador no ha solucionado, así mismo podrá eliminarlas.
RF-19	Se dispondrá del uso de la cámara del dispositivo móvil para el escaneo del código de barras de cada uno de los elementos.
RF-20	Los dos usuarios podrán generar y descargar el código de barras para cada uno de los elementos cuando lo requieran en el sitio web.
RF-21	El sistema estará en dos plataformas las cuales son plataforma web y en aplicación móvil para Android.
RF-22	El sistema reconocerá el código de barras para el proceso de checklist.
RF-23	El color del diseño de la interfaz del sistema en las dos plataformas debe estar basado en los colores institucionales de la universidad de Cundinamarca.
RF-24	El usuario administrador podrá ver toda la información de los elementos en sus diferentes inventarios, así mismo la foto de cada uno de los elementos y de igual manera actualizar información pertinente.
RF-25	El usuario administrador contará con búsquedas por distintos filtros como: (Placa, inventario, descripción y lugar físico) en sus dos plataformas.
RF-26	El usuario administrador en el módulo de entregas de inventario podrá visualizar los elementos chequeados y chequeados de cada uno de los checklist que se encuentre activos.

Fuente: Autores del proyecto.

10.1.2. Requerimientos no funcionales

Tabla 2

Requerimientos no funcionales.

Requerimiento	Descripción
RNF-01	Para el acceso al sistema debe de disponerse de conexión a internet en las dos plataformas.

RNF-02	En plataforma móvil los dispositivos deberán tener como mínimo la versión de Android 4.1 para su funcionamiento.
RNF-03	El motor de base de datos será mysql y podrá ser accedida de manera simultánea por las dos plataformas sin problemas de tiempos de respuesta.
RNF-04	La información suministrada en las dos plataformas debe ser legible, clara y de fácil comprensión por parte de los usuarios.
RNF-05	El sitio web podrá visualizarse desde cualquier dispositivo de manera tal que su interfaz se adapte a los requerimientos de la pantalla.
RNF-06	La aplicación requerirá una cámara del dispositivo de mínimo 5 megapíxeles.
RNF-07	El dispositivo móvil debe contar con al menos 20 Mb de almacenamiento para poder instalar la aplicación.
RNF-08	El sistema multiplataforma debe ser fácil de utilizar e intuitivo para que los usuarios lo usen.
RNF-09	El sistema multiplataforma debe proporcionar tiempos de respuesta rápidos en cada uno de los procedimientos requeridos.
RNF-10	La aplicación debe estar disponible en la play store, para su descarga y posterior instalación.
RNF-11	El sistema debe proporcionar todo tipo de seguridad a la información de los usuarios.
RNF-12	El sistema debe poderse usar por diferentes usuarios a la vez.
RNF-13	El sistema debe ser sometido a diferentes pruebas unitarias y de aceptación antes de su implementación.
RNF-14	para el uso de sistema multiplataforma es recomendable tener una conexión a internet con una velocidad mínima a 2 Mbps simétrica.
RNF-15	Se requiere el uso de navegadores de internet confiables como: Google Chrome, Mozilla, Opera y Microsoft Edge.

Fuente: Autores del proyecto.

10.1.3. Historias de Usuario

Para la realización de las historias de usuario se tuvo en cuenta la información recopilada directamente con el cliente donde se especificó cada una de las características de los requerimientos funcionales. (ver anexo 3). En esta fase se contemplan dos tipos de historias de usuario, las cuales son para sitio web e historias de usuario para aplicativo móvil Android:

Tabla 3

Historias de usuario.

Historias de usuario				
Sitio web				
Número	Historia de Usuario	Prioridad	Iteración	Programador responsable
01	Ingreso al sistema	Alta	1	Nestor Castellanos
02	Entregas de inventario	Alta	2	
03	Búsqueda de elementos	Alta	1	
04	Opciones de elementos	Alta	1	
05	Creación de código de barras	Media	2	
06	Encargados	Alta	1	
07	Asignación de inventarios	Alta	2	
08	Solicitudes usuario dependencia	Baja	3	
09	Respuesta solicitudes	Baja	3	
10	Reportes	Alta	2	
11	Notificaciones	Baja	3	
Aplicación móvil Android				
01	Ingreso a la aplicación móvil	Alta	1	Stiven Rodríguez
02	Entregas de inventario	Alta	2	
03	Búsqueda de elementos	Alta	1	
04	Opciones de elementos	Alta	1	
05	Asignación de inventarios	Alta	2	
06	Solicitudes usuario dependencia	Baja	3	
07	Respuesta solicitudes	Baja	3	
08	Notificaciones	Baja	3	

Fuente: Autores del proyecto.

Cada tabla de las historias de usuario contiene los siguientes ítems:

Aplicación móvil.

Tabla 4

Formato historias de usuario.

Historia de usuario	
Número: Este permite numerar e identificar cada una de las historias de usuario con el fin de organizar la información.	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre Historia: Describe de manera general una historia de usuario.	Riesgo en Desarrollo: Valor de complejidad que una historia de usuario representa en el momento de realizar el desarrollo.
Prioridad en Negocio: Grado de importancia que el cliente asigna a una historia de usuario, esta se mide de la siguiente manera: (Alta / Media /Baja)	Iteración asignada: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Plataforma de desarrollo: Tipo de plataforma a desarrollar.	
Programador responsable: Rol encargado del desarrollo de la historia de usuario.	
Descripción: información relevante que permite especificar la funcionalidad que se implementará con la realización de la historia de usuario.	
Observación: Información que se anexa con el fin de aclarar un requerimiento planteado en la historia de usuario permitiendo extender y especificar las tareas a realizar.	

Fuente: Autores del proyecto.

Sitio Web

Tabla 5

Historia de usuario sitio web ingreso al sistema.

Ingreso al sistema	
Número historia: 01	Usuario: Administrador y usuario dependencia
Nombre de historia: Ingreso al sistema.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 1

Programador responsable: Nestor Castellanos

Descripción: Creación de Login mediante el email y la contraseña, manejo de sesiones.

Observación: El email de registro, sea el correo institucional.

Fuente: Autores del proyecto.

Tabla 6

Historia de usuario sitio web entrega de inventario.

Entrega de inventario

Número historia: 02

Usuario: Administrador

Nombre de historia: Entrega de Inventario.

Prioridad en negocio: Alta
(Alta / Media /Baja)

Riesgo en desarrollo: Alta
(Alta / Media /Baja)

Plataforma de desarrollo: Sitio web

Iteración asignada: 2

Programador responsable: Nestor Castellanos

Descripción: Módulo para realizar un chequeo de los elementos, ingresando el código de la placa de manera manual y también por medio de un lector de código de barras.

Observación:

Fuente: Autores del proyecto.

Tabla 7

Historia de usuario sitio web búsqueda de elementos.

Búsqueda de elementos

Número historia: 03

Usuario: Administrador y usuario dependencia

Nombre de historia: Búsqueda de elementos.

Prioridad en negocio: Alta
(Alta / Media /Baja)

Riesgo en desarrollo: Media
(Alta / Media /Baja)

Plataforma de desarrollo: Sitio web

Iteración asignada: 1

Programador responsable: Nestor Castellanos

Descripción: Creación de módulo para realizar búsquedas de los elementos por:

- Nombre
- Código del inventario
- Lugar físico
- Elementos chequeados y no chequeados.

Observación: Al realizar una búsqueda se debe contar la opción de visualizar la imagen del elemento, si el elemento no cuenta con imagen se le mostrará el icono del sistema por defecto.

Fuente: Autores del proyecto.

Tabla 8*Historia de usuario sitio web opciones de elementos.*

Opciones de elementos	
Número historia: 04	Usuario: Administrador
Nombre de historia: Opciones de elementos.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 1
Programador responsable: Nestor Castellanos	
Descripción: Crear modulo opciones para las diferentes acciones: <ul style="list-style-type: none"> • Añadir elementos e ingresar la información e imagen de este. • Actualizar el estado del elemento a los cuatro diferentes estados (Bueno, Regular, Malo y Mantenimiento). • En la opción dar de baja se realiza el cambio de usabilidad a inactivo. 	
Observación: En la opción eliminar, solo colocar inactivo al elemento en la base de datos.	
Fuente: Autores del proyecto.	

Tabla 9*Historia de usuario sitio web creación de código de barras.*

Creación de Código de barras	
Número historia: 05	Usuario: Administrador y usuario dependencia
Nombre de historia: Creación de código de barras.	
Prioridad en negocio: Media (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 2
Programador responsable: Nestor Castellanos	
Descripción: Añadir al módulo opciones: generar código de barras con la placa del elemento que asigne el usuario.	
Observación: Se deben realizar de acuerdo a la norma NTC 3842	
Fuente: Autores del proyecto.	

Tabla 10*Historia de usuario sitio web encargados.*

Encargados	
Número historia: 06	Usuario: Administrador

Nombre de historia: Información encargados.	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Prioridad en negocio: Alta (Alta / Media /Baja)	Iteración asignada: 1
Plataforma de desarrollo: Sitio web	
Programador responsable: Nestor Castellanos	
Descripción: Crear formulario para ingresar información de las personas encargadas de los inventarios y formulario para actualizar dependencia asignada.	
Observación:	

Fuente: Autores del proyecto.

Tabla 11

Historia de usuario sitio web asignación de inventarios.

Asignación de inventarios	
Número historia: 07	Usuario: Administrador
Nombre de historia: Asignación de inventarios.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en Desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 2
Programador responsable: Nestor Castellanos	
Descripción: Crear módulo para asignar los inventarios por dependencia a la persona que sea encargada de igual manera poder trasladar un elemento de un inventario a otro.	
Observación: para asignar los inventarios tiene que estar ingresado los datos en la base de datos y contar con usuario y contraseña.	

Fuente: Autores del proyecto.

Tabla 12

Historia de usuario sitio web solicitudes usuario dependencia.

Solicitudes Usuario dependencia	
Número historia: 08	Usuario: Usuario dependencia
Nombre de historia: Solicitudes usuario dependencia.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 3
Programador responsable: Nestor Castellanos	

Descripción: Crear módulo de solicitudes para realizar actualización en el inventario en el aspecto de lugar físico y estado de los elementos.

Observación:

Fuente: Autores del proyecto.

Tabla 13

Historia de usuario sitio web respuesta solicitudes.

Respuesta solicitudes	
Número historia: 09	Usuario: Administrador
Nombre de historia: Respuesta solicitudes.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 3
Programador responsable: Nestor Castellanos	
Descripción: Añadir modulo solicitudes con recepción de estas y dar solución o eliminarla.	
Observación:	

Fuente: Autores del proyecto.

Tabla 14

Historia de usuario sitio web reportes.

Reportes	
Número historia: 10	Usuario: Administrador
Nombre de historia: Reportes.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 2
Programador responsable: Nestor Castellanos	
Descripción: Crear módulos de reportes y se guarden en formato pdf en la base de datos para los siguiente:	
<ul style="list-style-type: none"> • Reportes cuando se realice la entrega inicial y final de los inventarios. 	
Observación:	
<ul style="list-style-type: none"> • Mostrar estado de los elementos de cada uno de los inventarios en el sitio web. • El reporte debe contar con información relevante del inventario chequeado 	

Fuente: Autores del proyecto.

Tabla 15*Historia de usuario sitio web notificaciones.*

Notificaciones	
Número historia: 12	Usuario: Administrador
Nombre de historia: Notificaciones.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Baja (Alta / Media /Baja)
Plataforma de desarrollo: Sitio web	Iteración asignada: 3
Programador responsable: Nestor Castellanos	
Descripción: Mostrar Notificaciones cuando el usuario dependencia realice solicitudes de cambios de alguno de los elementos.	
Observación: Mostrar el total de solicitudes pendientes.	

Fuente: Autores del proyecto.

Aplicación Móvil**Tabla 16***Historia de usuario aplicativo móvil ingreso a la aplicación.*

Ingreso a la aplicación	
Número historia: 01	Usuario: Administrador y Usuario dependencia
Nombre de historia: Ingreso a la aplicación.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación móvil Android	Iteración asignada: 1
Programador responsable: Stiven Rodríguez	
Descripción: Creación de Login mediante el email y la contraseña y manejo de sesiones.	
Observación: El email de registro, sea el correo institucional.	

Fuente: Autores del proyecto.

Tabla 17

Historia de usuario aplicativo móvil entrega de inventario.

Entrega de inventario	
Número historia: 02	Usuario: Administrador
Nombre de historia: Entrega de inventarios.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación móvil Android	Iteración asignada: 2
Programador responsable: Stiven Rodríguez	
Descripción: General Módulo para realizar un chequeo de los elementos, ingresando el código de la placa de manera manual y también por medio de la cámara del dispositivo móvil como lector de código de barras.	
Observación:	
Fuente: Autores del proyecto.	

Tabla 18

Historia de usuario aplicativo móvil búsqueda de elementos.

Búsqueda de elementos	
Número historia: 03	Usuario: Administrador y usuario dependencia.
Nombre de historia: Búsqueda de elementos del inventario.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación móvil Android	Iteración asignada: 1
Programador responsable: Stiven Rodríguez	
Descripción: Creación de módulo para realizar consultas de los elementos por: <ul style="list-style-type: none"> • Nombre • Código del inventario • Lugar físico • Elementos chequeados y no chequeados. 	
Observación: Al realizar una búsqueda se debe contar la opción de visualizar la imagen del elemento, si el elemento no cuenta con imagen se le mostrará el icono del sistema por defecto.	
Fuente: Autores del proyecto.	

Tabla 19

Historia de usuario aplicativo móvil opciones de elementos.

Opciones de elementos	
Número historia: 04	Usuario: Administrador
Nombre de historia: Opciones de elementos.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación móvil Android	Iteración asignada: 1
Programador responsable: Stiven Rodríguez	
Descripción: Crear modulo opciones para las diferentes acciones: <ul style="list-style-type: none"> • Añadir elementos e ingresar la información e imagen de este. • Actualizar el estado del elemento a los cuatro diferentes estados (Bueno, Regular, Malo, Mantenimiento). • En la opción eliminar se realice el cambio de usabilidad a inactivo. 	
Observación: En la opción eliminar, solo colocar inactivo al elemento en la base de datos.	

Fuente: Autores del proyecto.

Tabla 20

Historia de usuario aplicativo móvil asignación de inventarios.

Asignación de inventarios	
Número historia: 05	Usuario: Administrador
Nombre de historia: Asignación de inventarios.	
Prioridad en negocio: Alta (Alta / Media /Baja)	Riesgo en Desarrollo: Alta (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación móvil Android	Iteración asignada: 2
Programador responsable: Stiven Rodríguez	
Descripción: Crear módulo para asignar los inventarios por dependencia a la persona que sea encargada de igual manera poder trasladar un elemento de un inventario a otro.	
Observación: para asignar los inventarios tiene que estar ingresado los datos en la base de datos y contar con usuario y contraseña.	

Fuente: Autores del proyecto.

Tabla 21*Historia de usuario aplicativo móvil solicitudes usuario dependencia.*

Solicitudes usuario dependencia	
Número historia: 06	Usuario: Usuario dependencia
Nombre de historia: Solicitudes usuario dependencia.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación Móvil Android.	Iteración asignada: 3
Programador responsable: Stiven Rodríguez	
Descripción: Crear módulo de solicitudes para realizar actualización en el inventario en el aspecto de lugar físico y estado de los elementos.	
Observación:	

Fuente: Autores del proyecto.

Tabla 22*Historia de usuario aplicativo móvil respuesta solicitudes.*

Respuesta solicitudes	
Número Historia: 07	Usuario: Administrador
Nombre de historia: Respuesta solicitudes.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Media (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación Móvil Android.	Iteración asignada: 3
Programador Responsable: Stiven Rodríguez	
Descripción: Añadir modulo solicitudes con recepción de estas y dar solución o eliminarla.	
Observación:	

Fuente: Autores del proyecto.

Tabla 23*Historia de usuario aplicativo móvil notificaciones.*

Notificaciones	
Número historia: 08	Usuario: Administrador
Nombre de historia: Notificaciones.	
Prioridad en negocio: Baja (Alta / Media /Baja)	Riesgo en desarrollo: Baja (Alta / Media /Baja)
Plataforma de desarrollo: Aplicación Móvil Android.	Iteración asignada: 3

Programador responsable: Stiven Rodríguez

Descripción: Mostrar notificaciones cuando el usuario dependencia realice solicitudes de cambios de alguno de los elementos.

Observación: Mostrar el total de solicitudes pendientes al iniciar la sesión o cuando se genere una nueva.

Fuente: Autores del proyecto.

10.1.4. Velocidad del proyecto

En esta etapa de la fase de planeación se representa una medida que especificara la velocidad con la que se desarrollara el proyecto, para realizar este proceso se debe tener en cuenta la cantidad de historias de usuario totales, para asignarlas en cada una de las iteraciones y el tiempo en el que serán desarrolladas, esto permite que se controle las tareas que se están haciendo y clasificar el nivel de importancia que tiene cada una de las historias de usuario con el fin de planificar cual debe desarrollarse de primeras, de esta manera se brinda un mayor cumplimiento al cliente mediante el plan de entregas.

Tabla 24

Velocidad del proyecto.

	Iteración 1	Iteración 2	Iteración 3
Horas	300	275	250
Semanas	12	11	10
Horas Semanales	25	25	25
Historias de usuario	7	6	6

Fuente: Autores del proyecto.

Por medio de la velocidad de proyecto se pudo evidenciar la cantidad de horas que deben trabajarse semanalmente para cada una de las iteraciones, dependiendo las historias de usuario asignadas por el cliente, se evidencia que para la primera iteración se tienen disponibles 300 horas en el transcurso de 12 semanas, para la segunda iteración cuenta con 275 horas divididas en 11 semanas y para la tercera iteración se culmina con 250 horas en el transcurso de 10

semanas, para la velocidad de proyecto se contempló tener en cuenta los días laborables hábiles para su ejecución.

10.1.5. División de iteraciones

El proyecto se dividió en 3 iteraciones donde se combinó con el cliente cuáles historias de usuario se realizarán en cada iteración para definir las partes del sistema funcionales que serían entregadas al final de cada iteración, así mismo se definió las tareas de cada historia de usuario y los desarrolladores encargados del desarrollo de estas. Cabe recalcar que los tiempos de cada iteración son diferentes por la relevancia e importancia que cada historia de usuario tiene para el cliente, en la primera iteración se tuvieron en cuenta las historias de usuario más importantes y de mayor relevancia, por tal motivo el tiempo para esta iteración es mayor a las demás. En la segunda iteración el nivel de importancia se reduce, pero es mayor que la última iteración, finalmente la tercera iteración es la que cuenta con el menor tiempo para la realización del cumplimiento de las historias de usuario asignadas, pero es la iteración más importante para el cliente porque integra la entrega final del sistema.

Tabla 25

Iteración 01.

Iteración 01			
Plataforma	Número de historia	Nombre	Número de semana
Web	01	Ingreso al sistema	1-3
	03	Búsqueda de elementos	4-5
	04	Opciones de elementos	6-10
	06	Encargados	11-12
Móvil	01	Ingreso a la aplicación móvil	1-4
	03	Búsqueda de elementos	5-9
	04	Opciones de elementos	10-12

Fuente: Autores del proyecto.

Tabla 26*Iteración 02.*

Iteración 02			
Plataforma	Número de historia	Nombre	Número de semana
Web	02	Entregas de inventario	1-5
	05	Creación de código de barras	6
	07	Asignación de inventarios	7-8
	10	Reportes	9-11
Móvil	02	Entregas de inventario	1-6
	05	Asignación de inventarios	7-11

Fuente: Autores del proyecto.

Tabla 27*Iteración 03.*

Iteración 03			
Plataforma	Número de historia	Nombre	Número de semana
Web	08	Solicitudes usuario dependencia	1-5
	09	Respuesta solicitudes	6-8
	11	Notificaciones	9-10
Móvil	06	Solicitudes usuario dependencia	1-5
	07	Respuesta solicitudes	6-8
	08	Notificaciones	9-10

Fuente: Autores del proyecto.

10.1.6. Plan de Entregas

Para estas entregas se tuvo en cuenta la duración de cada iteración, al final de estas se realizaron las entregas con el cliente de las historias de usuario asignadas y con la funcionalidad de cada una de ellas en el sistema, esto permite un mejor desempeño del proyecto debido a que el cliente siempre está mostrando un mayor interés ya que siempre está recibiendo resultados y avances del desarrollo del proyecto.

Tabla 28*Plan de entregas.*

Iteración	Fecha	Duración de la entrega horas
1	28 de agosto de 2020	1:45:00
2	20 de noviembre 2020	1:15:00
3	15 de febrero 2020	2:00:00

Fuente: Autores del proyecto.

Cada de una de las entregas se dividió en dos partes, en primera medida se mostraba las funcionalidades acordadas a desarrollar durante cada iteración para que de esta manera el cliente diera su aprobación o recomendación sobre los módulos desarrollados, al culminar la reunión se socializaron las historias de usuario para desarrollar durante la siguiente iteración, de igual manera acordar la siguiente.

10.1.7. Reunión inicial de iteración

En estas reuniones que son realizadas por el equipo de desarrolladores y el cliente se establecen las fechas para la entrega de los avances de cada iteración, donde se evidencien las tareas realizadas por cada desarrollador en las historias de usuario asignadas para esa iteración. Así mismo estas reuniones permiten al cliente y desarrolladores exponer nuevas problemáticas o soluciones por parte de cada uno de ellos acerca del desarrollo del proyecto.

Tabla 29*Reunión inicial de iteración.*

Iteración	Fecha de reunión	Fecha de entrega
1	1 de junio de 2020	28 de agosto de 2020
2	31 de agosto 2020	20 de noviembre 2020
3	2 de noviembre 2020	15 de febrero 2020

Fuente: Autores del proyecto.

10.1.8. Roles XP

Programador: Es el encargado de desarrollar cada una de las historias de usuario definidas con el cliente, estimar los tiempos requeridos y la prioridad para el proceso de desarrollo y realizar un test de pruebas donde encuentre y solucione posibles errores en el sistema.

Encargado de pruebas (tester): Este se encarga de realizarle pruebas al sistema en las 2 plataformas tanto en web y móvil, así mismo tiene la responsabilidad de publicar los resultados de los test con los integrantes del proyecto, este rol puede ser ocupado por cualquiera de los desarrolladores del sistema, ojalá este sea quien no haya realizado el desarrollo del módulo a probar.

Cliente: Es la persona que brinda los requerimientos para el sistema multiplataforma, de igual manera esta se encarga de seleccionar las historias de usuario por cada iteración y por ultima es la persona que aprueba el sistema o se deben realizar cambios.

Usuario: Persona específica que usará el sistema multiplataforma con un rol asignado que le permite acceder a todos los servicios que este tenga dentro de sus funcionalidades.

Consultor: persona externa al equipo de desarrollo que brinda conocimientos específicos en las distintas áreas a tratar durante el proceso del proyecto multiplataforma en sus múltiples fases de desarrollo, también sirviendo de guía para cumplir con todos los objetivos propuestos por el cliente.

Tabla 30*Roles XP.*

Rol	Integrante
Programador y encargado de pruebas	Nestor Ivan Castellanos Laiton Manuel Stiven Rodríguez Lamprea
Cliente	Elva Lucia Triana Ángel
Usuario	Personal administrativo que cuente con un inventario a cargo dentro de la universidad de Cundinamarca seccional Ubaté
Consultor	Directora de proyecto: Ing. Ana Lucia Hurtado Mesa Planta de docentes de la facultad de ingeniería.

Fuente: Autores del proyecto.

10.2. Diseño

La fase de diseño es realizada durante todo el ciclo de vida del proyecto, siendo sometida a diferentes cambios y modificaciones que se evidenciaron durante el transcurso del desarrollo del proyecto, para el sistema multiplataforma se contemplaron varios procedimientos basados en la metodología xp como: simplicidad en el diseño, metáfora del sistema, tarjetas crc y la refactorización. Basado en los requerimientos recopilados en la fase de planeación se tomaron las historias de usuario como base para el desarrollo de los ítems anteriormente nombrados, permitiendo que para la siguiente fase se facilite la comprensión de las funcionalidades que deben cumplirse en el proceso de codificación.

10.2.1. Simplicidad en el diseño

En la realización del sistema multiplataforma la metodología aplicada brinda herramientas y estrategias que buscan la agilización de la entrega de resultados al cliente, por tal motivo la sencillez en los diferentes procedimientos es fundamental para lograr estos objetivos, ya que se logra la optimización de tiempos y una mayor eficiencia en las tareas del proyecto. De acuerdo a lo anterior se tuvieron en cuenta los diagramas y diseños más relevantes y de mayor importancia para la fase de diseño, para esto se realizaron los diseños de la arquitectura cliente-servidor, arquitectura de funcionamiento sistema multiplataforma, tarjetas CRC, diagramas de secuencia, diagramas de clase y modelo entidad relación de la base de datos.

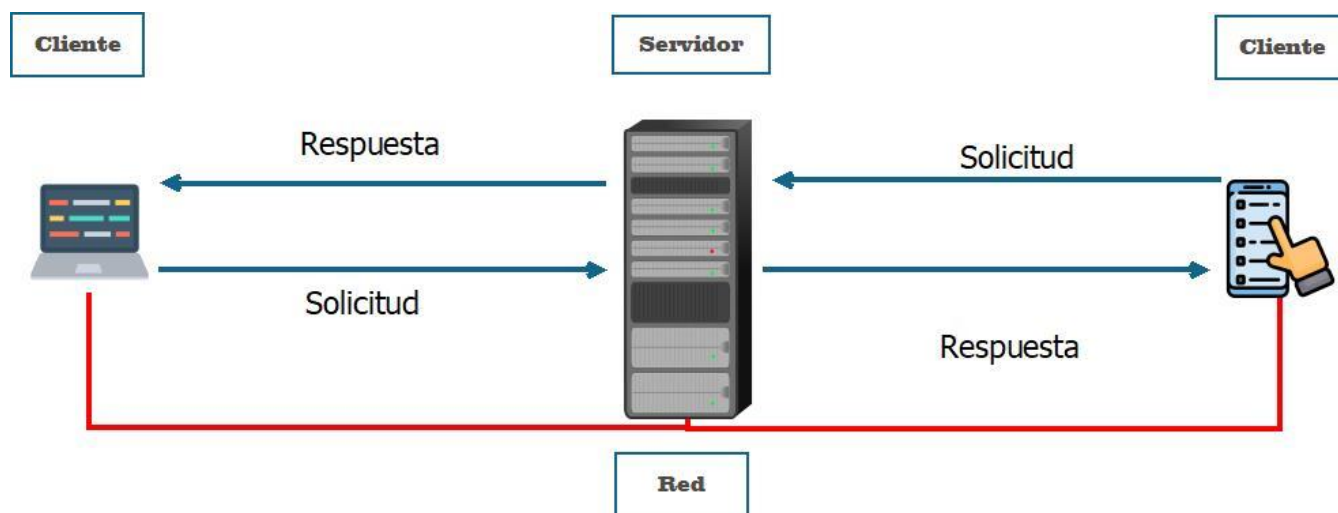
10.2.2. Metáfora del sistema

Para la comprensión del funcionamiento del sistema multiplataforma se plasman los diseños de las arquitecturas cliente-servidor y de funcionamiento como alternativas para lograr aplicar la sencillez del proyecto de acuerdo con la metodología y facilitar la codificación.

10.2.2.1. Arquitectura cliente-servidor

Figura 9

Arquitectura del sistema.



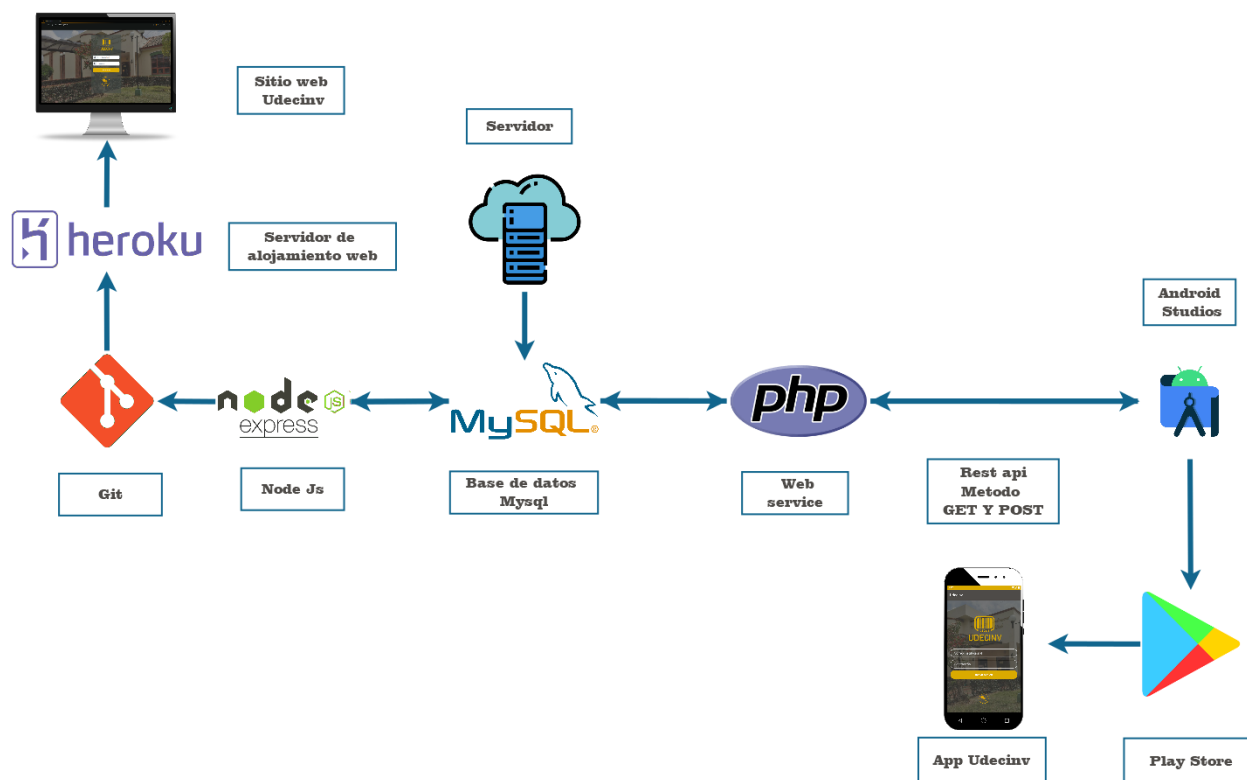
Fuente: Autores del proyecto.

En esta arquitectura se evidencia la centralización de la información en un servidor donde actúa como gestor de base de datos y servidor principal para el alojamiento del sistema tanto en sitio web como en aplicación móvil, posteriormente se puede ver los tipos de clientes que actúan como gestores de peticiones para recibir una respuesta por parte del servidor, estos clientes pueden estar presentes desde celulares Android y sitios web desde cualquier dispositivo, finalmente el que hará como intermediario entre el servidor y los clientes es la conexión a internet, la cual permitirá que la información sea distribuida entre todos los roles.

10.2.2.2. Arquitectura sistema multiplataforma Udecinv.

Figura 10

Arquitectura del sistema multiplataforma.



Fuente: Autores del proyecto.

Para la arquitectura del sistema multiplataforma, se manejan las plataformas web y móvil cada de una de estas cuenta con una serie de módulos que fueron especificadas por el cliente, cuenta con una totalidad de 19 módulos divididos en 11 para web y 9 en móvil, estos para la interacción del usuario administrador y el usuario dependencia, por lo tanto el sistema en web tiene los siguientes módulos: (Ingreso al sistema “Login”, entregas de inventario, búsqueda de elementos, opciones de elementos, creación de código de barras, encargados, asignación de inventarios, solicitudes usuario dependencia, respuesta solicitudes, reportes y notificaciones) cada uno de estos hace la integración de poder tener la gestión de inventarios desde el sitio web, de igual manera el sistema realiza una serie de funciones interconectando diferentes herramientas

como: (Heroku, Git, Node y Mysql) que son esenciales para poder tener una conexión en tiempo real con el aplicativo móvil, por consiguiente en la aplicación móvil se tiene los módulos: (Ingreso a la aplicación móvil, entregas de inventario, búsqueda de elementos, opciones de elementos, asignación de inventarios, solicitudes usuario dependencia, respuesta solicitudes y notificaciones) igualmente como en el sitio web esto realizan una integración para poder llevar a cabo la de gestión de inventario, para llevar a cabo la conexión con la base de datos se utiliza una Rest Api que es la encargada de interactuar con la web (“ World Wide Web”) realizando consultas a la base de datos que se encuentra alojada en un servidor en la nube.

10.2.2.3. Herramientas del sistema multiplataforma

Bootstrap 4, conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones web, se usó para estilizar los diseños de interfaz del sistema multiplataforma en sitio web.

Figura 11

Bootstrap 4.



Fuente: Tomado de bootstrap

CSS3, hojas de estilo en cascada que permiten estilizar lenguaje estructurado de html mediante su diseño gráfico, se usó para estilizar las interfaces en el sitio web del sistema multiplataforma.

Figura 12

CCS3.



Fuente: Tomado de CCS3

Html5, lenguaje estructurado de hipertexto para la creación de páginas web, el cual permitió realizar el proceso de desarrollo de las estructuras de interfaz para la manipulación del sistema en el sitio web.

Figura 13

Html5.



Fuente: Tomado de html

JavaScript, lenguaje de programación interpretado que se implementa normalmente para la visualización en navegadores web del lado del cliente, pero que en la actualidad también

puede darse del lado del servidor como en el entorno de programación nodejs, en el sistema multiplataforma este lenguaje se encuentra del lado del cliente y del servidor en diferentes funcionalidades gracias al uso de nodejs como lenguaje de programación para el backend.

Figura 14

JavaScript.



Fuente: Tomado de JS

Git, sistema de control de versiones distribuido y gratuito de código abierto diseñado para el manejo de proyectos, en este caso se implementa en la preparación de los archivos para subir al servidor.

Figura 15

Git.



Fuente: Tomado de Git

Nodejs, entorno de ejecución para JavaScript del lado del servidor construido en el motor de JavaScript v8 de Chrome, es una de las principales herramientas de desarrollo para sistema multiplataforma, ya que permite la manipulación de la información de las bases de datos con las vistas en el sitio web, además de ser el gestor de toda la estructura lógica del sistema en el sitio web desde el backend.

Figura 16

nodejs.



Fuente: Tomado de nodejs.

Express js, es un marco de aplicación web de back-End para nodejs de código abierto que permite facilitar la interacción de las diferentes clases utilizadas durante el desarrollo del sistema multiplataforma.

Figura 17

Express js.



Fuente: Tomado de Expressjs.

Heroku, es una plataforma como servicio de computación en la nube que soporta distintos lenguajes de programación, actualmente el sistema multiplataforma está alojado en este servidor permitiendo su acceso al sitio web desde cualquier parte del mundo donde se cuente con conexión a internet.

Figura 18

Heroku.



Fuente: Tomado de Heroku.

MySQL, Gestor de base de datos relacional, multihilo y multiusuario de código abierto, siendo uno de los más usados en el mundo, se usó para el almacenamiento de los datos que se utilizan para la gestión de inventarios en la Universidad de Cundinamarca seccional Ubaté.

Figura 19

Mysql.



Fuente: Tomado de mysql.

Android Studio, entorno de desarrollo oficial integrado oficial para plataformas de Android, se usó como entorno de codificación para todo el aspecto lógico y funcional y así mismo sus interfaces gráficas de la aplicación para teléfonos móviles, de igual manera el uso de los emuladores de Android con distintas Api para verificar su compatibilidad.

Figura 20

Android Studio.



Fuente: Tomado de Android.

Rest Api, Conjunto de reglas específicas que son utilizadas para comunicarse entre otras aplicaciones, es un mecanismo útil para conectar dos softwares entre sí, de esta manera garantizar el intercambio de información, se utilizó como puente de comunicación entre la aplicación móvil y el servidor donde se encuentra alojado las consultas a la base de datos por los métodos Get y Post.

Figura 21

Rest Api.



Fuente: Tomado de pixabyte.

Php 7, es un lenguaje de programación adaptado especialmente para el desarrollo web, se usó como intermediario entre el servidor y la base de datos realizando consultas y emitiendo una respuesta que será procesada por la aplicación móvil.

Figura 22

php7.



Fuente: Tomado de php.

Google Play, plataforma de distribución digital de aplicación móviles para dispositivos con sistema operativo Android operada por Google, se usó para el alojamiento de la aplicación móvil para que los usuarios puedan acceder de una manera más simple y segura.

Figura 23

Google Play.



Fuente: Tomado de play store.

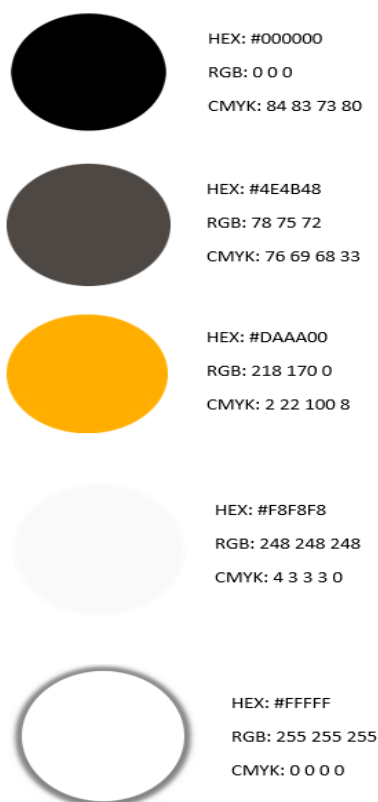
10.2.2.4. Definición de aspectos básicos de interfaz

Gama de colores

La gama de colores que se implementó para el sistema multiplataforma está basada en los colores institucionales de la Universidad de Cundinamarca dados en el manual de identidad institucional, estos se utilizaron en las distintas interfaces, formularios, páginas y logotipo del sistema Udecinv.

Figura 24

Gama de colores.



Fuente: Autores del proyecto.

10.2.2.5. Logotipo

El logo implementado en el sistema multiplataforma está basado en el uso del código de barras por tal motivo se puede apreciar este en la parte superior, así mismo hace parte del logo el nombre del sistema y el color amarillo utilizado como color secundario en la interfaz del sistema.

Figura 25

Logotipo.



Fuente: Autores del proyecto.

10.2.3. Tarjetas CRC

Las tarjetas crc permiten plasmar por medio de tablas la lógica procedimental en clases que será implementada para la realización del sistema multiplataforma, estas tablas están compuestas por el nombre de la clase, las responsabilidades y los colaboradores, los cuales permiten clasificar las funcionalidades con las que cuenta cada uno de los módulos, lo anteriormente mencionado permite darle una mayor sencillez a la codificación del proyecto aumentando la eficacia y reducir el tiempo de las entregas.

Tabla 31*Tarjetas CRC Login.*

Tarjeta CRC	
Nombre de la clase: Login	
Responsabilidades	Colaboradores
Registrar datos	
Validar información de usuario	PassEncrypt ()

Fuente: Autores del proyecto.

Tabla 32*Tarjetas CRC entregas de inventario.*

Tarjeta CRC	
Nombre de la clase: Entregas de inventario	
Responsabilidades	Colaboradores
Crear checklist	
Guardar checklist	
Finalizar checklist	Generar reporte

Fuente: Autores del proyecto.

Tabla 33*Tarjetas CRC búsqueda de elementos.*

Tarjeta CRC	
Nombre de la clase: Búsqueda de elementos	
Responsabilidades	Colaboradores
Ingresar atributo de elemento	
Buscar elemento	
Mostrar elemento	Lista inventarios

Fuente: Autores del proyecto.

Tabla 34*Tarjetas CRC opciones de elementos.*

Tarjeta CRC	
Nombre de la clase: Opciones de elementos	
Responsabilidades	Colaboradores
Ingresar elemento	
Modificar elemento	
Guardar elemento	
Ver elementos	Inventarios

Fuente: Autores del proyecto.

Tabla 35*Tarjetas CRC creación de código de barras.*

Tarjeta CRC	
Nombre de la clase: Creación código de barras	
Responsabilidades	Colaboradores
Crear código de barras	

Fuente: Autores del proyecto.

Tabla 36*Tarjetas CRC encargados.*

Tarjeta CRC	
Nombre de la clase: Encargados	
Responsabilidades	Colaboradores
Crear encargado	
Modificar encargado	
Guardar encargado	
Ver encargados	

Fuente: Autores del proyecto.

Tabla 37*Tarjetas CRC asignación de inventarios*

Tarjeta CRC	
Nombre de la clase: Asignación inventarios	

Responsabilidades	Colaboradores
Cargar encargados	inventarios
Asignar encargado	

Fuente: Autores del proyecto.

Tabla 38

Tarjetas CRC solicitudes usuario dependencia.

Tarjeta CRC	
Nombre de la clase: Solicitudes usuario dependencia	
Responsabilidades	Colaboradores
Crear solicitud	
Guardar solicitud	
Ver solicitudes	
Eliminar solicitud	

Fuente: Autores del proyecto.

Tabla 39

Tarjetas CRC respuesta solicitud.

Tarjeta CRC	
Nombre de la clase: Respuesta solicitud	
Responsabilidades	Colaboradores
Responder solicitud	
Ver solicitudes	
Eliminar solicitudes	

Fuente: Autores del proyecto.

Tabla 40

Tarjetas CRC reportes.

Tarjeta CRC	
Nombre de la clase: Reportes	
Responsabilidades	Colaboradores
Generar reporte pdf	
Ver reporte	Entregas

Fuente: Autores del proyecto.

Tabla 41

Tarjetas CRC notificaciones.

Tarjeta CRC	
Nombre de la clase: Notificaciones	
Responsabilidades	Colaboradores
Ver notificación	solicitudes

Fuente: Autores del proyecto.

Diagramas de caso de uso

El proceso de aplicar los casos en el sistema multiplataforma servirá para obtener un comportamiento o manera de uso en el sitio web y aplicación móvil por parte de los roles con lo que cuenta el sistema en este caso el usuario administrador o el usuario dependencia e, estos serán representado por medio de diagramas de UML, hay que mencionar que se clasificaron las tablas de caso de uso y los diagramas de secuencia según la iteración seleccionada por el cliente.

Lista de casos de uso web Usuario administrador

Tabla 42

lista de casos de uso web usuario administrador.

IDENTIFICADOR	NOMBRE	ITERACIÓN	ACTOR
CU_W01	Login	1	
CU_W02	Encargados	1	
CU_W03	Búsquedas	1	
CU_W04	Inventarios	2	
CU_W05	Opciones Elementos	2	
CU_W06	Generar código de barras	2	Usuario administrador
CU_W07	Entrega de Inventarios	2	
CU_W08	Solicitudes	3	
CU_W10	Reportes	2	
CU_W11	Notificaciones	3	

Fuente: Autores del proyecto.

Lista de casos de uso web Usuario dependencia

Tabla 43

lista de casos de uso web usuario dependencia.

IDENTIFICADOR	NOMBRE	ITERACIÓN	ACTOR
CU_W01	Login	1	
CU_W03	Búsquedas	1	
CU_W06	Generar código de barras	2	Usuario dependencia
CU_W09	Solicitudes	3	

Fuente: Autores del proyecto.

Lista de casos de uso móvil Usuario administrador

Tabla 44

lista de casos de uso móvil usuario administrador.

IDENTIFICADOR	NOMBRE	ITERACIÓN	ACTOR
CU_M01	Login	1	
CU_M02	Búsquedas	1	
CU_M03	Solicitudes	3	
CU_M05	Notificaciones	3	
CU_M06	Opciones elementos	1	Usuario administrador
CU_M07	Entregas de inventario	2	
CU_M08	Asignación de inventarios	2	

Fuente: Autores del proyecto.

Lista de casos de uso móvil Usuario dependencia

Tabla 45

lista de casos de uso móvil usuario dependencia.

IDENTIFICADOR	NOMBRE	ITERACIÓN	ACTOR
CU_M01	Login	1	

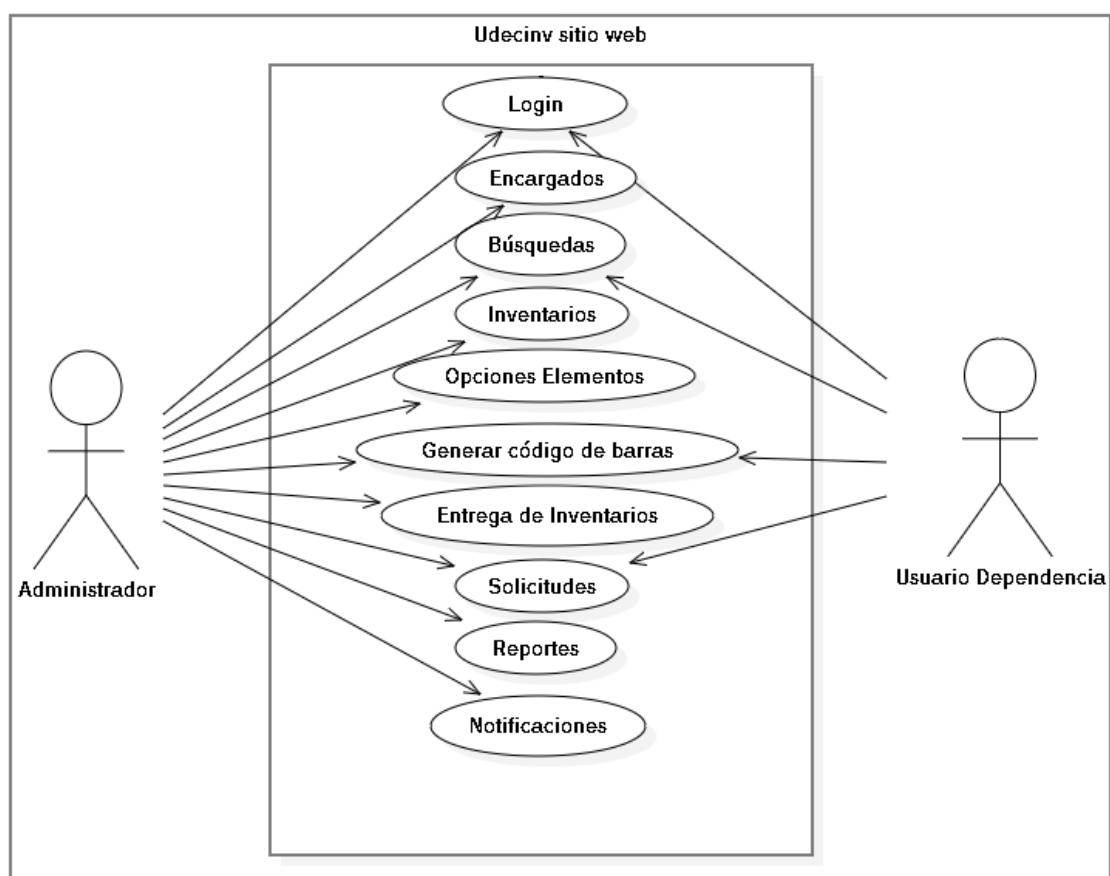
CU_M02	Búsquedas	1	Usuario dependencia
CU_M04	Solicitudes	3	

Fuente: Autores del proyecto.

10.2.4. Caso de uso sitio web

Figura 26

Diagrama de caso de uso sitio web.



Fuente: Autores del proyecto.

Tabla 46

Actor sitio web administrador.

Actor	Administrador	Identificador	AC01
-------	---------------	---------------	------

Descripción	El usuario administrador es el encargado del correcto funcionamiento del sistema multiplataforma, con este rol se puede realizar el proceso de entregas de los elementos de los inventarios por dependencia, agregar y asignar los inventarios al personal administrativo, cuenta con la opción de (Buscar, Agregar y Actualizar) los artículos del inventario. De igual manera podrá generar reportes sobre el checklist de los inventarios de la seccional Ubaté. Finalmente se encarga de llevar el control de las sesiones de los usuarios que cuenten con un inventario a cargo.	
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Dependencia, Email y Contraseña.	
Relación	Este rol tiene una relación indirecta con el usuario dependencia por medio del Login, búsquedas, generar código de barras y solicitudes.	
Referencias	Login, encargados, búsquedas, inventarios, opciones elementos, generar código de barras, entrega de inventarios, solicitudes, reportes y notificaciones.	
	Atributos	
Nombre	Descripción	Tipo
Id_Sesion	Identificador primario	Numérico
Nombres	Nombre del administrador	Alfabético
Apellidos	Apellido del administrador	Alfabético
Rol	Tipo de usuario	Alfabético
Dependencia	Área administrativa que tiene a su cargo varios inventarios.	Alfabético
Email	Correo institucional	Alfanumérico
Contraseña	Clave de verificación de usuario	Alfanumérico

Fuente: Autores del proyecto.

Tabla 47

Actor sitio web usuario dependencia.

Actor	Usuario Dependencia	Identificador	AC02
Descripción	El usuario dependencia está dirigido para las personas que se les asignan los inventarios por un periodo de tiempo, allí tendrán opciones de búsquedas de los elementos del inventario, realizar solicitudes al administrador para actualizar los elementos del inventario, cuenta con la opción de crear códigos de barras para los elementos que no cuenten con este.		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Dependencia, Email y Contraseña.		
Relación	Este rol tiene una relación indirecta con el usuario administrador por medio del Login, las búsquedas, generar código de barras y las solicitudes.		
Referencias	Login, búsquedas, generar código de barras, notificaciones.		
	Atributos		
Nombre	Descripción	Tipo	

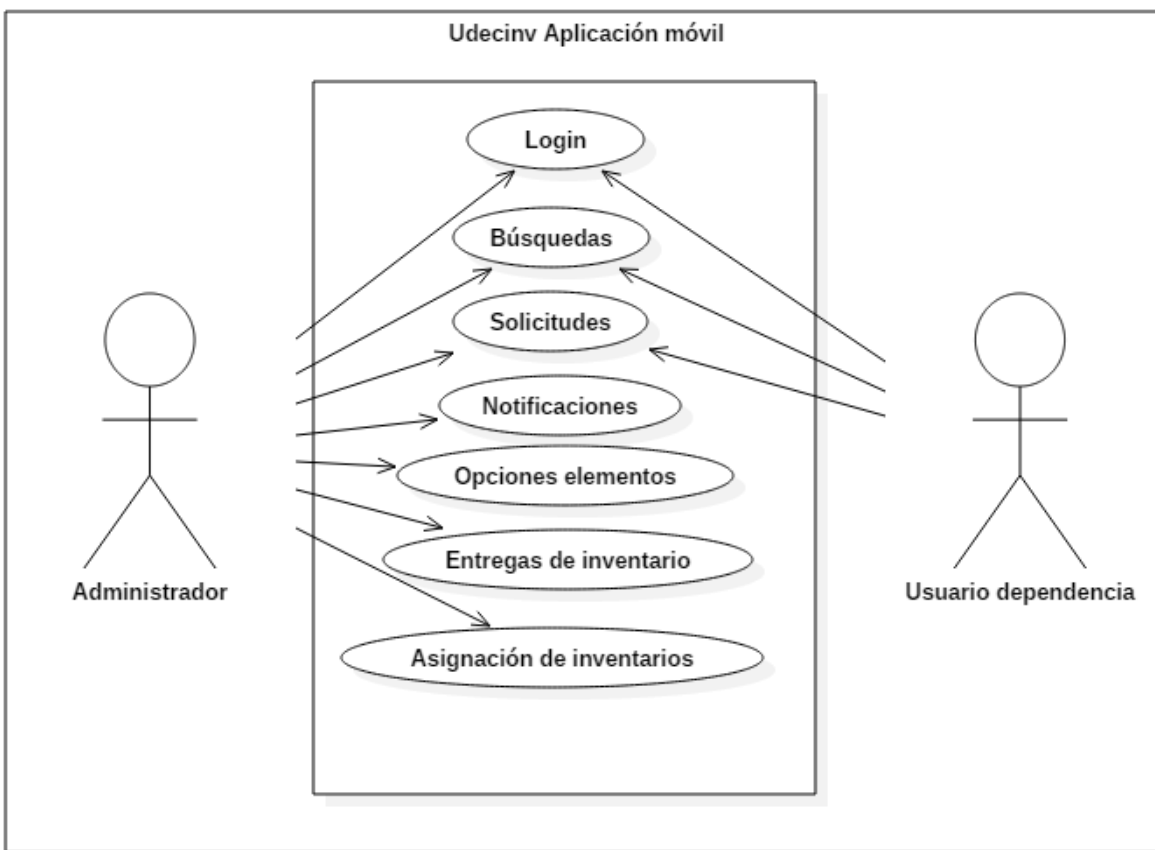
Id_Sesion	Identificador primario	Numérico
Nombres	Nombre del usuario dependencia	Alfabético
Apellidos	Apellido del usuario dependencia	Alfabético
Rol	Tipo de usuario	Alfabético
Dependencia	Área administrativa que tiene a su cargo varios inventarios.	Alfabético
Email	Correo institucional	Alfanumérico
Contraseña	Clave de verificación de usuario	Alfanumérico

Fuente: Autores del proyecto.

10.2.5. Caso de uso aplicación móvil

Figura 27

Diagrama de caso de uso aplicación móvil.



Fuente: Autores del proyecto.

Tabla 48

Actor aplicativo móvil administrador.

Fuente: Autores del proyecto.

Actor	Administrador	Identificador	AC01
Descripción	El usuario administrador es el encargado del correcto funcionamiento del sistema multiplataforma ya que su rol es el más importante, este cuenta con funciones como: entregas de inventarios, búsquedas (placa, descripción, inventario, lugar físico, elementos chequeado y no chequeados), opciones de elementos (añadir, dar de baja y actualizar), asignación de inventarios (asignar inventario a otro encargado y trasladar elemento de inventario), solucionar solicitudes, notificaciones sobre solicitudes pendientes a solucionar.		
Características	El actor cuenta con un: Id, Nombres, Apellidos, Rol, Email, Contraseña y Dependencia		
Relación	Este rol tiene una relación indirecta con el usuario dependencia por medio del Login, Búsquedas y Solicitudes.		
Referencias	Login, búsquedas, solicitudes, notificaciones, Opciones elementos, Entrega inventarios, Asignación de inventarios		
Atributos			
Nombre	Descripción		Tipo
Id_Sesion	Identificador primario		Numérico
Nombres	Nombre del administrador		Alfabético
Apellidos	Apellido del administrador		Alfabético
Rol	Tipo de usuario		Alfabético
Email	Correo institucional		Alfanumérico
Contraseña	Clave de verificación de usuario		Alfanumérico
Dependencia	Dependencia administrativa a la que pertenece		Alfanumérico

Fuente: Autores del proyecto.

Tabla 49

Actor aplicativo móvil usuario dependencia.

Actor	Usuario Dependencia	Identificador	AC02
Descripción	El usuario dependencia es para las personas que se le asigna los inventarios de las distintas dependencias administrativas, esta cuenta con funciones como: opciones de búsquedas (placa, descripción, inventario y lugar físico) y realizar solicitudes al administrador para actualizar los elementos del inventario (Lugar físico y cambio de estado de uso).		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Email, Contraseña y Email.		

Relación	Este rol tiene una relación indirecta con el administrador por medio del Login, Búsquedas y Solicitudes.		
Referencias	Login, búsquedas y solicitudes		
Atributos			
Nombre	Descripción		Tipo
Id_Sesion	Identificador primario		Numérico
Nombres	Nombre del usuario dependencia		Alfabético
Apellidos	Apellido del usuario dependencia		Alfabético
Rol	Tipo de usuario		Alfabético
Email	Correo institucional		Alfanumérico
Contraseña	Clave de verificación de usuario		Alfanumérico
Dependencia	Dependencia administrativa a la que pertenece		Alfanumérico

Fuente: Autores del proyecto.

Iteración 1

Caso de uso web

Tabla 50

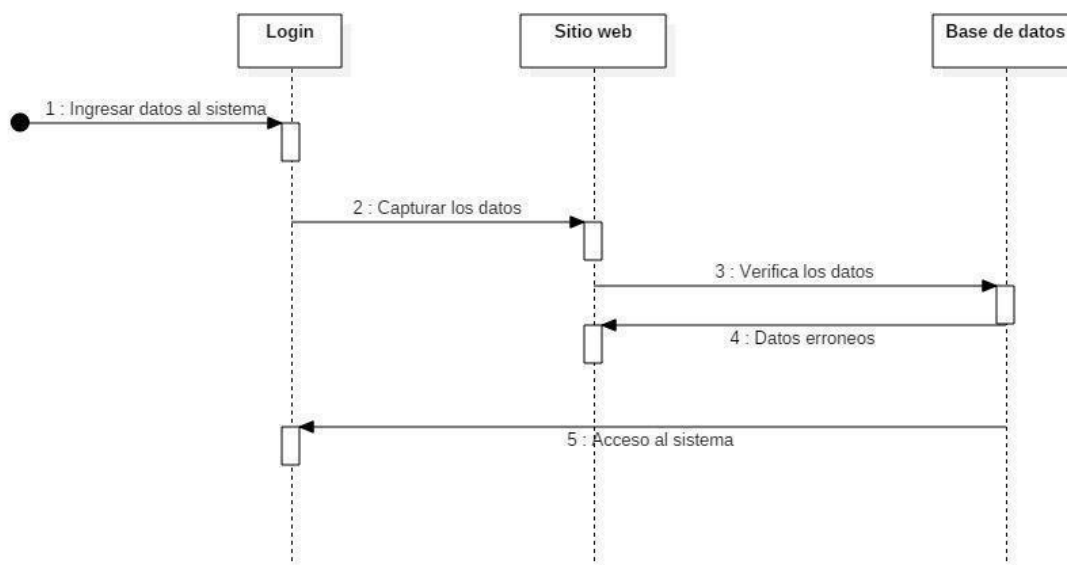
Caso de uso web: Login.

Caso de Uso	Login	Identificador	CU_W01
Actores	Administrador y Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	El usuario debe estar registrado.		
Postcondición	Verificar que los datos ingresados sean correctos.		
Descripción	El sitio web cuenta con un manejo de sesiones que según el rol asignado dará las opciones de uso dentro de este.		
Resumen			

Fuente: Autores del proyecto.

Figura 28

Diagrama de secuencia Login sitio web.



Fuente: Autores del proyecto.

Tabla 51

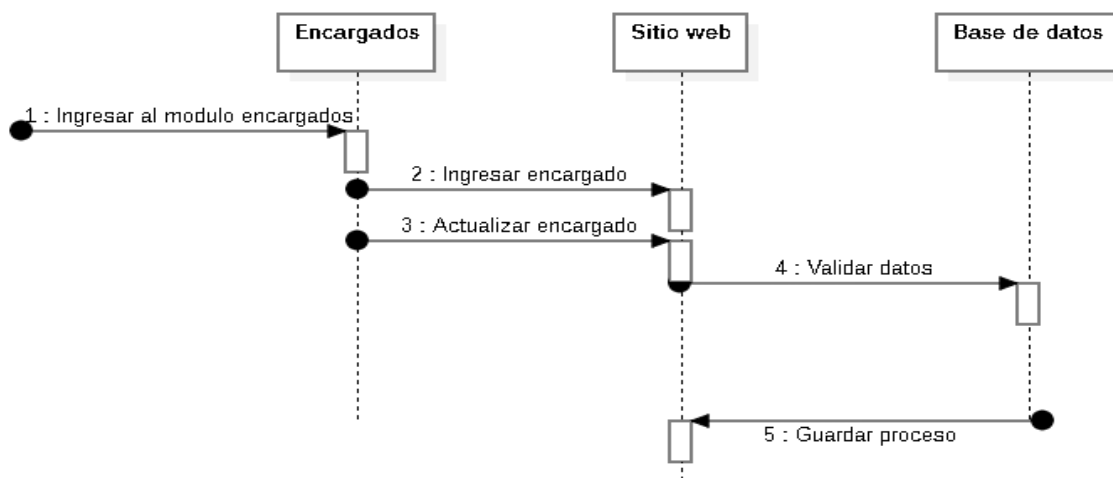
Caso de uso web: Encargados.

Caso de Uso	Encargados	Identificador	CU_W02
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	El usuario debe estar registrado como administrador.		
Postcondición	Mostrar información verídica de los encargados de los inventarios, verificar datos correctos.		
Descripción	En este módulo las opciones que más se destacan son el ingreso de nuevos encargados y modificación de los mismos con un usuario que le sea asignado.		
Resumen	Los encargados cumplen un rol importante en la gestión de inventarios, ya que cada elemento del inventario debe estar asignado a un inventario y a un encargado, esto permitiendo dar una mayor organización a la información.		

Fuente: Autores del proyecto.

Figura 29

Diagrama de secuencia encargados sitio web.



Fuente: Autores del proyecto.

Tabla 52

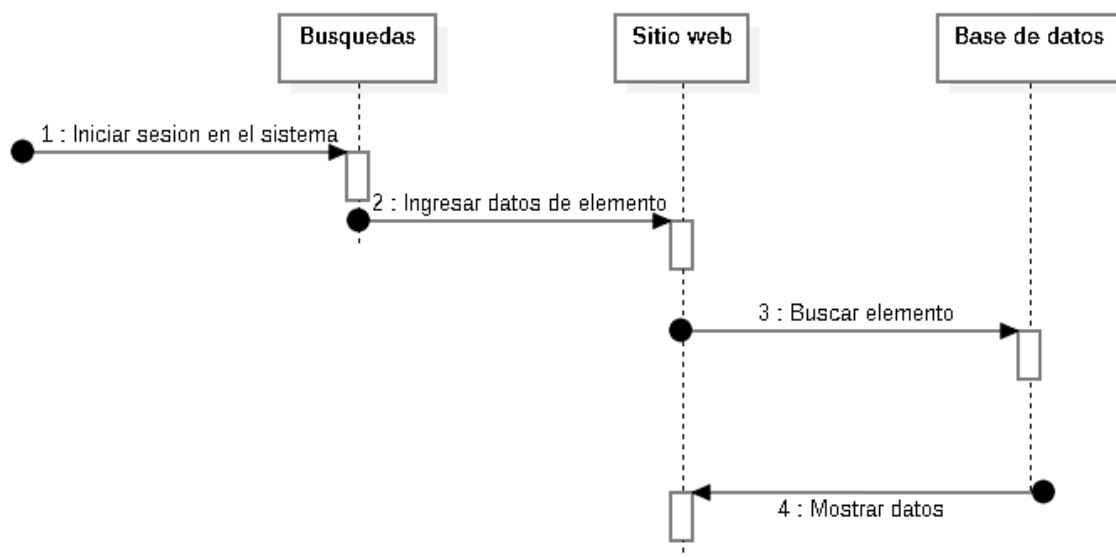
Caso de uso web: Búsquedas.

Caso de Uso	Búsquedas	Identificador	CU_W03
Actores	Administrador y Usuario Dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	Abrir el módulo de búsquedas.		
Postcondición	Mostrar información de la búsqueda realizada.		
Descripción	Los actores del sistema cuentan con la opción de realizar búsquedas por medio de diferentes filtros como: Placa, descripción, código del inventario, lugar físico, elementos chequeados y no chequeados.		
Resumen	Para las búsquedas de rol usuario dependencia sólo se visualizarán las asignada a la dependencia que pertenezca.		

Fuente: Autores del proyecto.

Figura 30

Diagrama de secuencia búsquedas sitio web.



Fuente: Autores del proyecto.

Caso de uso móvil

Tabla 53

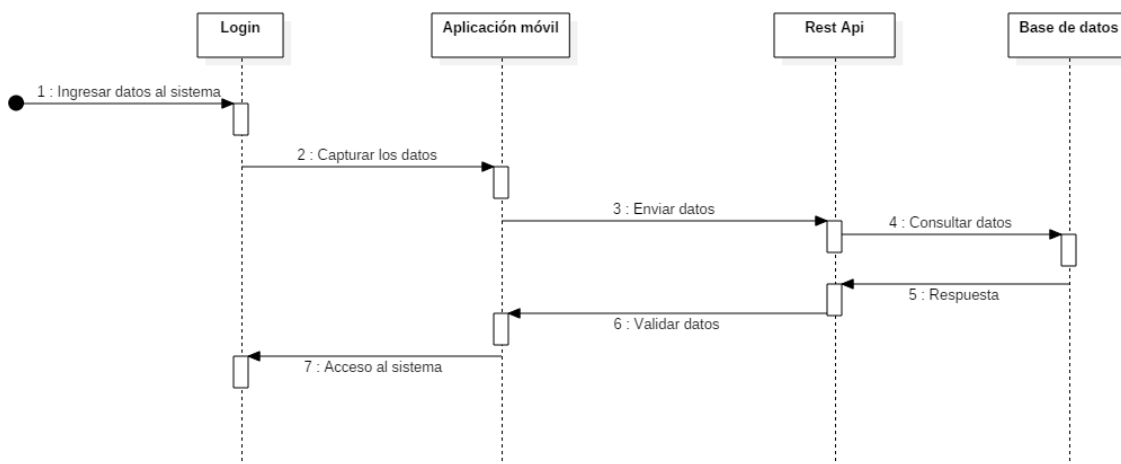
Caso de uso aplicativo móvil: Login.

Caso de Uso	Login	Identificador	CU_M01
Actores	Administrador y Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	El usuario debe estar registrado.		
Postcondición	Verificar que los datos ingresados sean correctos.		
Descripción	El sitio web cuenta con un manejo de sesiones que según el rol asignado dará las opciones de uso dentro de este.		
Resumen			

Fuente: Autores del proyecto.

Figura 31

Diagrama de secuencia Login aplicación móvil.



Fuente: Autores del proyecto.

Tabla 54

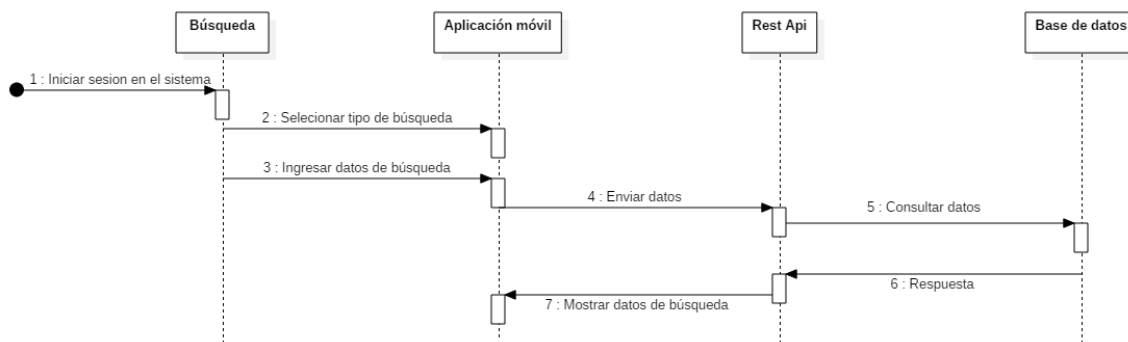
Caso de uso aplicativo móvil: Búsquedas.

Caso de Uso	Búsquedas	Identificador	CU_M02
Actores	Administrador y Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar el tipo de búsqueda a realizar.		
Postcondición	Mostrar la información de la búsqueda realizada.		
Descripción	Los actores del sistema cuentan con la opción de realizar búsquedas por medio de diferentes filtros como: placa, descripción, inventario, lugar físico, elementos chequeados y no chequeados.		
Resumen	Para las búsquedas del rol usuario dependencia sólo se visualizarán las asignadas a la dependencia que pertenezca.		

Fuente: Autores del proyecto.

Figura 32

Diagrama de secuencia búsquedas aplicación móvil.



Fuente: Autores del proyecto.

Tabla 55

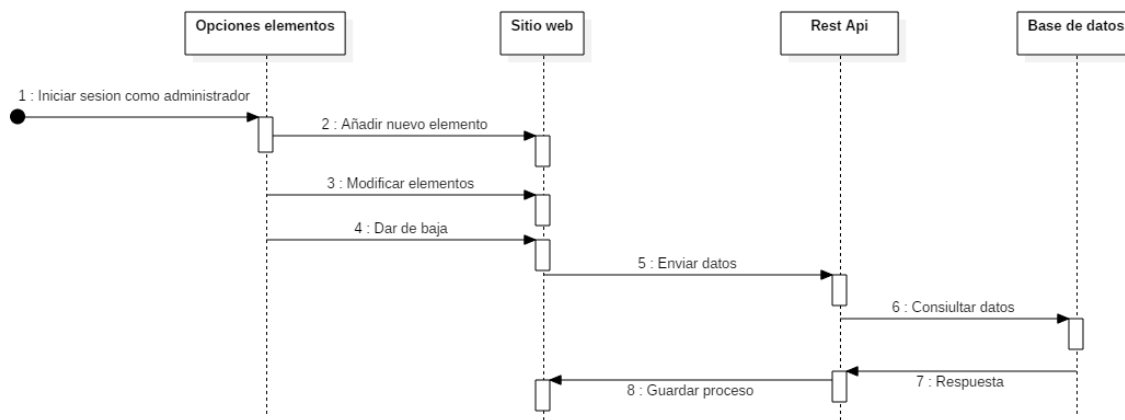
Caso de uso aplicativo móvil: Opciones elementos.

Caso de Uso	Opciones elementos	Identificador	CU_M03
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar opción a realizar.		
Postcondición	Ninguna.		
Descripción	El usuario administrador cuenta con la opción de añadir nuevos elementos al inventario de cada una de las dependencias, así mismo modificar la información de estos desde los datos hasta la imagen, finalmente el elemento puede darse de baja del sistema con la opción de cambio de estado.		
Resumen	Todos los elementos del inventario podrán ser modificados desde este módulo en cada uno de los inventarios donde estos se encuentren, haciendo una consulta a la base de datos con la información a ser actualizada.		

Fuente: Autores del proyecto.

Figura 33

Diagrama de secuencia opciones elementos móvil.



Fuente: Autores del proyecto.

Iteración 2

Caso de uso web

Tabla 56

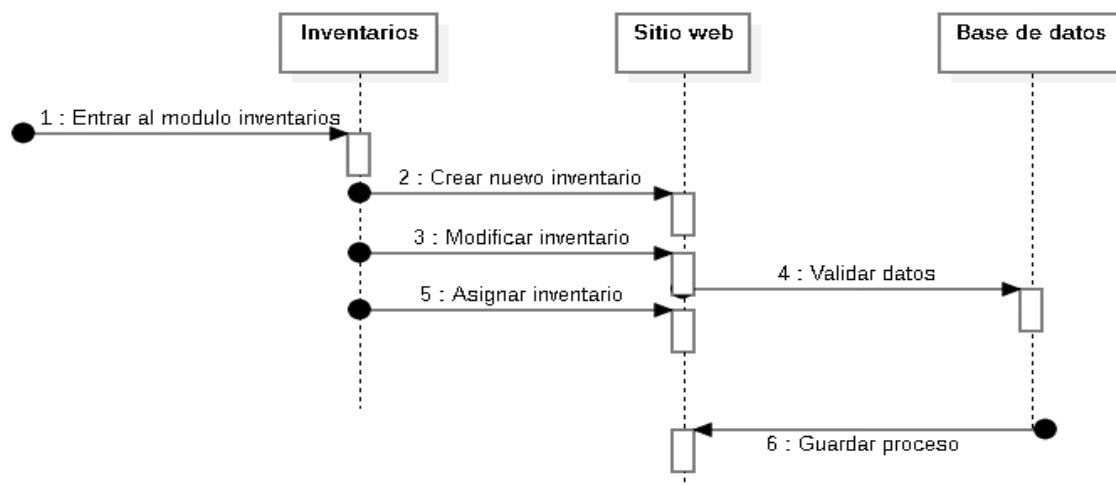
Caso de uso web: Inventarios.

Caso de Uso	Inventarios	Identificador	CU_W04
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Abrir el módulo de inventarios.		
Postcondición	Asignar elementos a el inventario.		
Descripción	Los inventarios cuentan con la opción de realizar la inserción de nuevos inventarios y modificar las dependencias a la que estos pertenecen, así mismo ser asignados a diferentes encargados disponibles en el sistema.		
Resumen	Los inventarios en el sistema cumplen el rol de separar todos los elementos en diferentes dependencias y organizarlos para la fácil obtención de la información que estos puedan proporcionar en la gestión de inventarios		

Fuente: Autores del proyecto.

Figura 34

Diagrama de secuencia inventario sitio web.



Fuente: Autores del proyecto.

Tabla 57

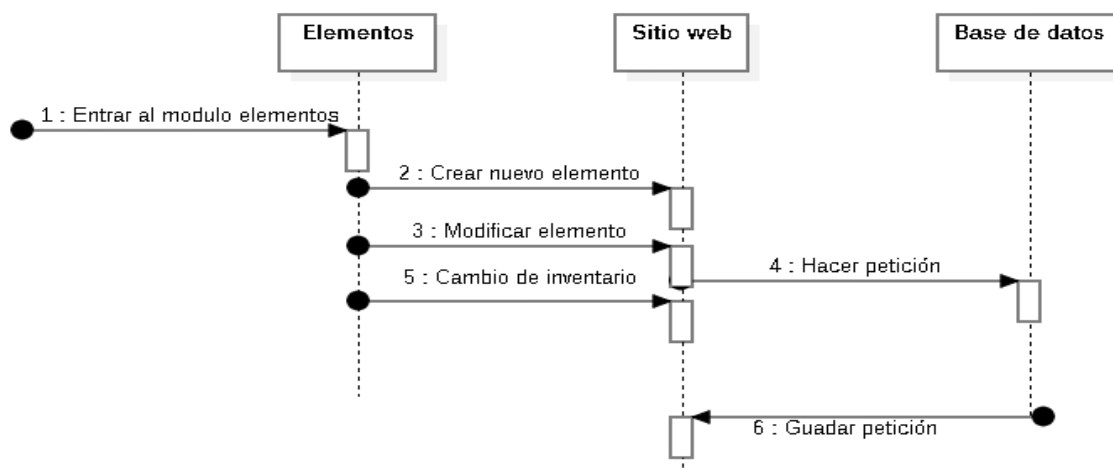
Caso de uso web: Opciones elementos.

Caso de Uso	Opciones Elementos	Identificador	CU_W05
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar inventario.		
Postcondición	Mostrar datos de los elementos almacenados.		
Descripción	El administrador cuenta con la opción de añadir nuevos elementos al inventario de cada una de las dependencias, así mismo modificar la información de estos desde los datos hasta la imagen, finalmente el elemento puede darse de baja del sistema con la opción de cambio de estado.		
Resumen	Todos los elementos del inventario podrán ser modificados desde este módulo en cada uno de los inventarios donde estos se encuentren, haciendo una consulta a la base de datos con la información a ser actualizada.		

Fuente: Autores del proyecto.

Figura 35

Diagrama de secuencia elementos sitio web.



Fuente: Autores del proyecto.

Tabla 58

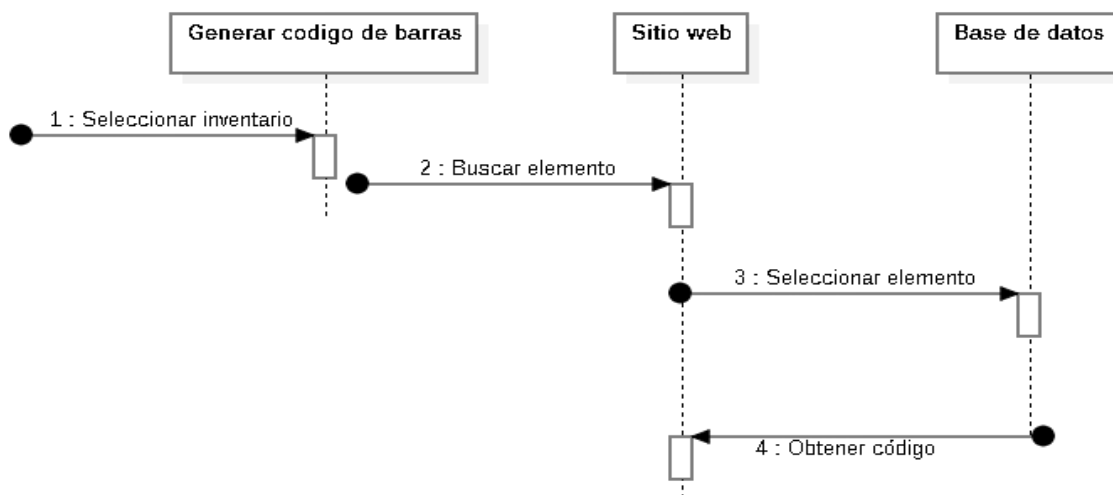
Caso de uso web: Generar código de barras.

Caso de Uso	Generar código de barras	Identificador	CU_W06
Actores	Administrador y Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar elementos para generar código de barras.		
Postcondición	Descargar código de barras.		
Descripción	Los actores del sistema cuentan con la opción de generar código de barras para los elementos de los inventarios que no cuenten con etiqueta de código de barras.		
Resumen	La generación de código de barras permite actualizar las etiquetas a los elementos que físicamente no cuentan con ello.		

Fuente: Autores del proyecto.

Figura 36

Diagrama de secuencia generar código de barras sitio web.



Fuente: Autores del proyecto.

Tabla 59

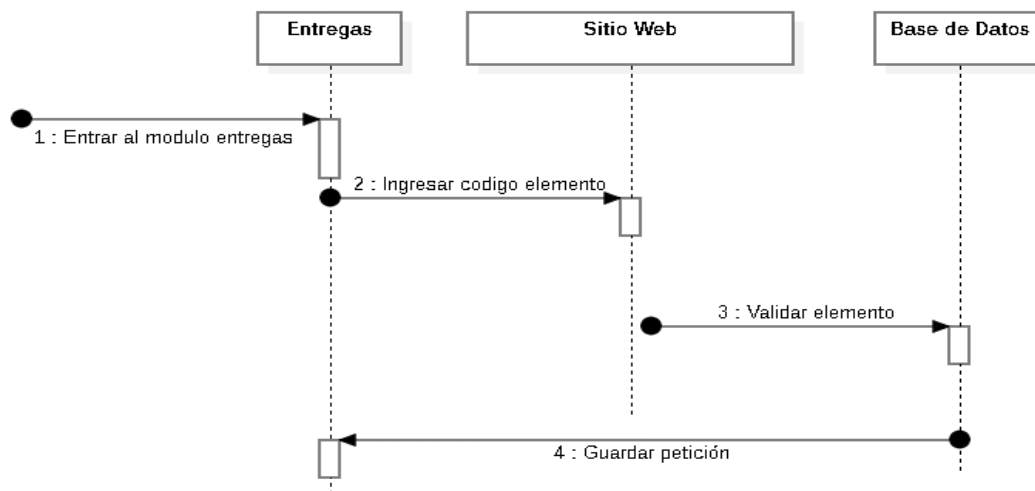
Caso de uso web: Entrega de inventarios.

Caso de Uso	Entrega de Inventarios	Identificador	CU_W07
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Ingresar al módulo de entregas de inventario.		
Postcondición	Validar elementos chequeados y no chequeados, generar pdf y almacenar en base de datos.		
Descripción	La entrega inicial y final de los inventarios permite al usuario administrador verificar que todos los elementos se encuentren en el inventario y poder determinar cuáles no, de esta manera generar un resumen del proceso en pdf.		
Resumen			

Fuente: Autores del proyecto.

Figura 37

Diagrama de secuencia entregas sitio web.



Fuente: Autores del proyecto.

Tabla 60

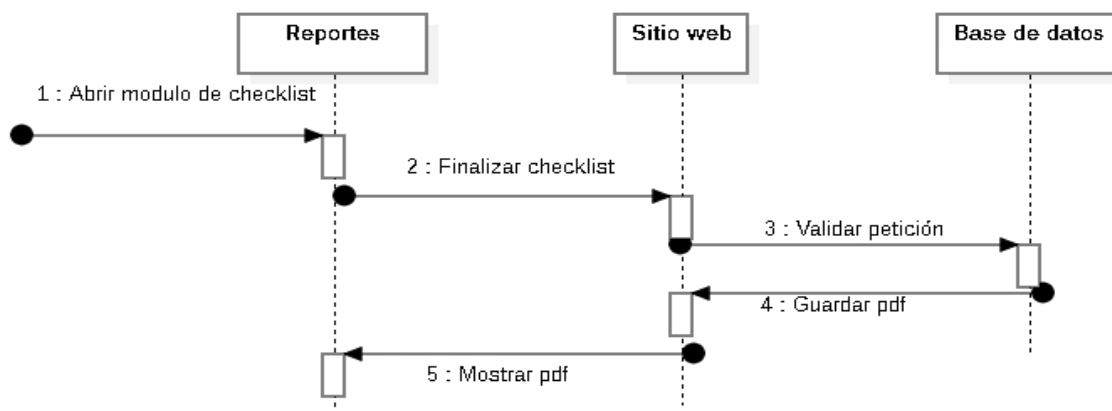
Caso de uso web: Reportes.

Caso de Uso	Reportes	Identificador	CU_W08
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Finalizar entrega de inventarios.		
Postcondición	Guardar reporte en formato pdf en la base de datos.		
Descripción	El administrador cuenta con la opción de generar reportes: Cuando se finalicen las entregas de inventario.		
Resumen			

Fuente: Autores del proyecto.

Figura 38

Diagrama de secuencia reportes sitio web.



Fuente: Autores del proyecto.

Caso de uso móvil

Tabla 61

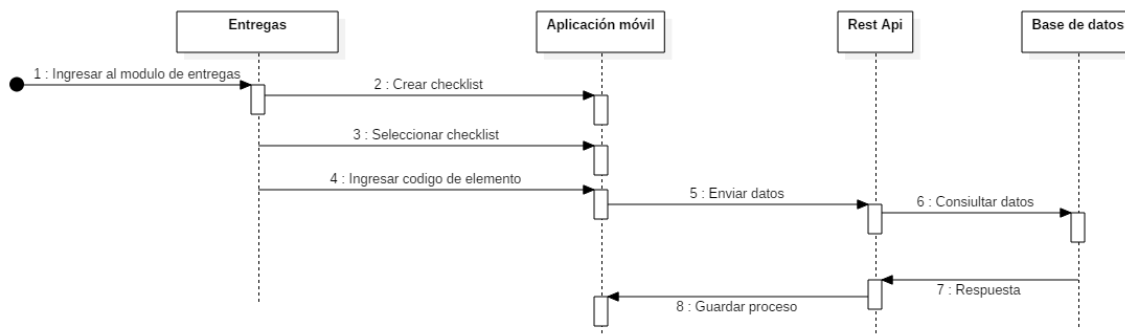
Caso de uso aplicativo móvil: entregas de inventario.

Caso de Uso	Entregas de inventario	Identificador	CU_M04
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Ingresar a módulo de checklist activos.		
Postcondición	Validar elementos chequeados y no chequeados.		
Descripción	La entrega de inventarios en la aplicación móvil permite que el proceso de entregas se agilice utilizando el lector de código de barras por medio de la cámara del dispositivo.		
Resumen	Para el proceso de chequeo de los elementos se debe tener en cuenta la validación de que el elemento exista y pertenezca a dicho inventario seleccionado.		

Fuente: Autores del proyecto.

Figura 39

Diagrama de secuencia entregas aplicación móvil.



Fuente: Autores del proyecto.

Tabla 62

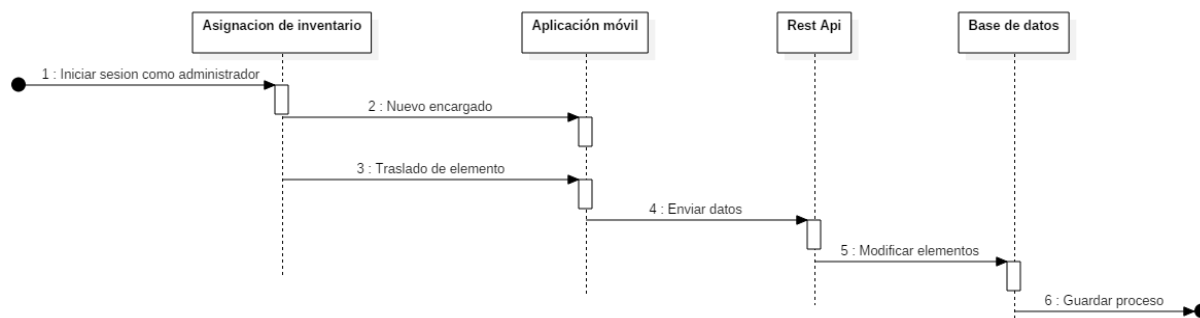
Caso de uso aplicativo móvil: Asignación de inventarios.

Caso de Uso	Asignación de inventarios	Identificador	CU_M05
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar opción de cambio de encargado.		
Postcondición	Mostrar formulario.		
Descripción	El usuario administrador contará con las dos opciones de asignación de inventarios por traslado de elemento y cambio de encargado.		
Resumen	Para la asignación de inventarios por cambio de encargado se mostrará la lista de inventarios del sistema y se seleccionará el inventario enseguida se asignará el nuevo encargado, para la opción de traslado de elemento de inventario debe tener por lo mínimo un elemento.		

Fuente: Autores del proyecto.

Figura 40

Diagrama de secuencia asignación de inventarios aplicación móvil.



Fuente: Autores del proyecto.

Iteración 3

Caso de uso web

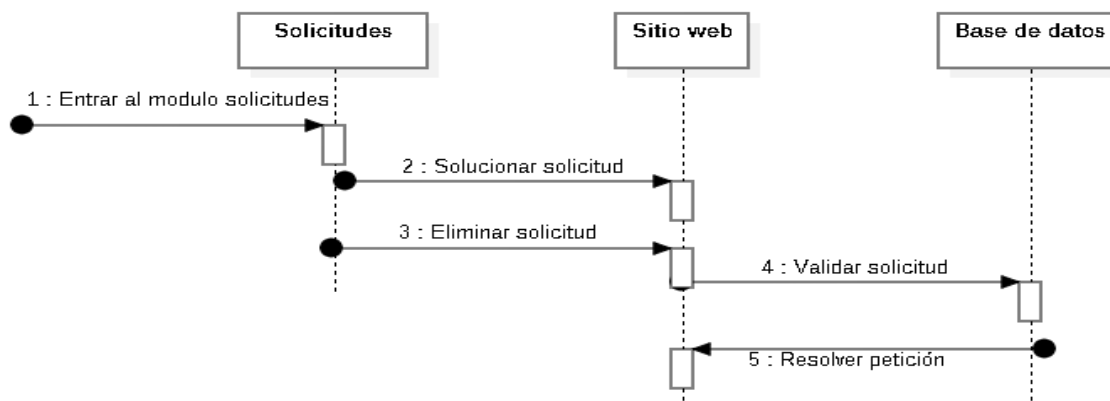
Tabla 63

Caso de uso web: Solicitudes administrador.

Caso de Uso	Solicitudes	Identificador	CU_W09
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Ingresar al módulo solicitudes.		
Postcondición	Mostrar lista de solicitudes.		
Descripción	El usuario administrador visualiza las solicitudes realizadas por el usuario dependencia y realiza la acción conveniente a esta.		
Resumen	El usuario administrador da respuesta o elimina las solicitudes generadas por el usuario administrador.		

Fuente: Autores del proyecto.

Figura 41
Diagrama de secuencia solicitudes administrador sitio web.



Fuente: Autores del proyecto.

Tabla 64

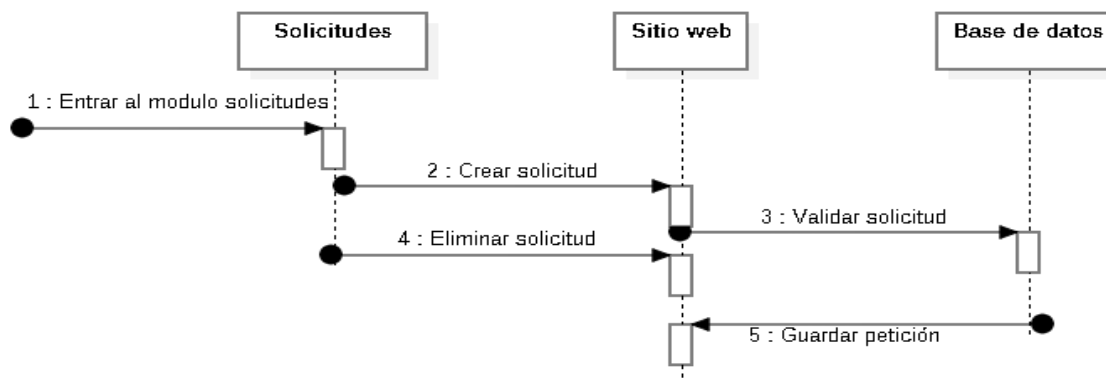
Caso de uso web: Solicitudes usuario dependencia.

Caso de Uso	Solicitudes	Identificador	CU_W10
Actor	Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	Ingresar al módulo solicitudes.		
Postcondición	Enviar solicitud a administrador		
Descripción	El usuario dependencia crea una solicitud sobre la información a cambiar en el estado de uso o lugar físico del elemento		
Resumen	Las solicitudes son creadas por el usuario dependencia para que posteriormente sean solucionadas o eliminadas por el usuario administrador.		

Fuente: Autores del proyecto.

Figura 42

Diagrama de secuencia solicitudes usuario dependencia sitio web.



Fuente: Autores del proyecto.

Tabla 65

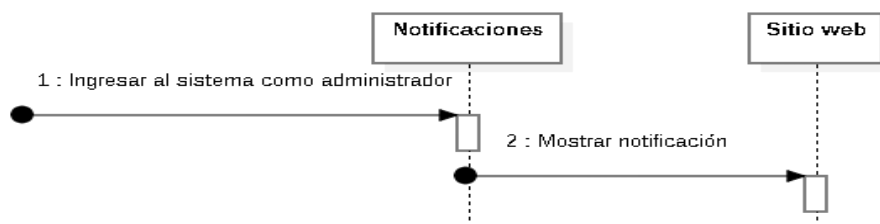
Caso de uso web: Notificaciones.

Caso de Uso	Notificaciones	Identificador	CU_W11
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Ingresar a el sistema con el usuario administrador.		
Postcondición	Ninguna		
Descripción	El sistema muestra en la página principal las notificaciones de las solicitudes creadas por el usuario dependencia que aún están pendientes por resolver.		
Resumen			

Fuente: Autores del proyecto.

Figura 43

Diagrama de secuencia notificaciones sitio web.



Fuente: Autores del proyecto.

Caso de uso móvil

Tabla 66

Caso de uso aplicativo móvil: Solicitudes administrador.

Caso de Uso	Solicitudes	Identificador	CU_M06
Actores	Usuario dependencia		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar el elemento a realizar solicitud.		
Postcondición	Mostrar lista de solicitudes realizada.		
Descripción	El usuario dependencia crea una solicitud sobre la información a cambiar en el estado de uso o lugar físico del elemento.		
Resumen	Las solicitudes son creadas por el usuario dependencia para que posteriormente sean solucionadas o eliminadas por el usuario administrador.		

Fuente: Autores del proyecto.

Tabla 67

Caso de uso aplicativo móvil: Solicitudes usuario dependencia.

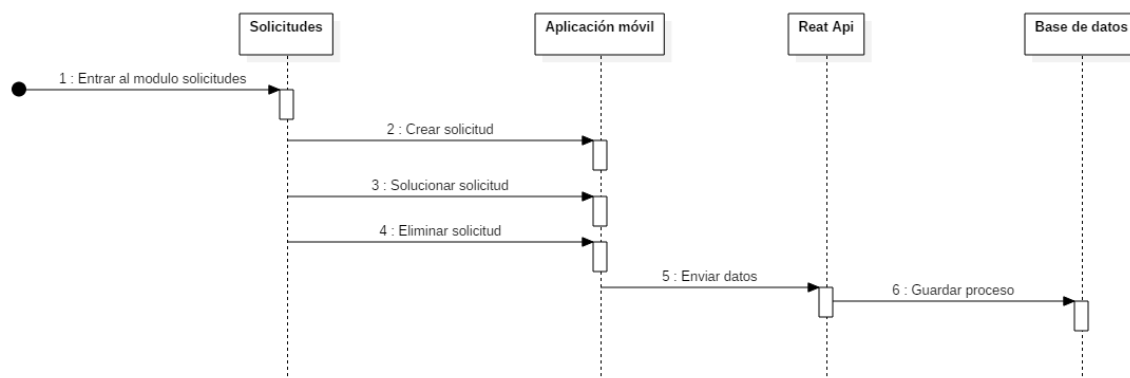
Caso de Uso	Solicitudes	Identificador	CU_M07
Actor	Administrador		
Tipo	Primario	Referencia	Ninguna
Precondición	Seleccionar el tipo de solicitud a revisar.		
Postcondición	Mostrar lista de solicitudes realizada.		

Descripción	El usuario administrador visualiza las solicitudes realizadas por usuario dependencia y realiza la acción conveniente a esta.
Resumen	El usuario administrador da respuesta o elimina las solicitudes generadas por el usuario administrador.

Fuente: Autores del proyecto.

Figura 44

Diagrama de secuencia solicitudes aplicación móvil.



Fuente: Autores del proyecto.

Tabla 68

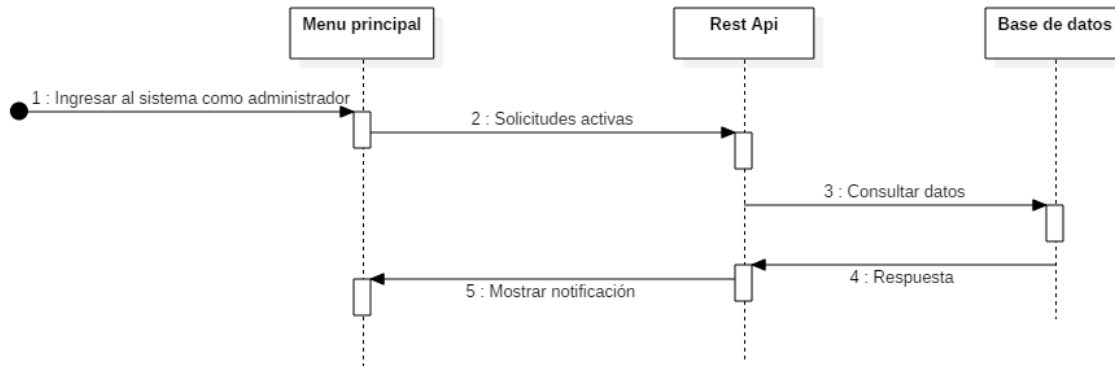
Caso de uso aplicativo móvil: Notificaciones.

Caso de Uso	Notificaciones	Identificador	CU_M08
Actor		Administrador	
Tipo	Primario	Referencia	Ninguna
Precondición	Tener solicitudes pendientes.		
Postcondición	Ninguna.		
Descripción	El usuario administrador al ingresar al sistema recibirá notificaciones si tiene solicitudes pendientes a resolver, de lo contrario no		
Resumen			

Fuente: Autores del proyecto.

Figura 45

Diagrama de secuencia notificaciones aplicación móvil.

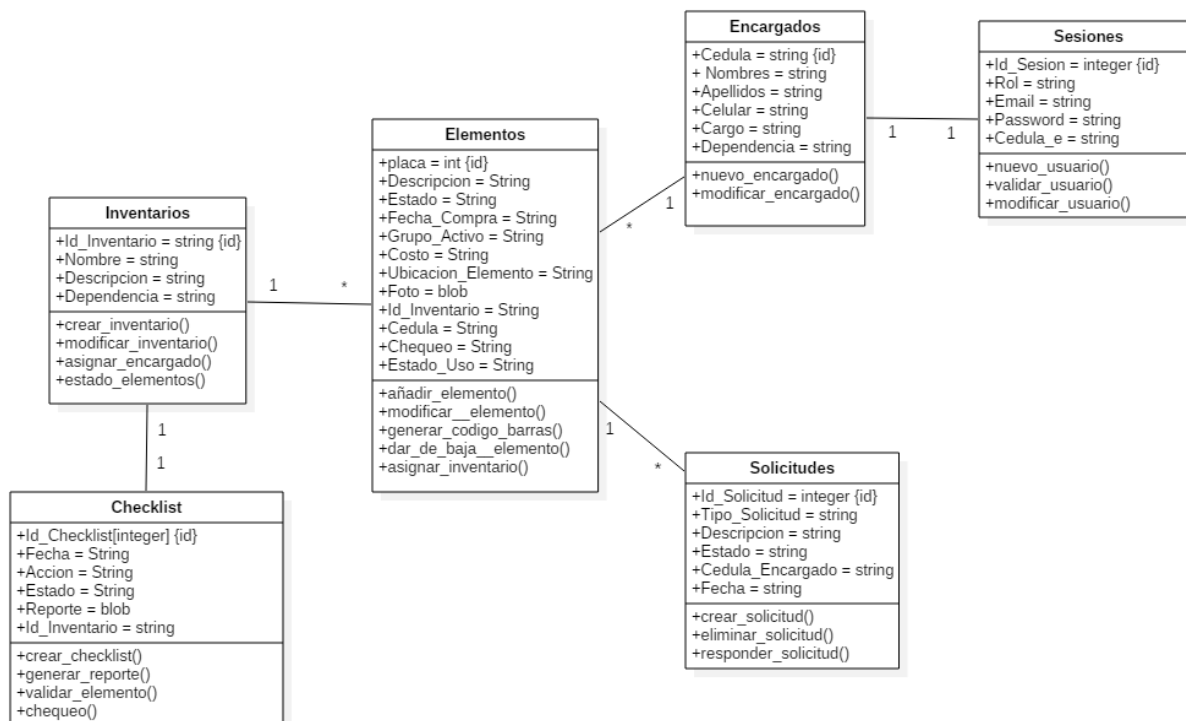


Fuente: Autores del proyecto.

10.2.6. Diagrama de clase sistema multiplataforma

Figura 46

Diagrama de clase sistema multiplataforma.



Fuente: Autores del proyecto.

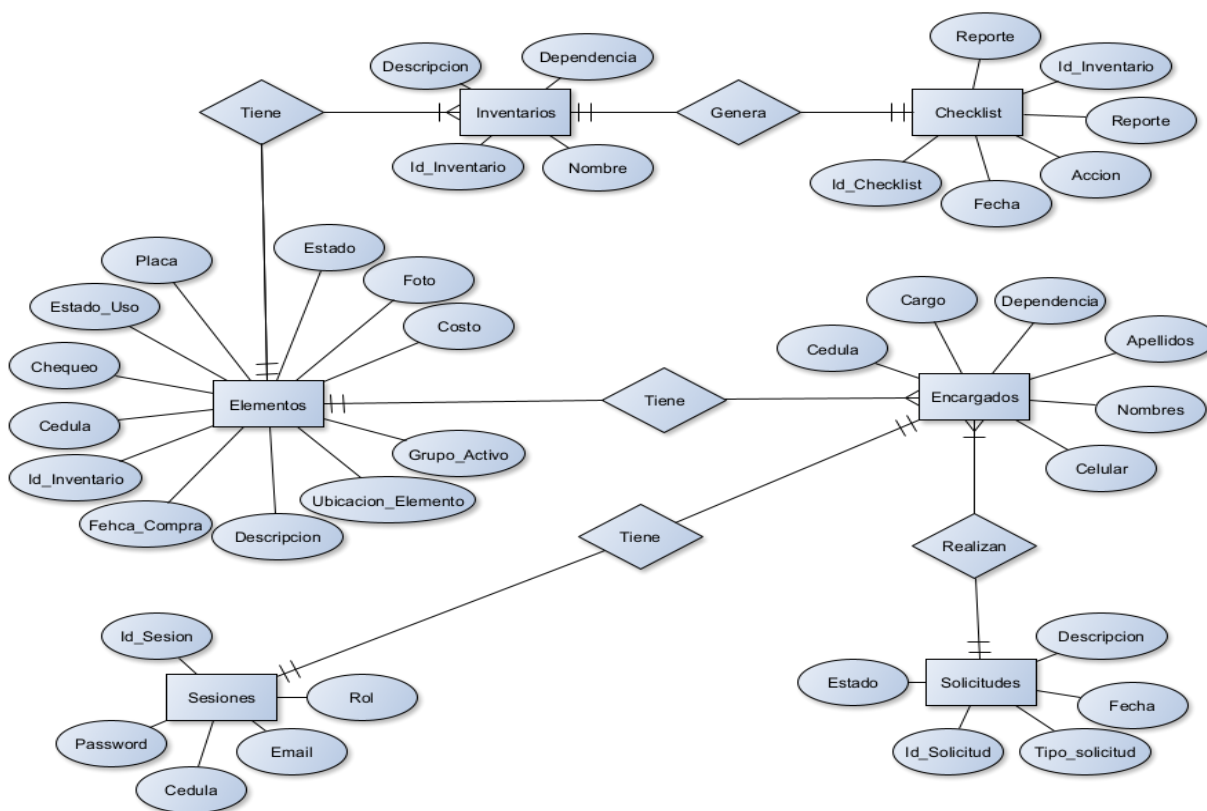
10.2.7. Base de datos

10.2.7.1. Modelos entidad relación base de datos

Este modelo es conocido como E-R el cual permite la representación de un gráfico que consta de entidades, relaciones y atributos, los cuales en unión logran dar el sentido lógico de la estructura de la información. En este esquema se evidencia el modelo de datos que permite plasmar las entidades más relevantes para la creación del sistema multiplataforma, así mismo sus relaciones y atributos que cada una de las entidades cuenta, los cuales fueron implementados en la base de datos para la organización de toda la información requerida para la gestión de inventarios.

Figura 47

Modelo entidad relación base de datos.



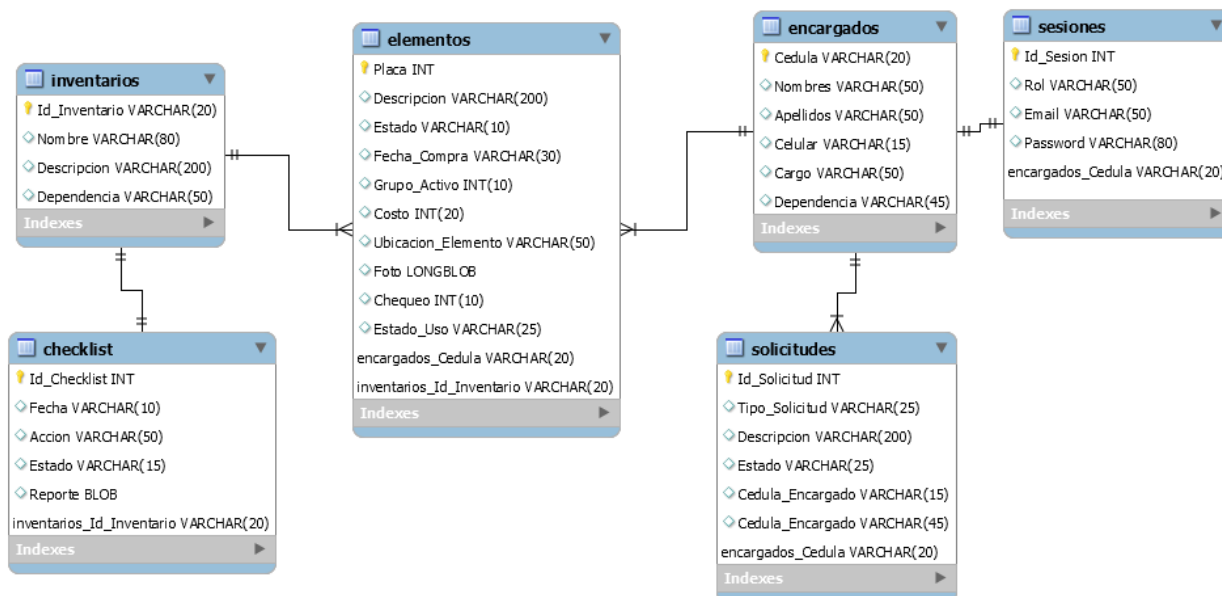
Fuente: Autores del proyecto.

10.2.7.2. Base de datos para sistema multiplataforma Udecinv

El principal aspecto del sistema multiplataforma para el funcionamiento y la obtención de una conexión en tiempo real en las dos plataformas es la base de datos relacional, teniendo como función principal almacenar todos los datos que se manejan para la gestión de inventarios, de esta manera clasificar mejor la información con el fin de que se pueda dar un buen manejo a esta, Udecinv trabaja con el gestor de bases de datos MySQL el cual es una base de datos relacional y de igual manera permite almacenar hasta 64 terabytes de información (*Número Máximo de Registros En Una Tabla de Base de Datos / Mysql, 2019.*) por cada tabla. Para la manipulación de los datos se está utilizando PhpMyAdmin el cual es una herramienta para la administración desde el servidor que dispone de una interfaz gráfica para navegar a través de esta y acceder a las opciones con las que cuenta.

Figura 48

Modelo relacional de base de datos.



Fuente: Autores del proyecto.

La base de datos del sistema multiplataforma cuenta con seis tablas, cinco de ellas son relacionadas entre sí, la tabla de sesiones es la única que no se encuentra relacionada con las demás, tomada de esta manera por parámetros de seguridad, ya que al no estar conectada con los demás brinda mayor seguridad frente un posible ataque a la información de la base de datos en donde se logra separar la información de la sesión con los datos de los inventarios de la universidad.

10.2.7.2.1. Tablas de la base de datos

- Sesiones
- Checklist
- Elementos
- Encargados
- Inventarios
- Solicitudes

Tabla sesiones

Tabla 69

Tabla sesiones.

Atributo	Descripción
Id_Sesion	Es el indicativo de cada usuario para el inicio de sesión
Rol	Rol de usuario
Email	Correo electrónico
Password	Contraseña encriptada con Bcrypt
Cedula	Llave foránea procedente de la tabla encargados

Fuente: Autores del proyecto.

Tabla de checklist

Tabla 70

Tabla checklist.

Atributo	Descripción
Id_Checklist	Identificador de la entrega
Fecha	Fecha de creación de la entrega
Acción	Tipo de entrega (Inicial, Final)
Estado	Estado de la entrega (Activo, Finalizado)
Id Inventario	Identificador primario del inventario
Reporte	Pdf con el reporte de las entregas

Fuente: Autores del proyecto.

Tabla de elementos

Tabla 71

Tabla elementos.

Atributo	Descripción
Placa	Identificador primario de cada elemento.
Descripcion	Descripción completa de las características del elemento.
Estado	Estado dentro de inventario (Activo o Inactivo) tomando inactivo como un elemento que ya no pertenece a ningún inventario.
Fecha_Compra	Fecha en la que el elemento fue comprado por la universidad de Cundinamarca.
Grupo_Activo	Dato utilizado por la universidad para la clasificación de los elementos.
Costo	Valor de compra del elemento
Ubicacion_Elemento	Ubicación física donde se encuentra el elemento ubicado.
Foto	Registro fotográfico de elemento
Id_Inventario	Identificador de inventario
Cedula	Documento de identidad del encargado del elemento
Chequeo	Número de la entrega del inventario.
Estado_uso	Estado de los elementos (Bueno, Malo, Regular, Mantenimiento)

Fuente: Autores del proyecto.

Tabla encargados

Tabla 72

Tabla encargados.

Atributo	Descripción
Cedula	Documento de identidad de los encargados
Nombres	Nombre del encargado del inventario
Apellidos	Apellido del encargado del inventario
Celular	Número de contacto del encargado
Cargo	Función administrativa en la universidad
Dependencia	Área administrativa a la que pertenece el inventario

Fuente: Autores del proyecto.

Tablas inventarios

Tabla 73

Tabla inventarios.

Atributo	Descripción
Id_Inventario	Identificador primario del inventario
Nombre	Nombre del inventario
Descripcion	Información detallada del inventario
Dependencia	Área administrativa a la que pertenece el inventario

Fuente: Autores del proyecto.

Tabla solicitudes

Tabla 74

Tabla solicitudes.

Atributo	Descripción
Id_Solicitud	Identificador primario de la solicitud
Tipo_Solicitud	Solicitud para el cambio de lugar físico o cambio de estado de uso (Bueno, Malo, Regular, Mantenimiento)
Cedula_Encargado	Documento de identidad del encargado que realiza la solicitud
Fecha	Fecha de creación de la solicitud

Fuente: Autores del proyecto.

10.3. Codificación

Una de las fases más importantes de la metodología implementada es la codificación, ya que es la fase donde se realiza la producción de las fases anteriores como la planificación y el diseño, en esta fase se contempla el desarrollo de las historias de usuario en su respectiva iteración asignada dando como resultado final los módulos que serán presentados al cliente al final de cada iteración.

10.3.1. Integraciones frecuentes

En el sistema multiplataforma al finalizar o realizar una modificación de código se socializo los cambios ejecutados en cualquiera de las dos plataformas, esto dando como beneficio que ninguno de los integrantes de desarrollo trabaje en versiones obsoletas o solucione una problemática ya resuelta.

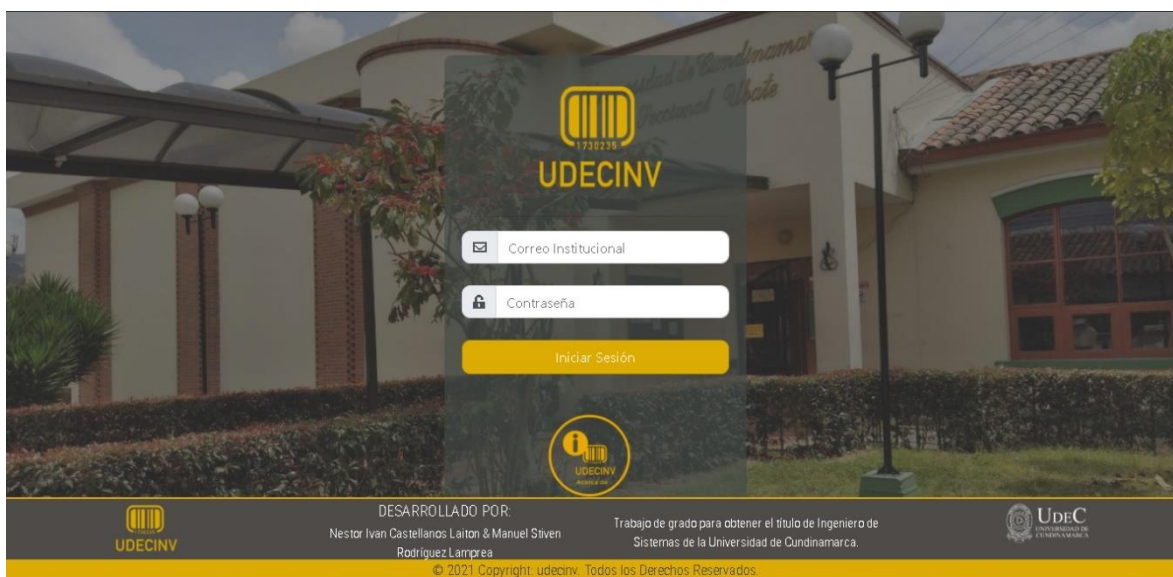
10.3.2. Codificación iteración 1

Codificación de sistema web

- Ingreso al sistema

Figura 49

Login sitio web.



Fuente: Autores del proyecto.

Esta es la página inicial del sistema en sitio web, en donde se puede apreciar la información del sistema mostrada en un footer el cual tiene el icono del sistema, el logo de la universidad de Cundinamarca y la información de los desarrolladores, así mismo el formulario que valida los 2 tipos de usuario que existen en el sistema (Usuario administrador, Usuario dependencia) por medio del correo institucional y una contraseña con mínimo 8 caracteres para una mayor seguridad de las sesiones.

Código en nodejs que permite validar los usuarios para el ingreso al sistema mediante la comparación de contraseñas encriptadas.

Figura 50

Código de validación de usuarios sitio web.

```
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;

const helpers = require('../funciones/contra');
const pool = require('../basedatos');

passport.serializeUser((user, done) => {
  done(null, user.Id_Sesion);
});

passport.deserializeUser(async (id, done) => {
  const rows = await pool.query('SELECT * FROM sesiones WHERE Id_Sesion = ?', [id]);
  done(null, rows[0]);
});

passport.use('local-signin', new LocalStrategy({
  usernameField: 'email',
  passwordField: 'contra',
  passReqToCallback: true
}, async (req, email, contra, done) => {

  const rows = await pool.query('SELECT * FROM sesiones WHERE Email = ?', [email]);
  if (rows.length > 0) {
    const user = rows[0];
    //console.log(user)
    const validPassword = await helpers.matchPassword(contra, user.Password)
    if (validPassword) {
      done(null, user, req.flash('success', 'bienvenido ' + user.Usuario));
    } else {
      done(null, false, req.flash('message2', 'contraseña incorrecta'));
    }
  } else {
    return done(null, false, req.flash('message', 'el usuario no existe.'));
  }
}));
```

Fuente: Autores del proyecto.

Conexión a base de datos mysql.

Figura 51

Código conexión a la base de datos del sistema multiplataforma.

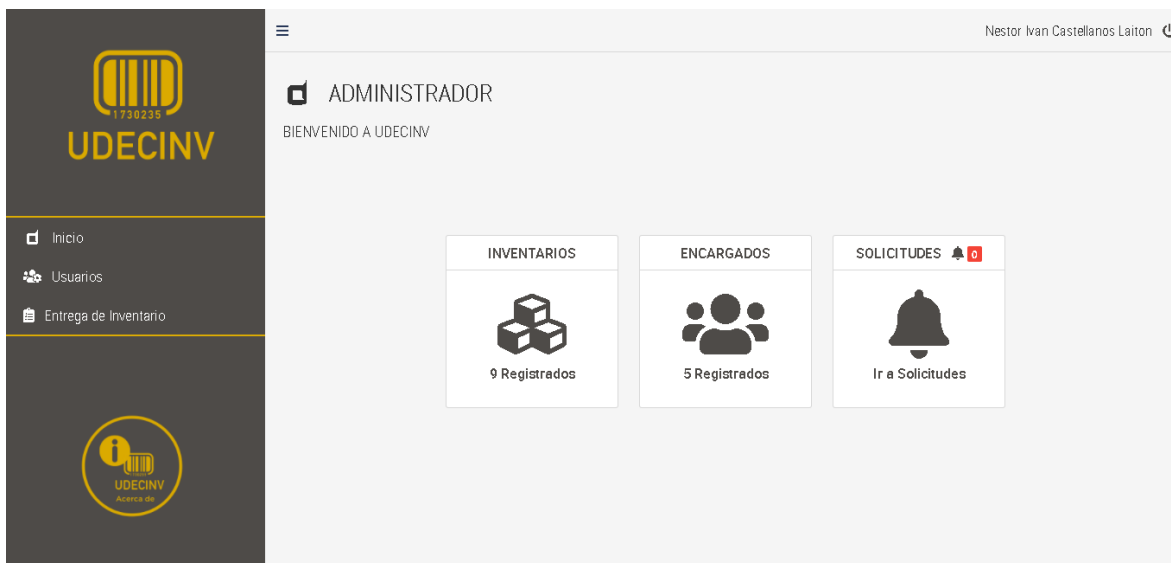
```
module.exports = {  
  
  database: {  
    //connectionLimit: 10,  
    host: 'host25.latinamericahosting.com',  
    user: 'molinodo_wp1409',  
    password: 'zqE~u-UaX9P',  
    database: 'molinodo_udecinvo'  
  }  
}
```

Fuente: Autores del proyecto.

Figura 52

Menú principal de usuario administrador sitio web.

Página principal usuario administrador



Fuente: Autores del proyecto.

En la página principal del usuario administrador se podrá encontrar los módulos de usuarios, encargados, inventario, solicitudes y entregas de inventarios, así mismo la opción de cierre de sesión para salir de forma correcta del sistema.

Figura 53

Menú principal de usuario dependencia sitio web.

Página principal usuario dependencia



Fuente: Autores del proyecto.

La página principal del usuario dependencia permite la visualización de los inventarios asignados a el encargado que tiene la sesión activa, así mismo puede ver las solicitudes que han sido generadas por ese usuario.

Código de lenguaje estructurado de html que permite crear la estructura principal de la interfaz que maneja todo el sistema.

Figura 54

Código de lenguaje estructurado de html.

```

<!DOCTYPE html>
<html lang="es">
  <!-- head -->
  <head>
    <title>Administrador</title>
    <% include ../partes/heads %>
  </head>
  <!-- Fin head -->
  <!-- body -->
  <body>
    <!-- contenedor principal -->
    <main class="full-box main-container">
      <section class="full-box nav-lateral">
        <div class="full-box nav-lateral-bg show-nav-lateral"></div>
        <div class="full-box nav-lateral-content">
          <% include ../partes/logo %>
          <div class="full-box nav-lateral-bar"></div>
          <nav class="full-box nav-lateral-menu">
            <ul>
              <li>
                <a href="/administrador" ><i class="fab fa-dashcube fa-fw"></i> &nbsp;&nbsp;&nbsp; Inicio</a>
              </li>
              <li>
                <a href="/usuarios"><i class="fas fa-users-cog"></i> &nbsp;&nbsp;&nbsp; Usuarios</a>
              </li>
              <li>
                <a href="/checklist"><i class="fas fa-clipboard-list"></i> &nbsp;&nbsp;&nbsp; Entrega de Inventario</a>
              </li>
            </ul>
          </nav>
          <div class="full-box nav-lateral-bar"></div>
          <br>
          <% include ../partes/acercade %>
        </div>
      </section>

```

- Búsqueda de elementos

Búsquedas en usuario administrador

Figura 55

Ventana de búsquedas usuario administrador sitio web.

The screenshot shows the UDECINV web application interface. On the left is a dark sidebar with the UDECINV logo and navigation options: 'Inventarios', 'Buscar Elemento', and 'Opciones'. The main content area features a search modal titled 'Buscar Elemento' with a search input field containing '51'. Below the modal is a table with the following data:

#	Placa	Descripción	Ubicación Del Elemento	Ver Elemento
1	51387	MESA CENTRAL DE LABORATORIO 3900X1500X900 MM	Cafeteria	Ver
2	51388	MESA CENTRAL DE LABORATORIO 3900X1500X900 MM	LABORATORIO DE FISICA	Ver
3	51391	MESA MURAL DE LABORATORIO 5600X750X900 MM	LABORATORIO DE FISICA	Ver
4	51393	MESA ESPECIAL PARA BALANZA ANALITICA 900X900X900 MM	LABORATORIO DE FISICA	Ver
5	51396	ARMARIO DE ALMACENAMIENTO ESPECIAL PARA LIQUIDOS INFLAMABLES	LABORATORIO DE FISICA	Ver
51	3006025	SILLAS AUXILIARES SIN BRAZOS LINEA 800, ESTRUCTURA EN MADERA ASIENTOS Y ESPALDAR TAPIZADOS PAPO COLOR MOSTAZA	LABORATORIO DE FISICA	Ver

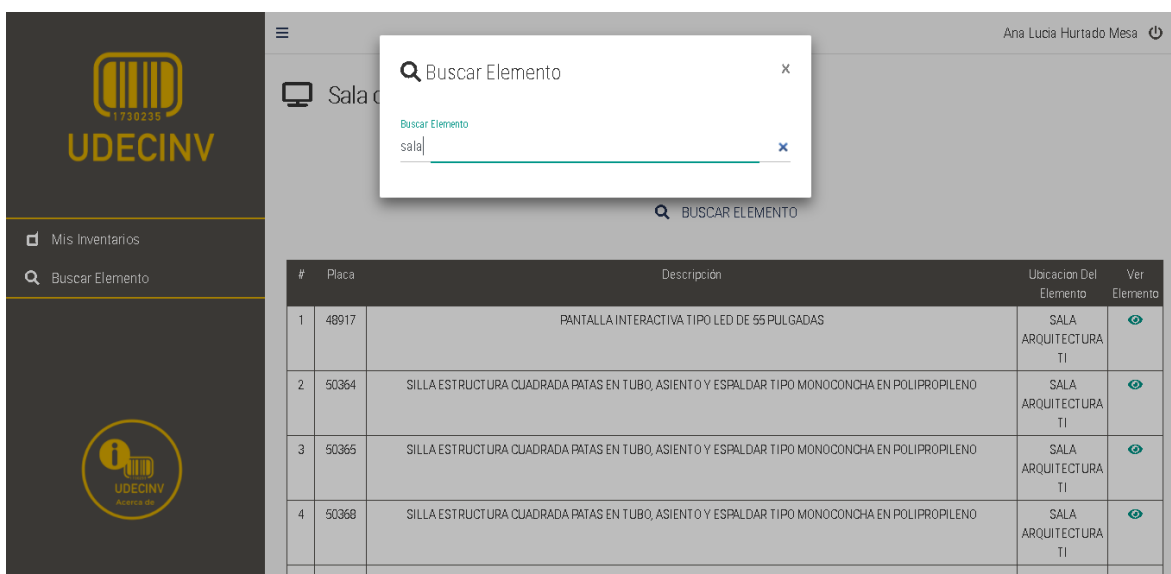
Fuente: Autores del proyecto.

Las búsquedas en el usuario administrador permiten buscar los elementos por cualquier atributo que sea ingresado en el buscador, la placa, la descripción o el lugar físico donde se encuentra el elemento.

Búsquedas en usuario dependencia.

Figura 56

Ventana de búsquedas usuario dependencia sitio web.



The screenshot shows the UDECINV web application interface. A search modal is open, displaying the text 'Buscar Elemento' and a search input field containing 'sala'. Below the modal, a table lists inventory items. The table has the following columns: #, Placa, Descripción, Ubicación Del Elemento, and Ver Elemento. The data rows are as follows:

#	Placa	Descripción	Ubicación Del Elemento	Ver Elemento
1	48917	PANTALLA INTERACTIVA TIPO LED DE 55 PULGADAS	SALA ARQUITECTURA TI	
2	50364	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	
3	50365	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	
4	50368	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	
5	50370	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA	

Fuente: Autores del proyecto.

Las búsquedas en el usuario dependencia permiten buscar los elementos de los inventarios asignados a el encargado por cualquier atributo que sea ingresado en el buscador, la placa, la descripción o el lugar físico donde se encuentra el elemento.

Algoritmo en JavaScript que permite realizar el filtrado de las búsquedas

Figura 57

Algoritmo en JavaScript para filtros en sitio web.

```

<script>
$(document).ready(function(){
  $("#buscar").on("keyup", function() {
    var value = $(this).val().toLowerCase();
    $("#tabla tr").filter(function() {
      $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
    });
  });
});
</script>

```

Fuente: Autores del proyecto.

- Opciones de elementos

Ver elementos

Figura 58

Ventana para ver información de elementos en sitio web.

The screenshot displays a web application interface for inventory management. On the left is a dark sidebar with the UDECINV logo (1728235) and navigation options: 'Laboratorio de física' and 'Opciones'. The main content area shows details for 'BOMBA DE VACIO VOLMO'. It includes fields for 'Codigo Inventario: E-103', 'Inventario: Laboratorio de física', and 'Encargado: Ana Lucia Hurtado Mesa'. Below this is a table with the following data:

Placa	Estado de Uso	Ubicacion Del Elemento
3005205	Bueno	LABORATORIO DE FISICA
Fecha de Compra	Grupo Activo	Costo
30/12/1996	166002	\$ 60000

Below the table is a photograph of the vacuum pump. At the bottom of the main area are two buttons: 'MODIFICAR' and 'CÓDIGO DE BARRAS'.

Fuente: Autores del proyecto.

Modificar elementos

Figura 59

Ventana para modificar elementos.

The screenshot shows the 'Modificar Elemento' window with the following data:

Placa	51387	Descripcion Elemento	MESA CENTRAL DE LABORATORIO 3900X1500X900 MM
Estado	Activo	Ubicacion del Elemento	Cafeteria
Estado De Uso	Bueno		

Fuente: Autores del proyecto.

Añadir elementos

Figura 60

Ventana para agregar nuevo elemento.

The screenshot shows the 'Nuevo Elemento' window with the following data:

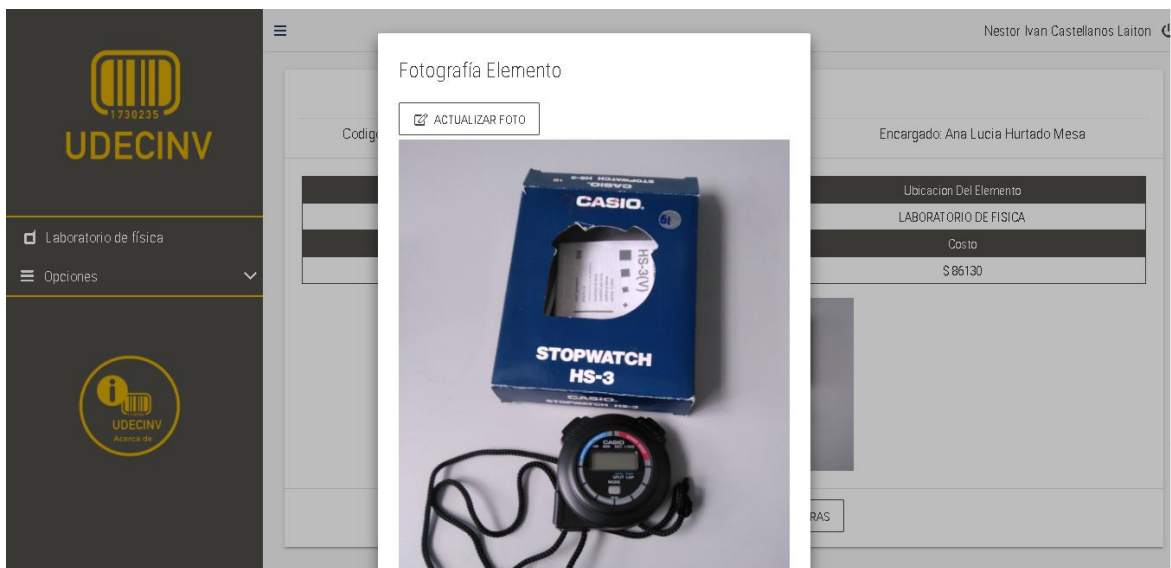
Placa		Descripcion Elemento	
Fecha de Compra	dd/mm/aaaa	Grupo Activo	
Costo		Nombre Encargado
Ubicación Elemento		Estado De Uso

Fuente: Autores del proyecto.

Actualizar foto de los elementos

Figura 61

Ventana para actualizar foto del elemento.



Fuente: Autores del proyecto.

Las opciones disponibles para los elementos son las de añadir elemento, modificar elemento, ver información del elemento y modificar la fotografía de este, estas opciones se basan en el uso de formularios que le permiten al usuario ingresar la información de una manera intuitiva y sencilla.

Figura 62

Código para realizar consultas a la base de datos.

Código en nodejs que permite realizar consultas a la base de datos mysql.

```

const bcrypt = require('bcryptjs');
const pool = require('../basedatos');
//Pagina principal administrador

controller.administrador = (req, res) =>{

pool.query('SELECT * FROM inventarios', (err, datos) => {
  ninven = datos.length;

pool.query('SELECT * FROM encargados', (err, datos2) => {
  nencar = datos2.length;

pool.query('SELECT * FROM elementos WHERE Estado = ?', ['Inactivo'], (err, datos4) => {
  nelemb = datos4.length;

pool.query('SELECT COUNT(*) as total from solicitudes WHERE Estado = ?',['Activo'], (err, cant) => {
  var tot = cant[0].total;

  if (err) {
    res.json(err);
  }
  res.render('administrador/administrador', {
    ninven: ninven , nencar:nencar, nelemb:nelemb, stot:tot
  });
});
});
});
});
});
};

```

Fuente: Autores del proyecto.

- Encargados

Figura 63

Ventana encargados sitio web.

Cedula	Nombres	Apellidos	Celular	Email	Cargo	Modificar
100200200	Julian	Cabezas Rodriguez	312525202	j@hotmail.com	veterinario	
100200300	Fernando	Pachon	3112553696	n@hotmail.com	veterinario	
	Nestor Ivan	Castellanos Laiton			Ingeniero de soporte	
	Manuel	rodriguez			profesor	
	Ana Lucia	Hurtado Mesa			Direccion del programa ingenieria de sistemas	

Fuente: Autores del proyecto.

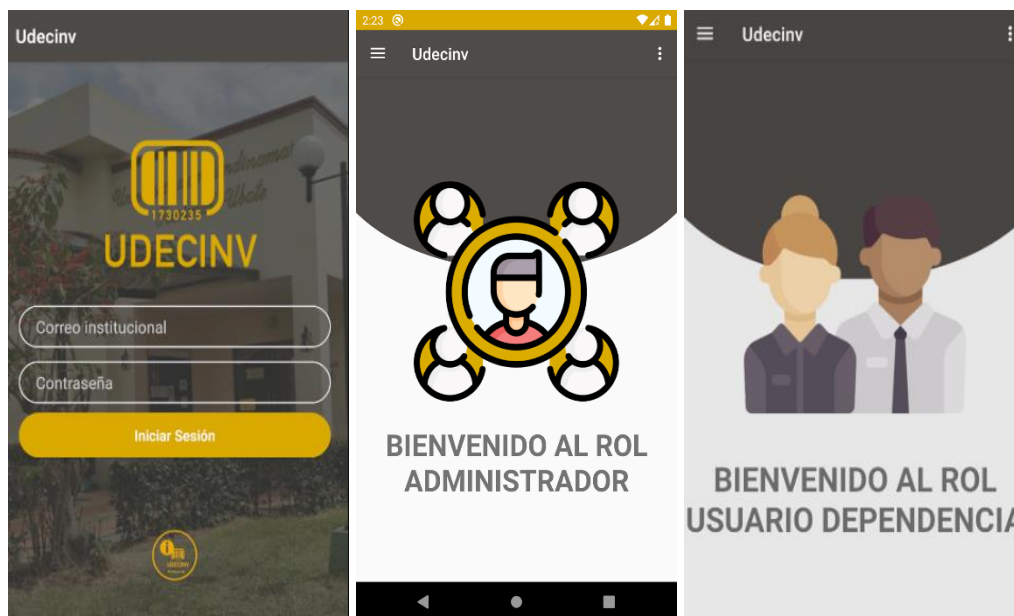
En el módulo encargados se pueden modificar los encargados, ingresar nuevos y ver la cantidad de registros desde el rol administrador.

Codificación de sistema móvil Android

- Ingreso a la aplicación móvil.

Figura 64

Ventana de ingreso aplicación móvil.

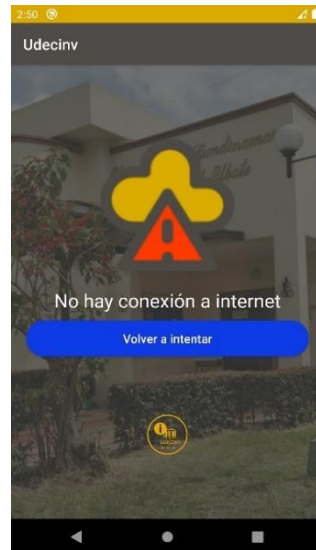


Fuente: Autores del proyecto.

La pantalla inicial de la aplicación Udecinv cuenta dos campos para acceder al sistema el primero será ingresar el correo institucional y el segundo se para ingresar la contraseña, para acceder al sistema se debe ingresar los datos y dar en el botón de iniciar sesión, este validará la información ingresada por usuario, de esta manera lo re direccionará a la venta principal de cada usuario.

Figura 65

Ventana advertencia de conexión a internet.



Fuente: Autores del proyecto.

Para el funcionamiento del sistema multiplataforma deberá de contar con acceso a internet sea por wifi o plan de datos, si el sistema no cuenta conexión a internet se mostrará una ventana con una advertencia.

Figura 66

Ventana de información sobre el aplicativo móvil.



Fuente: Autores del proyecto.

La pantalla principal del sistema cuenta con un botón de acerca de, este mostrará información relevante de la información de la aplicación y de los desarrolladores.

Código de php por el cual permite realizar consulta a la base de datos y verificar si los datos ingresados por el usuario son existentes, También la verificación de las contraseñas encriptadas.

Figura 67

Código de validación de usuarios en móvil.

```

<?PHP
include "Conexion.php";

mysqli_set_charset($conexion, "utf8");

    if(isset($_GET["Email"]) && isset($_GET["Password"])){
        $Email=$_GET['Email'];
        $contra=$_GET['Password'];
        $json=array();

        $sql = mysqli_query($conexion,"SELECT * FROM sesiones where Email = '$Email' ");
        if($vec=mysqli_fetch_array($sql)){
            if(password_verify($contra,$vec['Password'])){

                $result["Rol"]=$vec['Rol'];
                $result["Email"]=$vec['Email'];
                $json['sesiones'][]=$result;
            }
        }else {
            $result["Rol"]="No registrado";
            $result["Id"]="No";
            $json ['sesiones'][]=$result;
        }

        if($json==null){
            $result["Rol"]="No";
            $result["Id"]="No";
            $json ['sesiones'][]=$result;
        }
    }

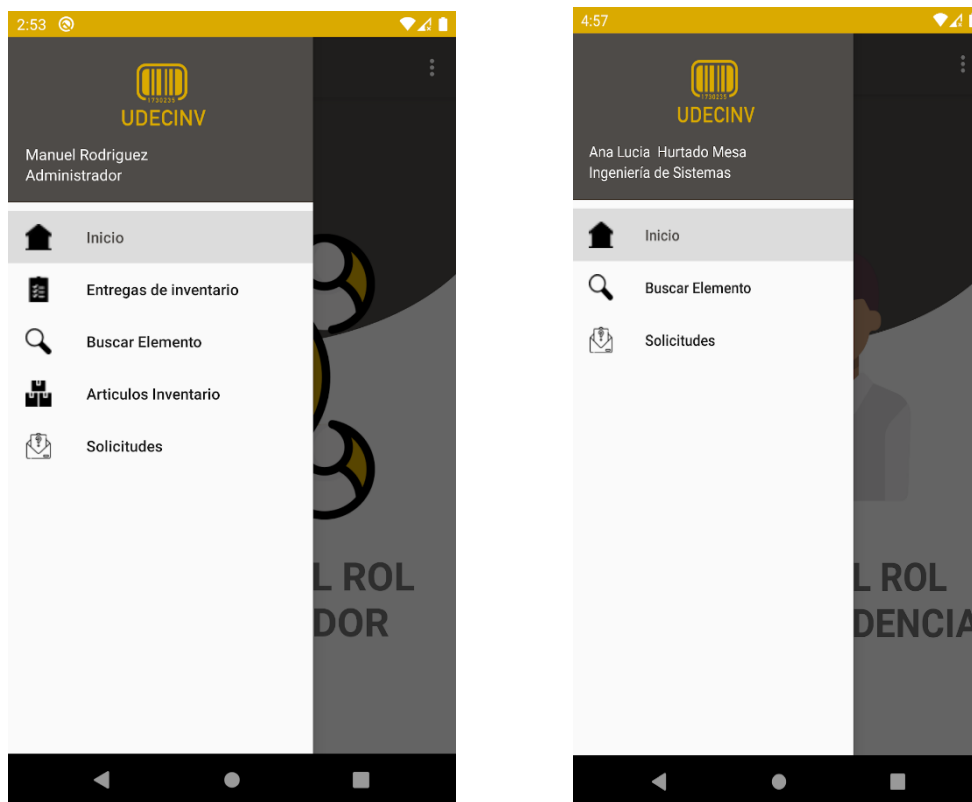
```

Fuente: Autores del proyecto.

Figura 68

Ventana de menú lateral aplicación móvil.

- Menú lateral de opciones



Fuente: Autores del proyecto.

La aplicación Udecinv cuenta con un menú desplegable al lado izquierdo donde se puede visualizar la información del usuario y las distintas opciones con las que puede interactuar el usuario dependiendo su rol.

- Búsqueda de elementos

Figura 69

Ventana de menú lateral aplicación móvil.

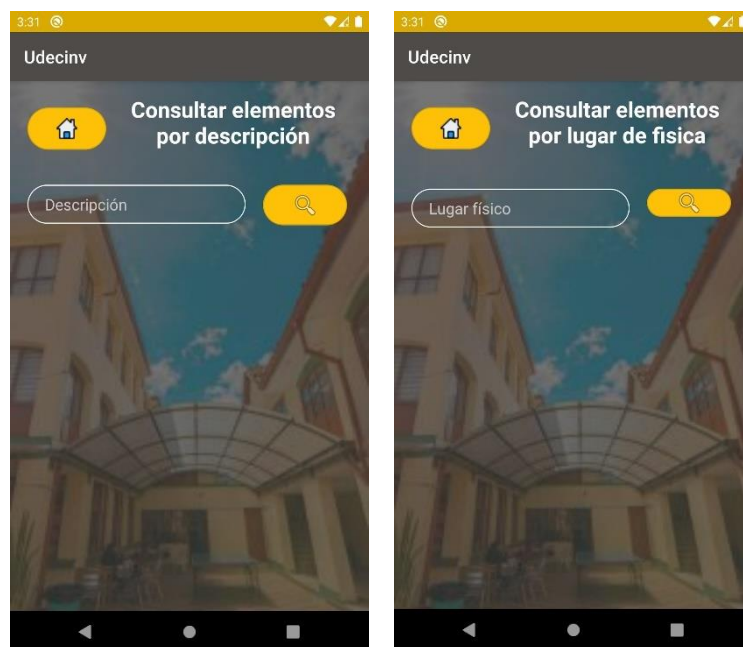


Fuente: Autores del proyecto.

En la pantalla de búsquedas se encuentran las distintas opciones por las que se pueden realizar sea por: placa, descripción, inventario y lugar físico.

Figura 70

Ventana de ingreso de texto para búsquedas aplicación móvil.



Fuente: Autores del proyecto.

Para las búsquedas de descripción y lugar físico cuenta con un campo de texto para ingresar el elemento a buscar por sus características.

Figura 71

Ventana búsquedas de inventarios aplicación móvil.

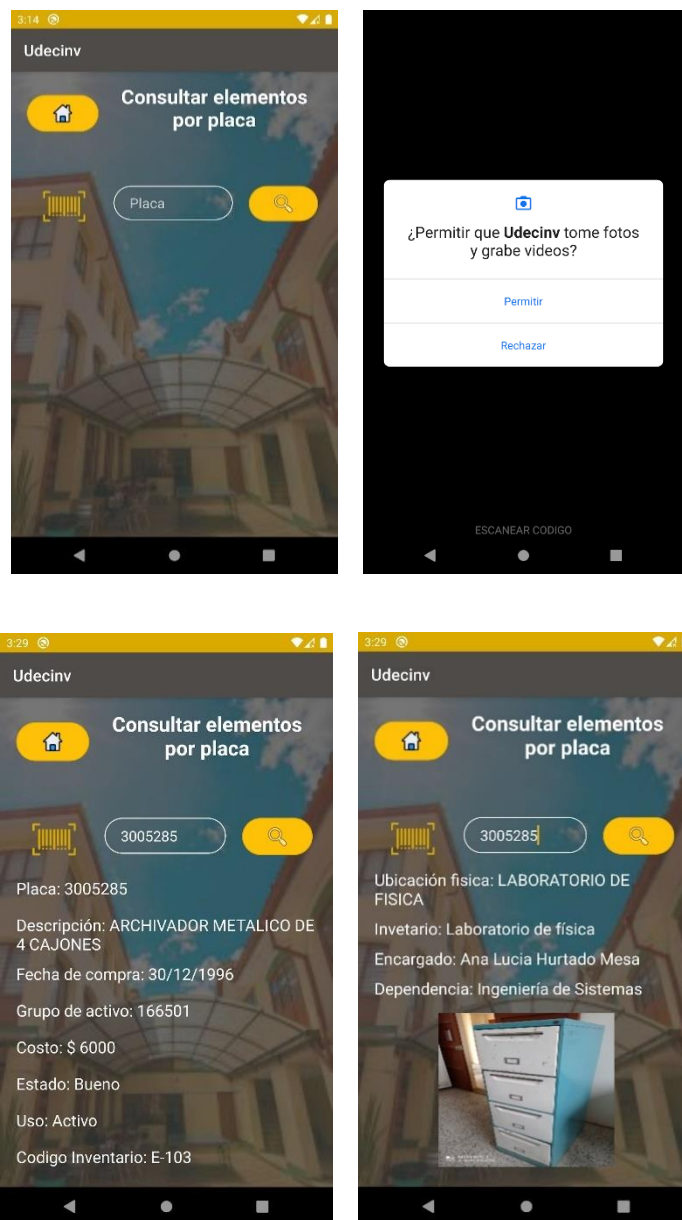


Fuente: Autores del proyecto.

Para la búsqueda por inventario cuenta con una lista desplegable donde se muestra los inventarios con los que cuenta el sistema, de esta manera el usuario debe dar clic en el que quiere realizar la búsqueda.

Figura 72

Ventanas de búsqueda por placa aplicación móvil.



Fuente: Autores del proyecto.

Para las búsquedas por placa se tuvo en cuenta el lector de código de barras el cual permite realizar la búsqueda de una manera más ágil a su vez también cuenta con la opción de digitar el número de placa manualmente.

Código de java para el funcionamiento del lector de código de barras, para las etiquetas de los elementos.

Figura 73

Código de funcionamiento de lector de código de barra aplicación móvil.

```

// camara para escanear
public void escanear() {
    IntentIntegrator intent = new IntentIntegrator( activity: this);
    intent.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
    intent.setPrompt("ESCANEAR ELEMENTO");
    intent.setCameraId(0);
    intent.setOrientationLocked(false);
    intent.setBeepEnabled(false);
    intent.setCaptureActivity(CaptureActivityPortrait.class);
    intent.setBarcodeImageEnabled(false);
    intent.initiateScan();
}

// Respuesta del lector de codigo de barra
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    // escaner
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    Vibrator vibrator = (Vibrator) getApplicationContext().getSystemService(Context.VIBRATOR_SERVICE);
    if(result != null) {
        if(result.getContents() == null) {
            Toast.makeText( context: Bcodigo.this, text: "Cancelaste el escaneo", Toast.LENGTH_SHORT).show();
        } else {
            Placa.setText(result.getContents().toString());
            consultarpla();
            vibrator.vibrate( milliseconds: 150);
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
}

```

Fuente: Autores del proyecto.

Código de java para poder acceder a la cámara de dispositivo para tomar la foto.

Figura 74

Ventana para tomar foto en el aplicativo móvil

```
//permisos para tomar foto

private void fotopermisos() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ActivityCompat.checkSelfPermission( context: Anadir.this, Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
            tomarfoto();
        } else {
            ActivityCompat.requestPermissions( activity: Anadir.this, new String[]{Manifest.permission.CAMERA}, REQUEST_PERMISOS_CAMARA);
        }
    } else {
        tomarfoto();
    }
}

private void tomarfoto() {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, Cod_foto);
    }
}

case Cod_foto:
    if (resultCode == Activity.RESULT_OK && data != null) {
        bitmap = (Bitmap) data.getExtras().get("data");
        foto.setImageBitmap(bitmap);
        bitmap = redimensionarImagen(bitmap, anchoNuevo: 1000, altoNuevo: 800);
    } else {
        Toast.makeText( context: this, text: "No se ha tomado ninguna foto ", Toast.LENGTH_SHORT).show();
    }
}
}
```

```
private Bitmap redimensionarImagen(Bitmap bitmap, float anchoNuevo, float altoNuevo) {  
  
    int ancho = bitmap.getWidth();  
    int alto = bitmap.getHeight();  
  
    if (ancho > anchoNuevo || alto > altoNuevo) {  
        float escalaAncho = anchoNuevo / ancho;  
        float escalaAlto = altoNuevo / alto;  
  
        Matrix matrix = new Matrix();  
        matrix.postScale(escalaAncho, escalaAlto);  
  
        return Bitmap.createBitmap(bitmap, x: 0, y: 0, ancho, alto, matrix, filter: false);  
    } else {  
        return bitmap;  
    }  
}
```

Fuente: Autores del proyecto.

- Opciones de elementos.

Figura 75

Ventana de opción de elementos aplicación móvil.

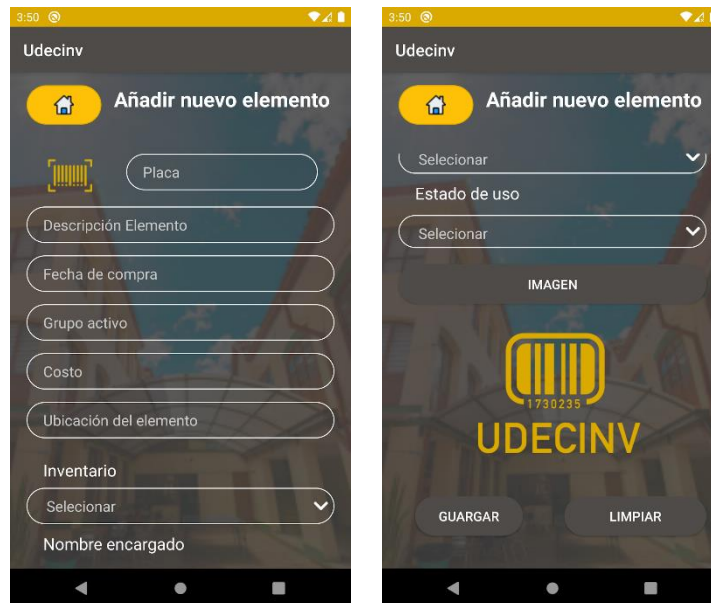


Fuente: Autores del proyecto.

En la pantalla de opciones inventarios cuenta con 3 opciones que son: (añadir elemento, actualizar elementos y dar de baja)

Figura 76

Ventana de añadir nuevo elemento aplicación móvil.



Fuente: Autores del proyecto.

En la opción de añadir elemento cuenta con los campos respectivos para ingresar un nuevo elemento al sistema, a su vez cuenta con un botón para tomar la foto del elemento, activar el lector de código de barras y en la parte inferior con la elección de limpiar los campos o guardar el elemento.

Figura 77

Ventana de actualizar información del elemento aplicación móvil.



Fuente: Autores del proyecto.

En actualizar el elemento se cuenta con la opción de modificar la información como: descripción, ubicación, estado de uso y la imagen de este. Para realizar esta modificación se debe primero consultar el elemento por el número de placa para cargar la información actual, en su parte inferior cuenta con dos botones para limpiar todos los campos y guardar la modificación realizada. Este también cuenta con la opción de utilizar el lector de código de barra.

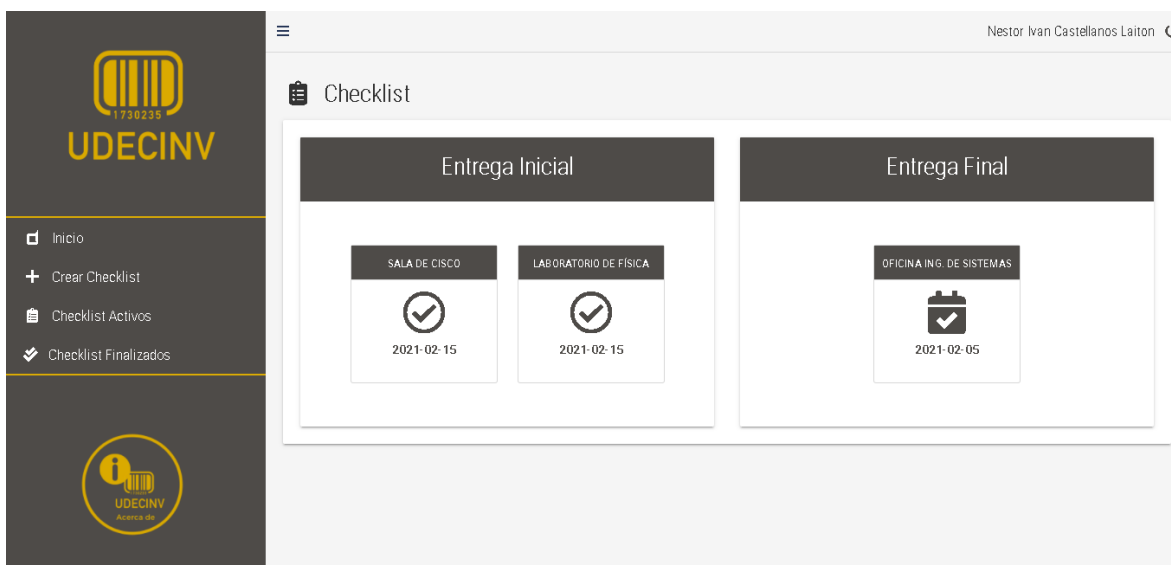
10.3.3. Codificación iteración 2

Codificación de sistema web

- Entregas de inventario.

Figura 78

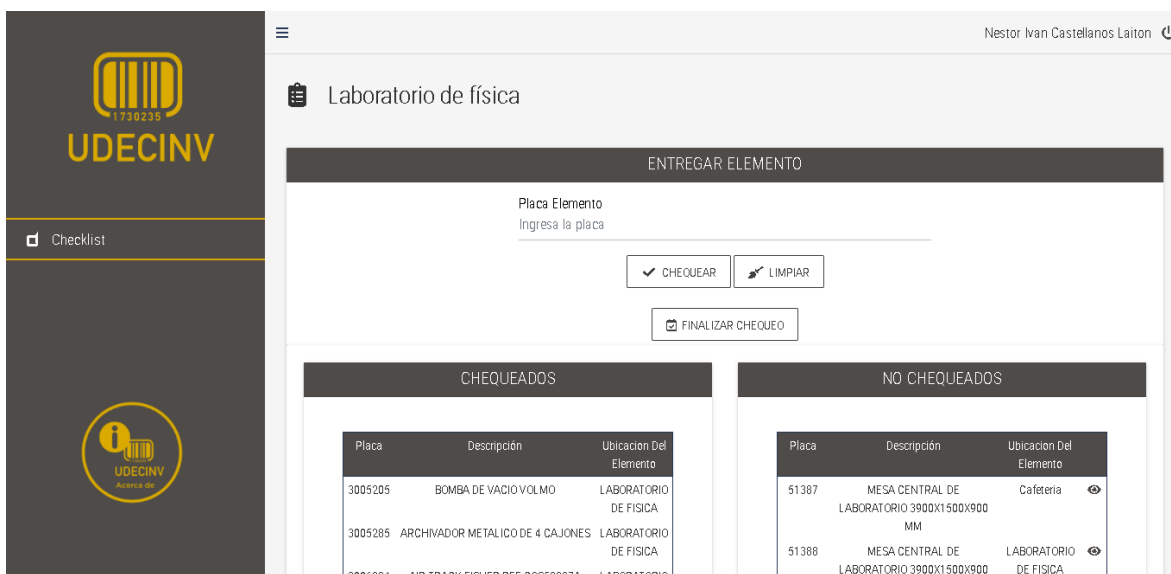
Ventana de entrega de inventarios sitio web.



Fuente: Autores del proyecto.

Figura 79

Ventana de chequeo sitio web.



Fuente: Autores del proyecto.

En el módulo de entregas se encuentran las opciones de creación de checklist para entregas iniciales y finales, de igual forma se pueden evidenciar los checklist activos y los reportes de los

checklist finalizados. Al ingresar a un checklist activo se puede evidenciar los elementos que ya han sido chequeados y los faltantes, en esta página también se puede agregar la placa de los elementos para chequearlos y finalizar el checklist.

- Creación de código de barras.

Figura 80

Ventana de chequeo sitio web.



Fuente: Autores del proyecto.

Los dos roles pueden realizar la generación de códigos de barras para los elementos que físicamente no cuenten con ellos, podrán descargarlo desde el sitio web para su posterior impresión en papel tipo etiqueta para ser adheridos a los elementos en el inventario.

Código en JavaScript para generar código de barras

Figura 81

Código para generar código de barras.

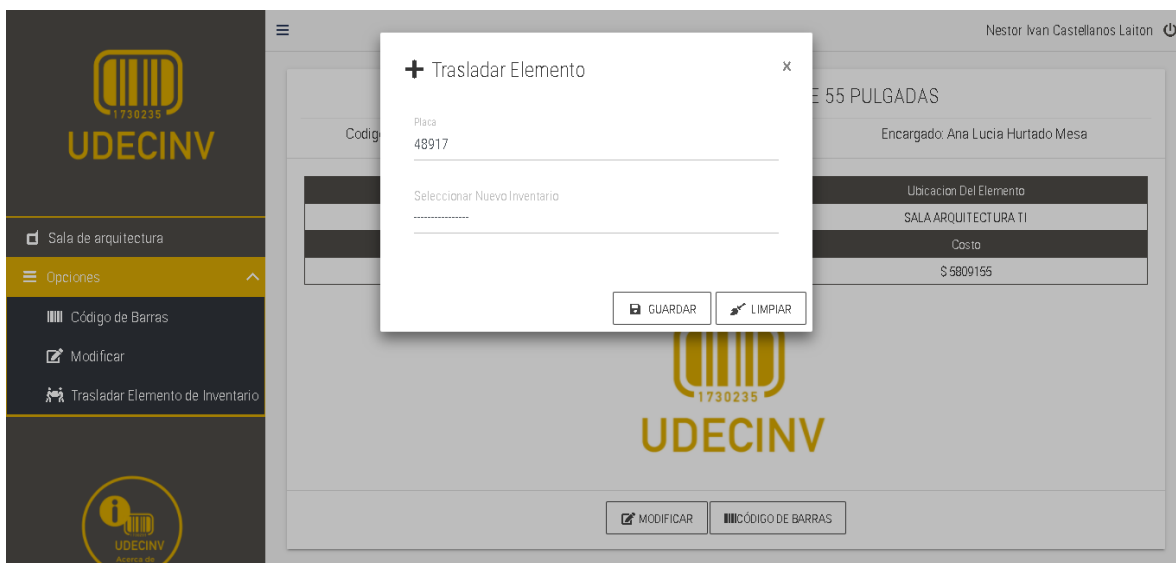
```
function generarcodigobarras(placa) {  
    d=placa.split('|');  
    var x = d[0];  
  
    var titulo=d[1];  
    document.getElementById('nombre').innerHTML=titulo;  
    $('#cod').val(d[0]);  
  
    JsBarcode("#codigobarras",x);  
}
```

Fuente: Autores del proyecto.

- Asignación de inventarios.

Figura 82

Ventana trasladar elemento sitio web.



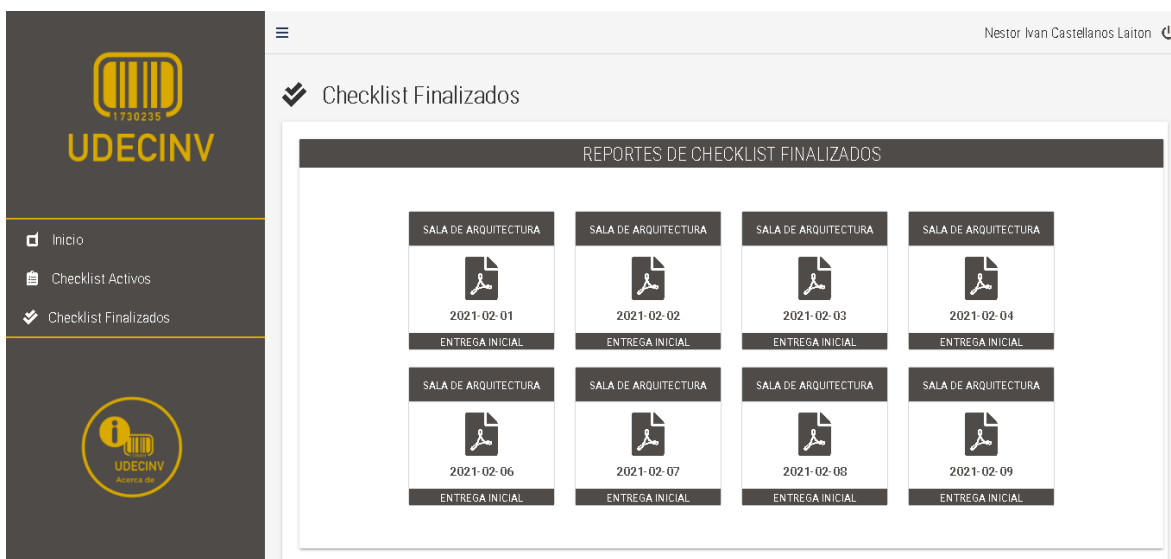
Fuente: Autores del proyecto.

La asignación de inventarios se puede realizar solo desde el usuario administrador en donde se encuentra el elemento a ser actualizado, permitiéndole al usuario realizar esta opción de una manera más ágil y organizada, mediante un formulario de fácil comprensión.

- Reportes

Figura 83.

Ventana de reportes sitio web.



Fuente: Autores del proyecto.

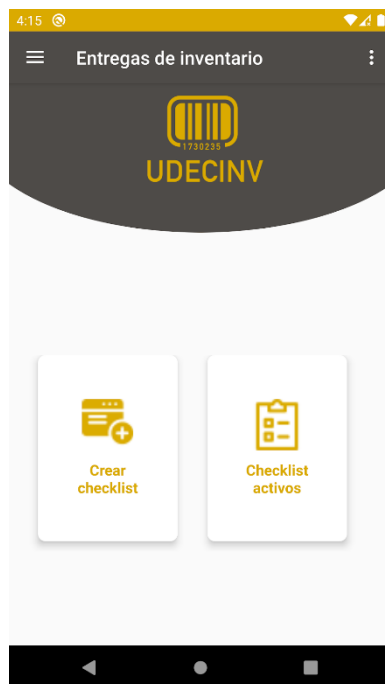
Los reportes de los checklist finalizados se pueden ver en formato pdf con las opciones de abrir en el navegador y poder descargarlos al computador del usuario.

Codificación de sistema móvil Android.

- Entregas de inventario

Figura 84

Ventana de entregas de inventario aplicación móvil.

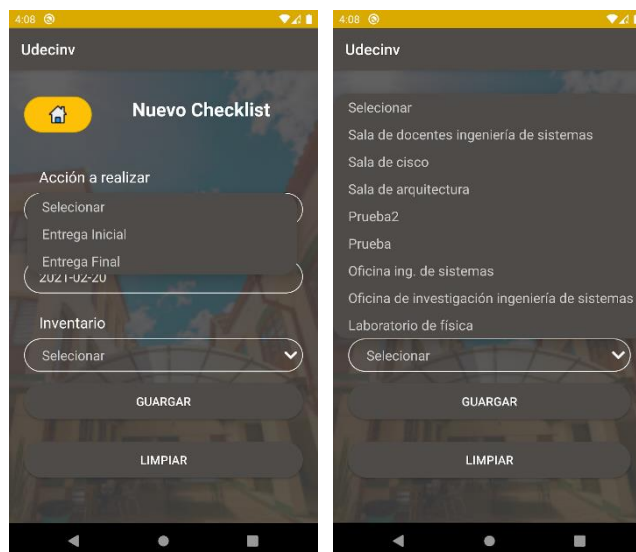


Fuente: Autores del proyecto.

En entregas de inventario se cuenta con dos opciones: crear checklist y checklist activos.

Figura 85

Ventana de crear checklist aplicación móvil.

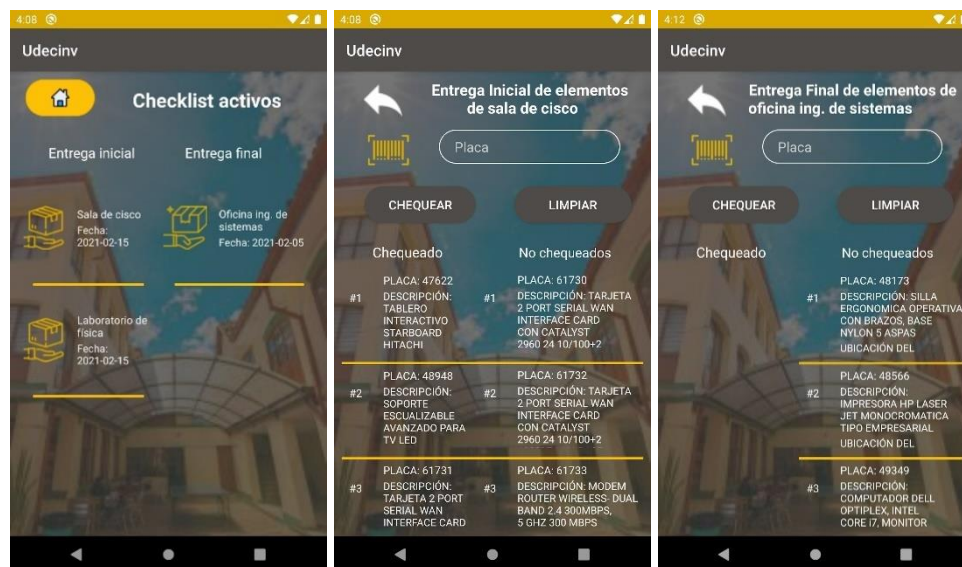


Fuente: Autores del proyecto.

En crear checklist se cuenta con la opción de seleccionar que tipo de acción se va a realizar si va a hacer entrega inicial o final, de igual manera el día que se realiza y el inventario a que se le va a realizar el proceso de chequeo de los elementos.

Figura 86

Ventana de checklist aplicación móvil.



Fuente: Autores del proyecto.

En la opción de checklist activos se encuentra los checklist que están en proceso de chequeo, el usuario administrador da clic sobre el checklist a realizar, de esta manera se visualiza la pantalla donde chequeo de elementos donde se encuentra los que han sido chequeado y los que aún se encuentran pendientes, para el proceso de chequeo se puede realizar de manera manual o por medio de lector de código de barras.

- Asignación de inventarios.

Figura 87

Ventana de opción de inventario aplicación móvil.

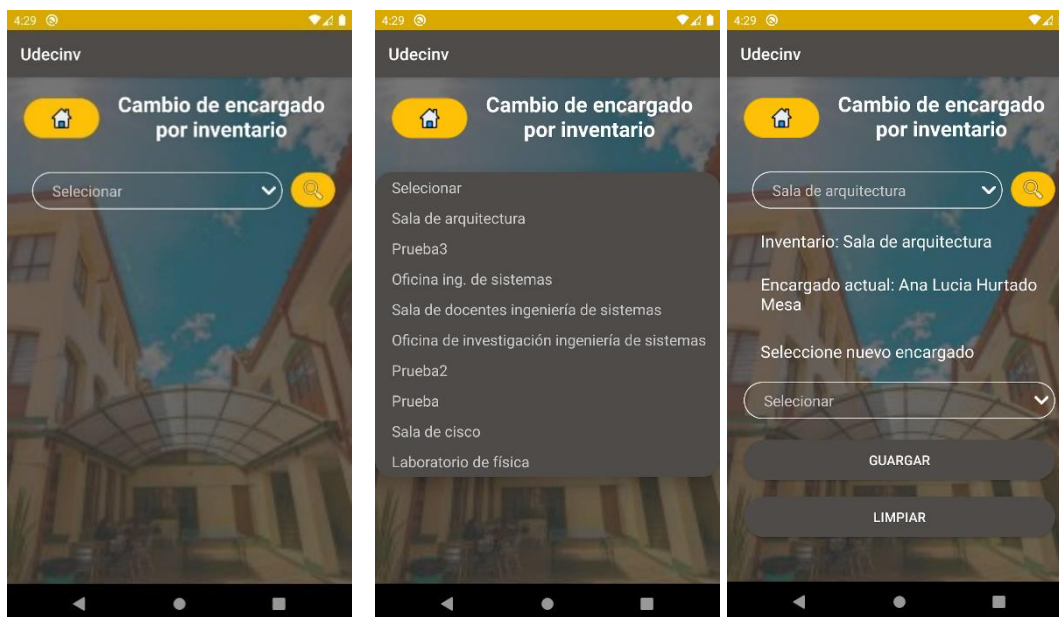


Fuente: Autores del proyecto.

En asignación de inventarios se encuentra el cambio de encargado por inventario o el traslado de elementos de inventario.

Figura 88

Ventana de cambio de encargado por inventario en aplicación móvil.

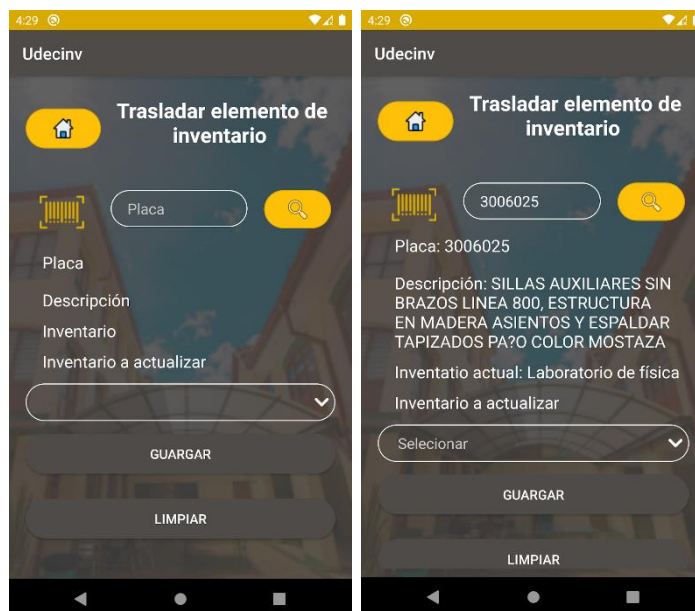


Fuente: Autores del proyecto.

En cambio, de encargado se selecciona el inventario a cambiar de encargado, dando clic en el inventario a actualizar, se despliega la información del inventario y el encargado actual y un menú desplegable con los encargados.

Figura 89

Ventana de trasladar elemento de inventario aplicación móvil.



Fuente: Autores del proyecto.

En trasladar elemento se busca el elemento por el número de placa, de esta manera se mostrará información relevante del sistema y un menú desplegable con los inventarios a donde se quiere trasladar el elemento.

10.3.4. Codificación iteración 3

Codificación de sistema web

- Solicitudes usuario dependencia

Figura 90

Ventana de crear solicitud sitio web.

The image shows a web application interface with a modal dialog box for creating a modification request. The dialog box is titled '+ Crear Solicitud De Modificación' and contains the following fields:

- Plata: 3005213
- Ubicación del Elemento: LABORATORIO DE FISICA
- Estado De Uso: Bueno

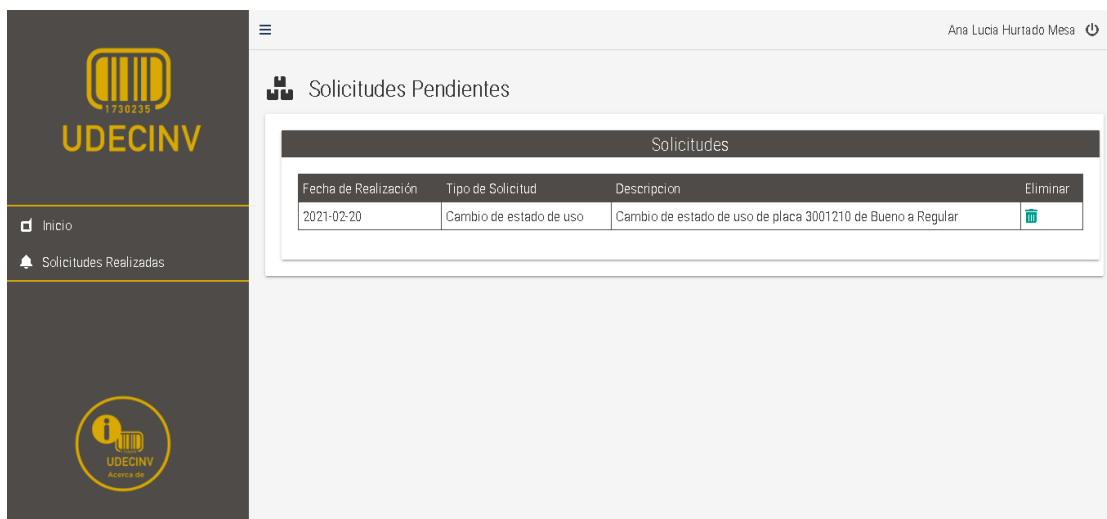
A 'GUARDAR' button is located at the bottom right of the dialog. The background interface includes a sidebar with the UDECINV logo and navigation options: 'Laboratorio de física', 'Crear Solicitud', and 'Codigo de Barras'. The main content area displays a table with the following data:

Ubicación Del Elemento
LABORATORIO DE FISICA
Costo
35200


Fuente: Autores del proyecto.

Figura 91

Ventana de solicitudes pendientes sitio web.



The screenshot shows a web application interface for UDECINV. On the left is a dark sidebar with the UDECINV logo and navigation links for 'Inicio' and 'Solicitudes Realizadas'. The main content area is titled 'Solicitudes Pendientes' and contains a table with the following data:

Fecha de Realización	Tipo de Solicitud	Descripción	Eliminar
2021-02-20	Cambio de estado de uso	Cambio de estado de uso de placa 3001210 de Bueno a Regular	

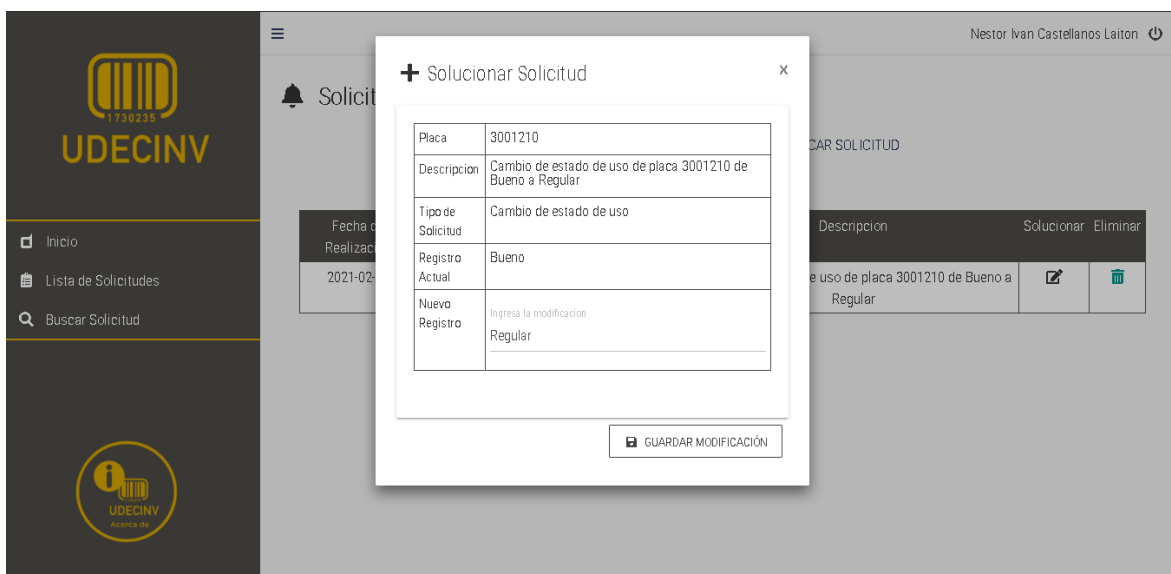
Fuente: Autores del proyecto.

El usuario dependencia puede generar solicitudes del estado y ubicación de los elementos para que se realice un cambio respecto a estas características, puede ver las solicitudes que no han sido resueltas y eliminar las que no son relevantes.

- Respuesta solicitudes.

Figura 92

Ventana de crear solicitudes sitio web.



Fuente: Autores del proyecto.

El usuario administrador puede resolver estas solicitudes de tal manera que se cumpla con lo requerido por el encargado o por el contrario puede eliminar la solicitud y no dar solución a esta, el usuario administrador puede encontrar estas solicitudes desde la página principal de su rol.

- Notificaciones.

Figura 93

Ventana de crear solicitudes sitio web.



Fuente: Autores del proyecto.

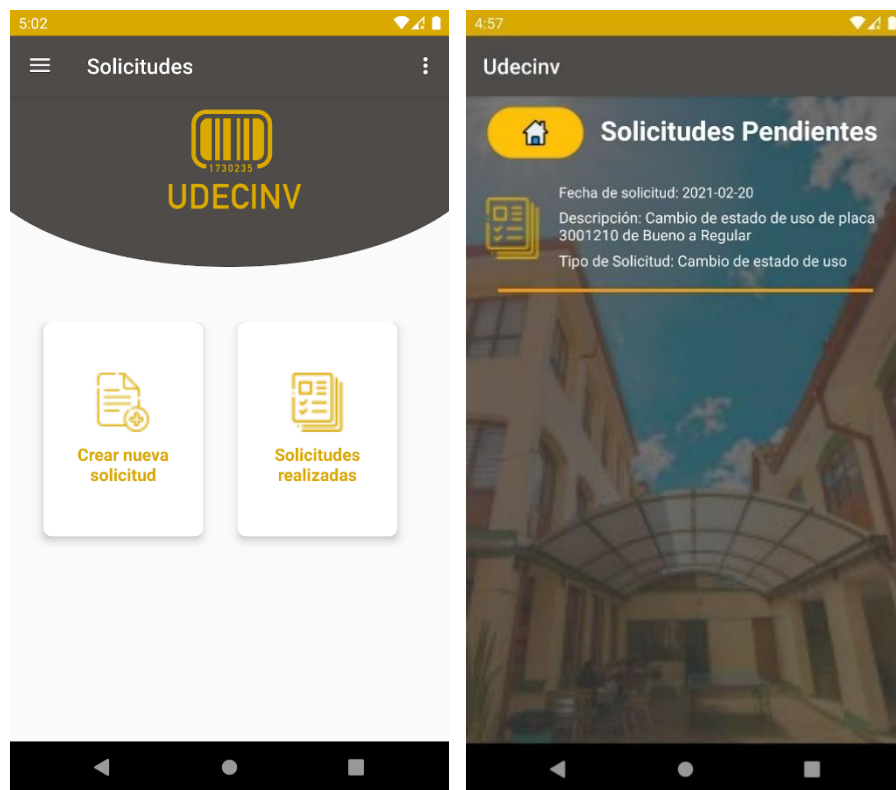
Las notificaciones permiten al usuario administrador ver la cantidad de solicitudes que tiene pendiente por responder, al ingresar al sistema por este rol se puede evidenciar la cantidad de solicitudes que tiene desde el sitio web.

Codificación de sistema móvil Android

- Solicitudes usuario dependencia

Figura 94

Ventana solicitudes aplicación móvil.



Fuente: Autores del proyecto.

En solicitudes por usuario dependencia cuenta con las opciones de Crear una nueva solicitud y de visualizar las solicitudes realizadas.

Figura 95

Ventana crear nueva solicitudes aplicación móvil.



Fuente: Autores del proyecto.

Para crear una solicitud se debe primero consultar el elemento al cual se le quiere realizar una solicitud ya sea por cambio de lugar físico o cambio de estado de elemento. Al momento de realizar la búsqueda se visualiza la información actual de elementos y una imagen para tener certeza de que si es elemento a realizar solicitud.

Figura 96

Ventana solicitudes realizadas usuario dependencia aplicación móvil.



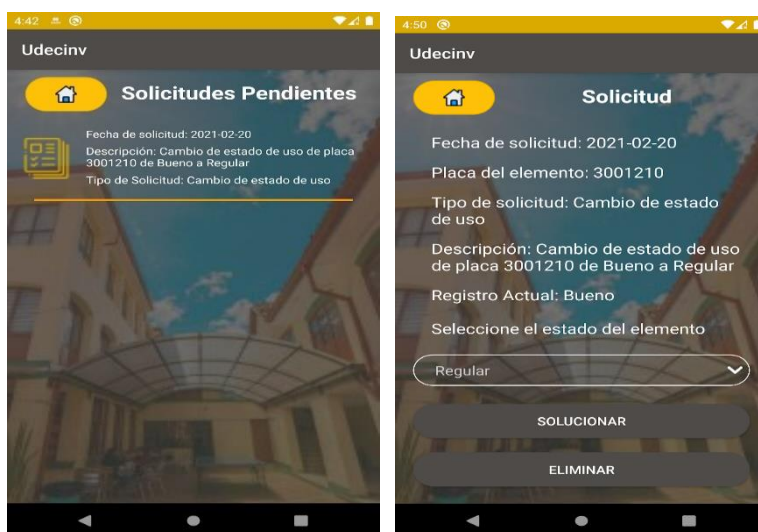
Fuente: Autores del proyecto.

El usuario dependencia al visualizar la solicitud de la información contará con la opción de poder eliminar la solicitud realizada.

- Respuesta solicitudes.

Figura 97

Ventana solicitudes usuario administrador aplicación móvil.



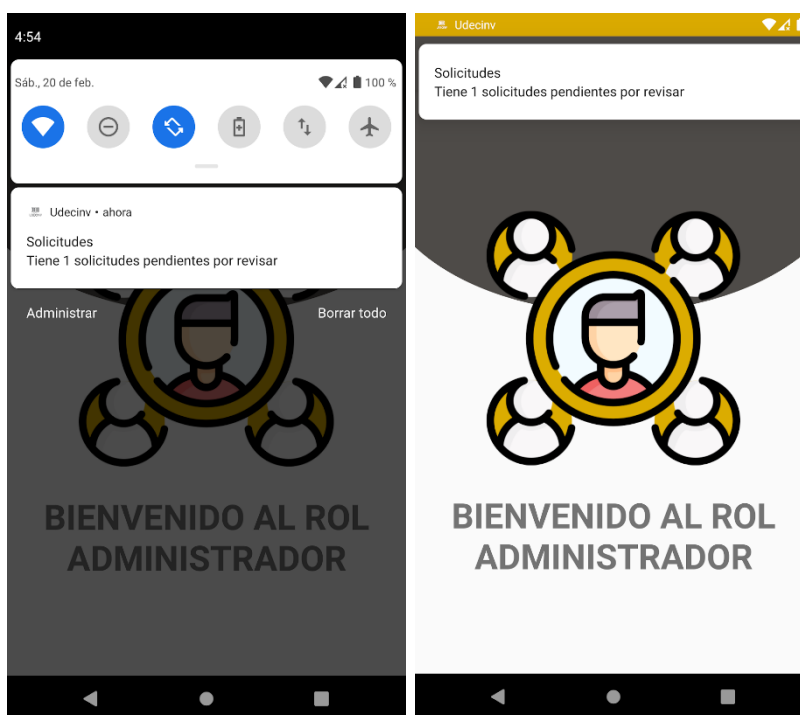
Fuente: Autores del proyecto.

En respuesta de solicitudes es donde el usuario administrador puede ver las solicitudes que ha generado el usuario administrador así de esta manera se puede seleccionar el tipo de solicitud a resolver, allí se muestra la información de la solicitud de registro actual y los cambios a realizar, cuenta con dos opciones de solucionar solicitud o eliminar esta.

Figura 98

Notificaciones aplicación móvil.

- Notificaciones



Fuente: Autores del proyecto.

En el rol de usuario dependencia cuenta con el funcionamiento de notificaciones que se generan cuando se encuentran solicitudes activas a resolver, se puede dar clic sobre la notificación y de esta manera lo re direccionará a las solicitudes pendientes a resolver.

Estructura de archivos del sistema en sitio web.

Figura 99

Estructura de archivos sitio web.

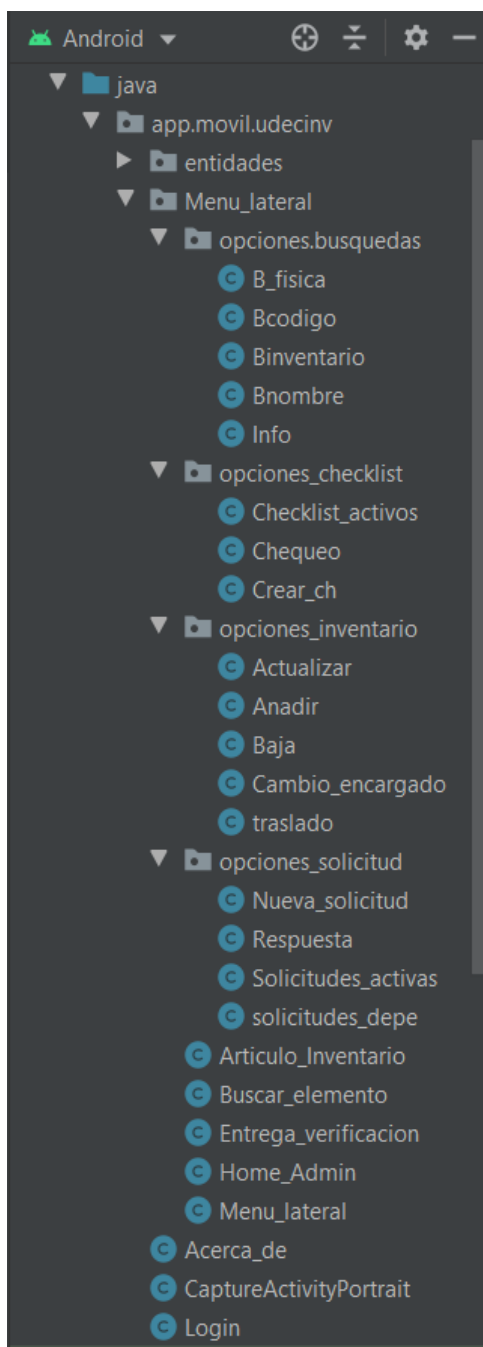


Fuente: Autores del proyecto.

Estructura de archivos del sistema en aplicación móvil.

Figura 100

Estructura de archivos sitio web.

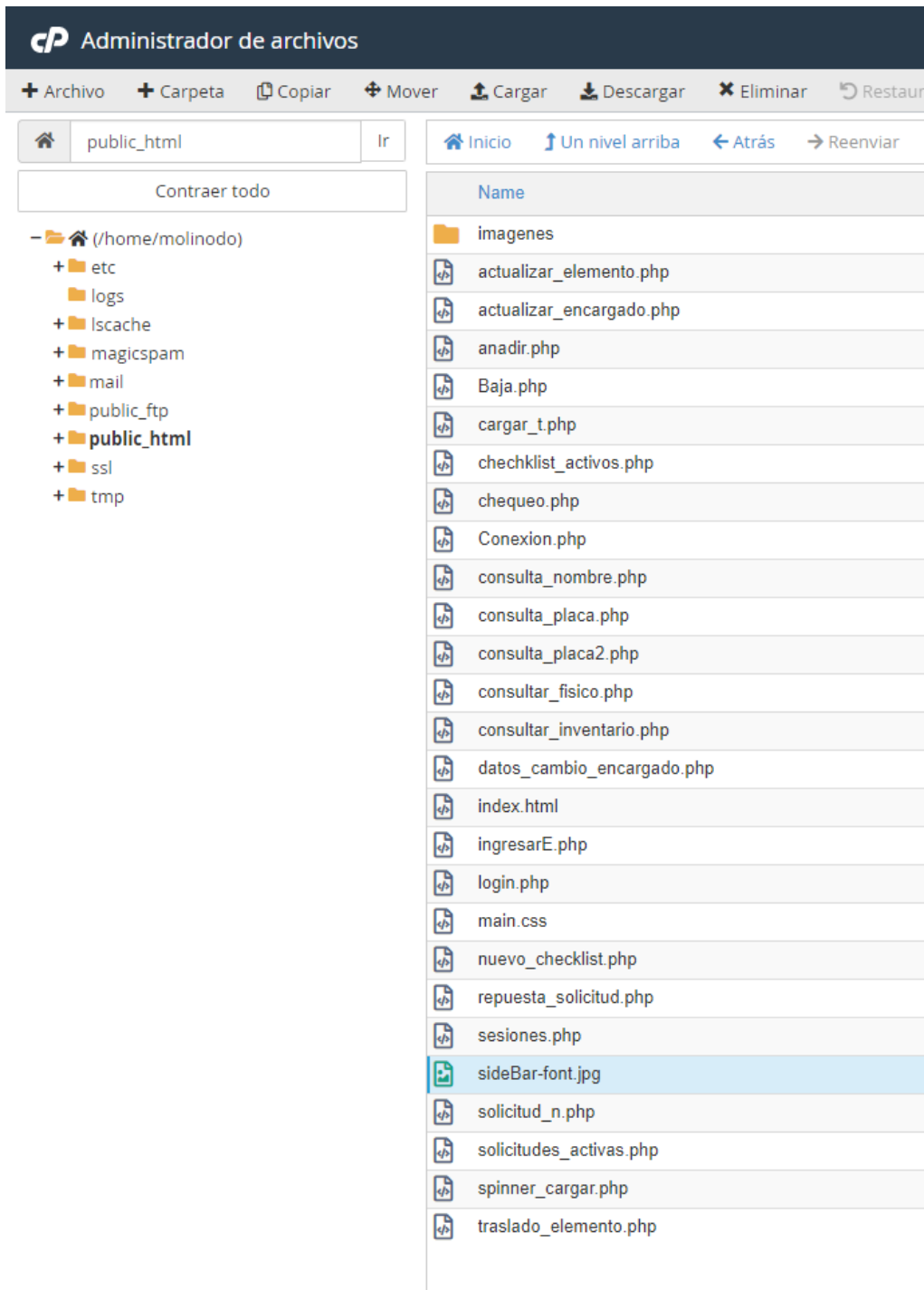


Fuente: Autores del proyecto.

Estructura del sistema en aplicación móvil como función de Rest Api en el servidor.

Figura 101

Estructura de archivos en el servidor.



Fuente: Autores del proyecto.

Rutas de navegación entre módulos de usuario administrador en sistema web

Figura 102

Rutas de navegación usuario administrador sitio web.

```
const enrutador = require('express').Router();
const passport = require('passport');
const controlAdmin = require('../controlador/controladmin');
const { isLoggedInadmin } = require('../funciones/cierreautomatico');

enrutador.get('/', (req, res, next) => {
  res.render('index');
});

//validar tipo de rol
enrutador.get('/validar', controlAdmin.valida);

//login
enrutador.post('/signin', passport.authenticate('local-signin', {
  successRedirect: '/validar',
  failureRedirect: '/',
  failureFlash: true
}));

//nuevo usuario
enrutador.get('/usuarios', isLoggedInadmin, controlAdmin.listausuarios);
enrutador.get('/NAdmin:id', isLoggedInadmin, controlAdmin.NAdmin);
enrutador.post('/addencarusu', isLoggedInadmin, controlAdmin.saveencarusu);
enrutador.post('/updateencarusu', isLoggedInadmin, controlAdmin.updateencarusu);
enrutador.post('/updateusuencar', isLoggedInadmin, controlAdmin.updateusuencar);

//encargados
enrutador.get('/encargados', isLoggedInadmin, controlAdmin.listaencargados);
enrutador.get('/administrador', isAuthenticatedadmin, controlAdmin.administrador);

enrutador.get('/logout', (req, res, next) => {
  req.logout();
  res.redirect('/');
});
```

Fuente: Autores del proyecto.

Rutas de navegación entre módulos de usuario dependencia en sistema web.

Figura 103

Rutas de navegación usuario dependencia sitio web.

```
const enrutador = require('express').Router();
const controlDepen = require('../controlador/controldependencia');

const { isLoggedInUsuD } = require('../funciones/cierreautomatico');

enrutador.get('/dependencia', isAuthenticatedUsuD, controlDepen.minventarioo);

enrutador.get('/elemendepen:id',isLoggedInUsuD, controlDepen.elementdepen);
enrutador.get('/mielemento:id', isLoggedInUsuD, controlDepen.mielemento);

enrutador.post('/addsololicitud', isLoggedInUsuD, controlDepen.addsololicitud);

enrutador.get('/solrealizadas', isLoggedInUsuD, controlDepen.solrealizadas);
enrutador.get('/eliminarsol:id',isLoggedInUsuD, controlDepen.eliminarsol);

//usuario dependencia
function isAuthenticatedUsuD(req, res, next) {
  var usuarioD = 'Usuario Dependencia';
  if(req.isAuthenticated() && req.user.Rol == usuarioD) {
    return next();
  }
  res.redirect('/')
}

module.exports = enrutador;
```

Fuente: Autores del proyecto.

10.4. Pruebas

10.4.1. Pruebas unitarias

El proceso de pruebas unitarias es esencial en el proyecto, este está enfocado hacia la verificación del código por parte de sus desarrolladores, permite realizar la entrega del producto sin inconvenientes que puedan perjudicar su uso cuando el cliente lo esté ejecutando, así mismo incrementa la calidad del código del desarrollador. Es importante destacar que para que las pruebas unitarias sean exitosas deben cumplir con diferentes estándares como poder automatizarse y repetir la pruebas varias veces sin generar ningún error, de la misma manera cualquier desarrollador puede hacer pruebas después de solucionar el error con el fin de que se verifique que el error ha sido solucionado correctamente.

Durante la realización de las pruebas en el sistema multiplataforma se hizo la verificación de cada uno de los módulos donde se logró encontrar diferentes errores que fueron solucionados de forma inmediata, como fallas en algunos formularios, verificación de las consultas a la base de datos, seguimiento de los datos desde el front-end hasta llegar a ser procesados en el back-End, verificación de usuarios e información suministrada por ellos en los formularios entre otras pruebas como las de concurrencia, seguridad y velocidad del sistema.

Figura 104

Validación de usuario sitio web.

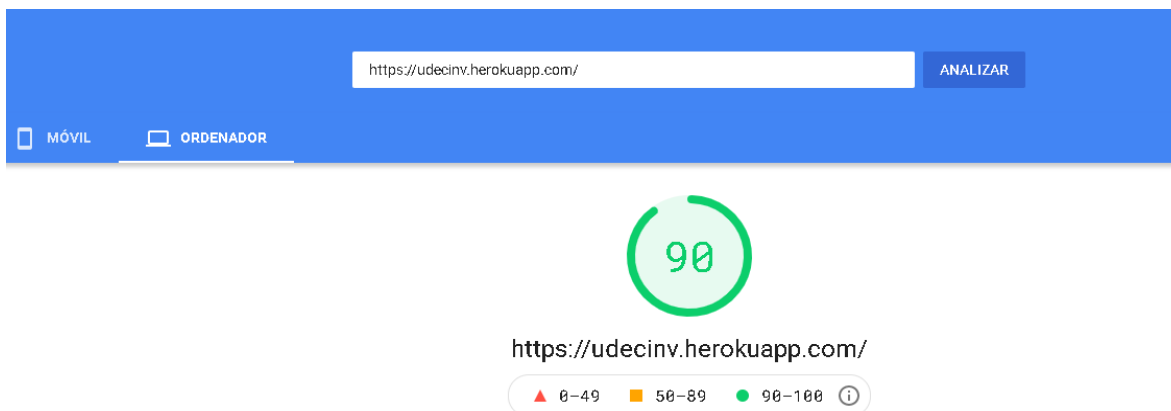


Fuente: Autores del proyecto.

10.4.1.1. Prueba de velocidad sitio web

Figura 105

Test de velocidad de sitio web.



Fuente: Autores del proyecto.

Se utilizó pagespeed el cual es un sitio para hacer pruebas a las páginas web que realiza esta función de manera gratuita y online, en este sitio se puede evidenciar los rangos de velocidad de los sitios, donde se determina de la siguiente manera: Malo: 08-49, Regular: 50-89 y Bueno: 90-100, en este test realizado el sitio nos muestra un puntaje de 90 que indica que la velocidad del sistema multiplataforma en sitio web es buena.

Prueba de concurrencia sitio web con Jmeter

Figura 106

pruebas de concurrencia sitio web.

prueba con 500 usuarios

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento
Peticion HTTP Reques...	500	1782	157	4411	901,17	0,00%	93,1/sec
Total	500	1782	157	4411	901,17	0,00%	93,1/sec

prueba con 1000 usuarios

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec
Peticion HTTP Reques...	1000	99174	670	1293421	310084,31	9,40%	46,4/min	4,63
Total	1000	99174	670	1293421	310084,31	9,40%	46,4/min	4,63

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes	Sent Bytes	Latency
1	23:30:31.435	Udecinv Web 1-205	Peticion HTTP Request	670	✔	6544	168	670
2	23:30:31.432	Udecinv Web 1-202	Peticion HTTP Request	769	✔	6548	168	769
3	23:30:31.434	Udecinv Web 1-203	Peticion HTTP Request	856	✔	6546	168	856
4	23:30:31.428	Udecinv Web 1-198	Peticion HTTP Request	882	✔	6546	168	882
5	23:30:31.424	Udecinv Web 1-193	Peticion HTTP Request	897	✔	6544	168	897
6	23:30:31.418	Udecinv Web 1-188	Peticion HTTP Request	953	✔	6550	168	953
7	23:30:31.437	Udecinv Web 1-207	Peticion HTTP Request	952	✔	6546	168	952
8	23:30:31.426	Udecinv Web 1-196	Peticion HTTP Request	999	✔	6552	168	999
9	23:30:31.423	Udecinv Web 1-192	Peticion HTTP Request	1194	✔	6548	168	1194
10	23:30:31.416	Udecinv Web 1-186	Peticion HTTP Request	1204	✔	6552	168	1204
11	23:30:31.412	Udecinv Web 1-181	Peticion HTTP Request	1247	✔	6550	168	1247
12	23:30:31.414	Udecinv Web 1-183	Peticion HTTP Request	1250	✔	6548	168	1250
13	23:30:31.405	Udecinv Web 1-175	Peticion HTTP Request	1264	✔	6544	168	1264
14	23:30:31.410	Udecinv Web 1-180	Peticion HTTP Request	1281	✔	6546	168	1281
15	23:30:31.414	Udecinv Web 1-184	Peticion HTTP Request	1292	✔	6546	168	1292

Fuente: Autores del proyecto.

Las pruebas de concurrencia se realizaron mediante la aplicación Jmeter donde se ejecutaron pruebas con 500 y 1000 usuarios comprobando la carga y estrés del sitio web en rendimiento y tiempos de respuesta del sistema, como resultado de estas pruebas se obtuvo un correcto funcionamiento de los tiempos de respuesta del sistema a pesar de estar implementado en un servidor gratuito que no garantiza todas las ventajas de rendimiento de un servidor pago. En la prueba con una muestra de 500 usuarios se obtuvo como resultado un tiempo mínimo de 157 mls y un máximo de 4411 mls, dando esto un margen de error del 0% en las peticiones realizadas con http request por el método get, mientras que en la prueba con una muestra con 1000 usuarios se obtuvo como resultado un tiempo mínimo de 670 mls y un máximo de 1293421 mls obteniendo como resultado un margen de error del 9.40% de las peticiones realizadas.

10.4.1.2. pruebas de compatibilidad

El sistema multiplataforma para aplicación móvil fue desarrollado para celulares con una versión de Android mayor a la 4.1 esto brindando una compatibilidad de 16.738 dispositivos que podrán descargar la aplicación desde la play store.

Se realizaron pruebas en los siguientes dispositivos nombrados en la tabla probando su funcionalidad y adaptación a la pantalla, comprobando de que cada una de las opciones de sistema funcionaran de una manera eficiente.

Tabla 75

Pruebas de compatibilidad.

Marca	Referencia	Versión de Android	Resultado
Xiaomi	Redmi Note 8	9	Correcto
Huawei	P9 lite smart	6	Correcto
Motorola	Moto g	5	Correcto
Samsung	A71	10	Correcto
Huawei	P9 smart	8	Correcto
Huawei	Mate 10 lite	7	Correcto
Xiaomi	Redmi note 9	10	Correcto

Fuente: Autores del proyecto.

10.4.2. Pruebas de aceptación

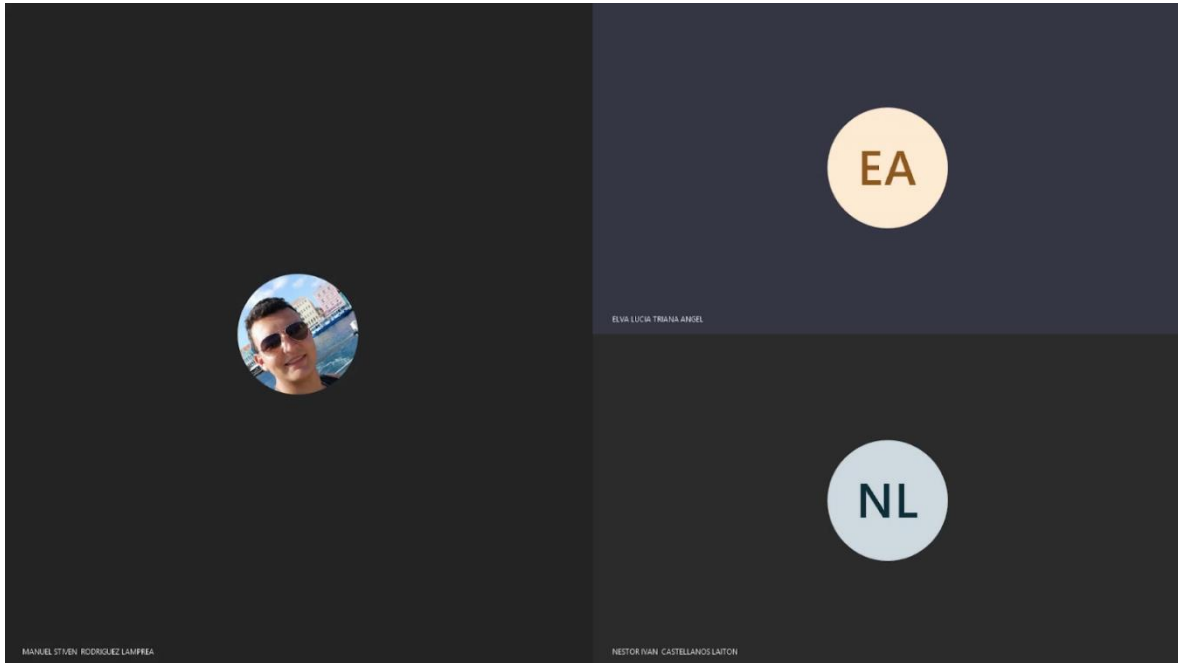
Para el proceso de pruebas de aceptación con el cliente se realizó una serie de reuniones para verificar si los módulos planteados para la iteración asignada fueron cumplidos o faltaron tareas pendientes por realizar. Durante cada iteración al finalizar el cliente verifica cada uno de los módulos verificando que cumpla con lo acordado en las historias de usuarios de esta manera el cliente nos da una respuesta de cumplimiento o una observación a cambios a realizar.

Para las pruebas de aceptación en la última iteración se agendó una reunión en la Universidad de Cundinamarca seccional Ubaté con el cliente para validar que todos los módulos y opciones cumplieran y diera el resultado esperado, en este proceso se tomó como prueba el inventario de Laboratorio de física que pertenece al programa de ingeniería de sistemas.

Pruebas de aceptación con clientes por medio de videollamada en cada una de las iteraciones.

Figura 107

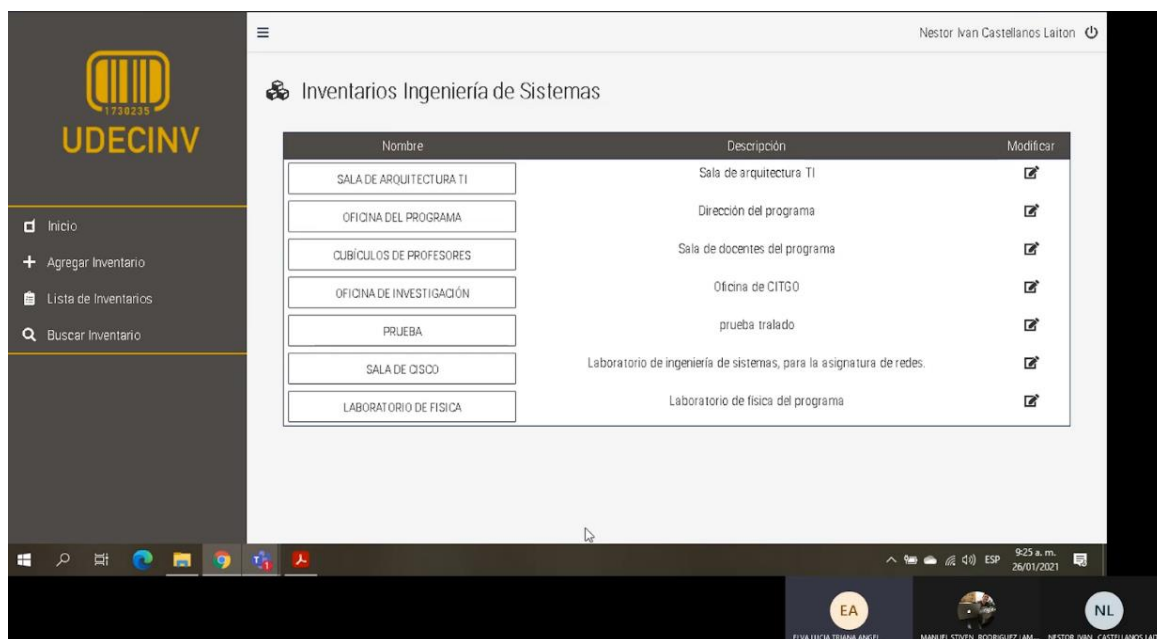
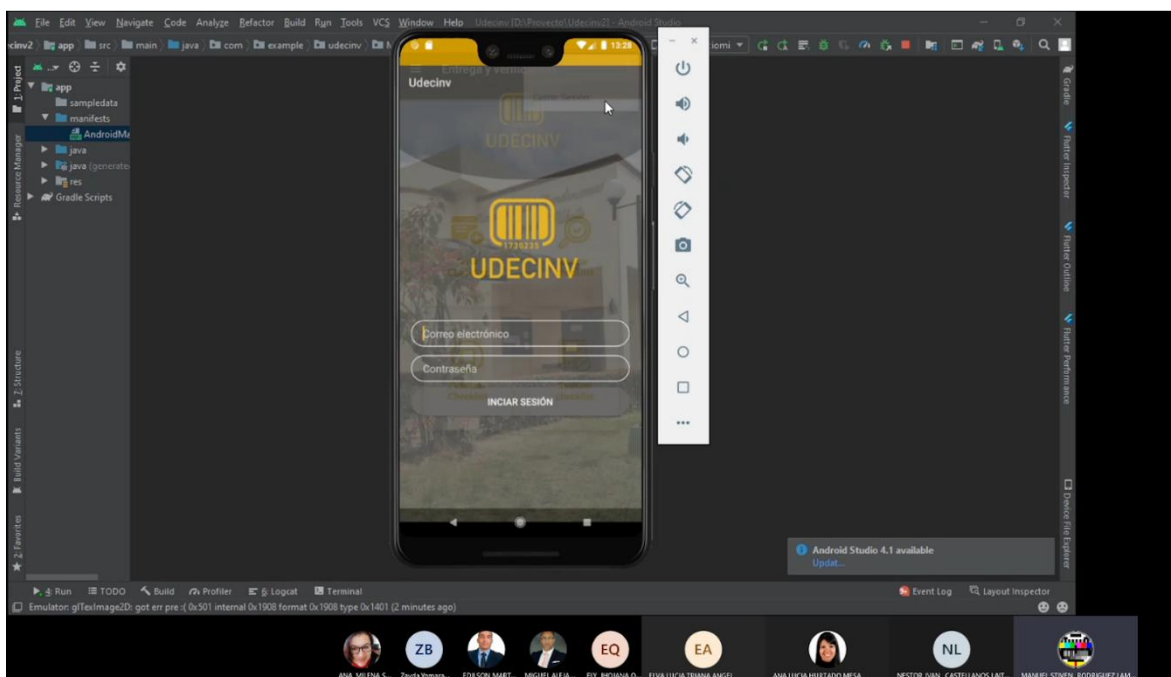
Reuniones por video conferencia.

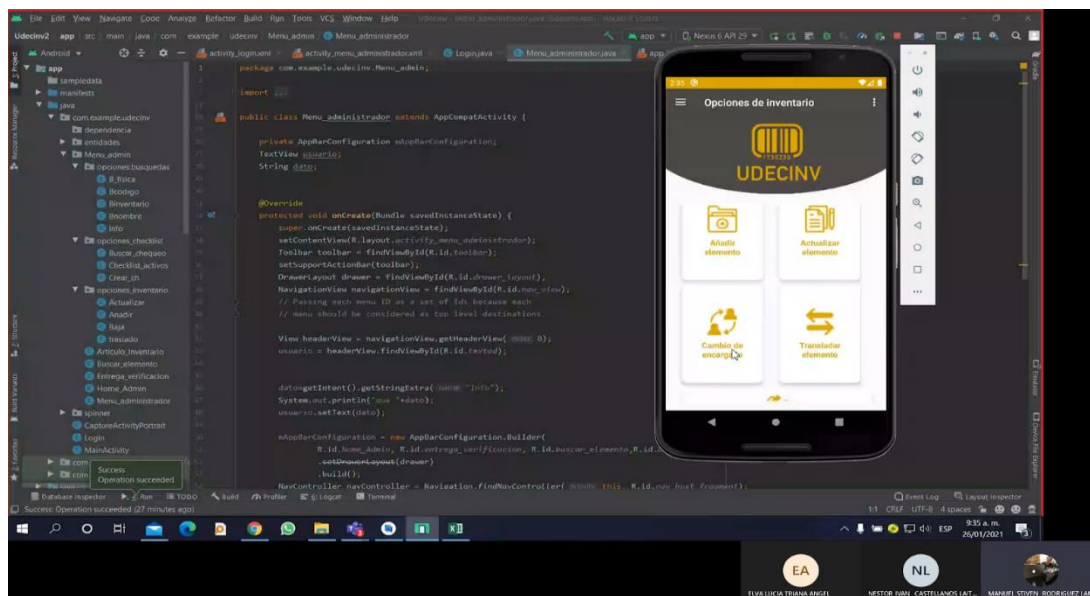


Fuente: Autores del proyecto.

Figura 108

Reuniones por video conferencia avances del proyecto.





Fuente: Autores del proyecto.

Figura 109

Pruebas de aceptación sitio web con Elva Triana encargada de almacén.



Fuente: Autores del proyecto.

Figura 110

Pruebas de aceptación aplicación móvil con Elva Triana encargada de almacén.



Fuente: Autores del proyecto.

Figura 111

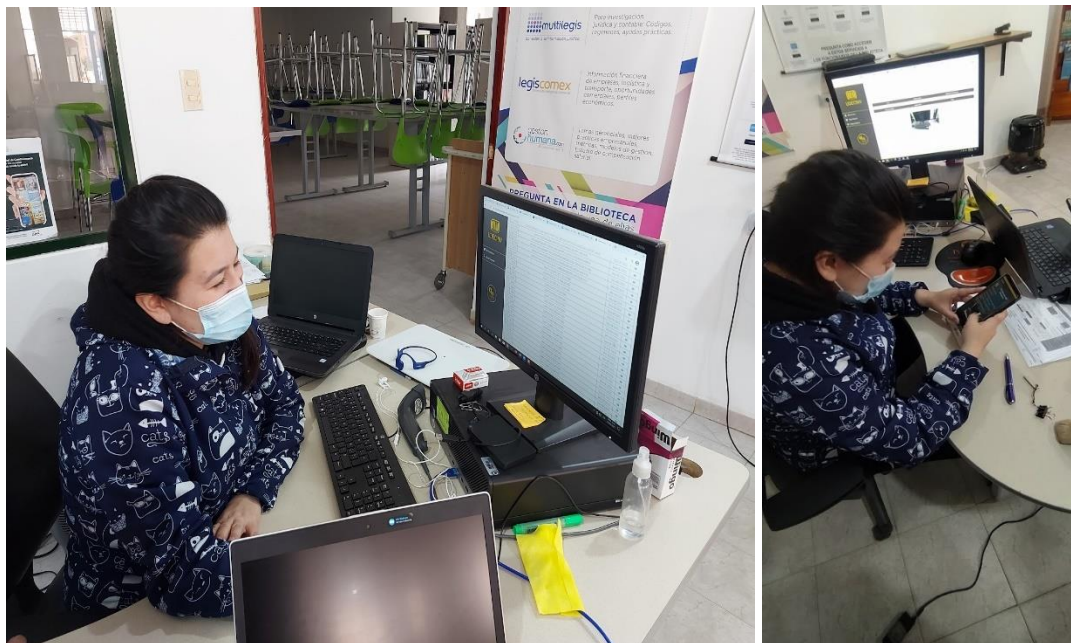
Pruebas de aceptación, encuesta verificación de módulos sistema multiplataforma.



Fuente: Autores del proyecto.

Figura 112

Pruebas de aceptación rol usuario dependencia, Ingeniería Ana Milena Soto Encargada de oficina de sistemas y tecnología.



Fuente: Autores del proyecto.

10.4.3. Detección y documentación de errores

Tabla 76

Detección y documentación de errores.

#	Error	Descripción	Solución
01	Renderización de imágenes	El servidor se saturó al cargar todas las imágenes al tiempo en la vista del sitio web desde la base de datos.	Se realizó un filtro por inventarios donde se muestra solo la información básica de todos los elementos con la opción de ver solo la foto del elemento seleccionado.
02	Validación de placas	En el proceso de checklist de los inventarios, no se tuvo en cuenta si las placas eran existentes o estaban en el inventario que se estaba chequeando esto generaba un error en la consulta SQL, dando paso a un bloqueo en el servidor.	Se validó la existencia de las placas en el inventario y que su estado fuera activo.

03	Botón limpiar	<p>El proceso de modificar genera un formulario con la información a modificar precargada en este, esto aumenta la eficacia del sistema, pero al formulario tiene un botón con la opción de limpiar esto generaba que esta información se eliminará y al guardar la modificación se generaba error en la consulta a la base de datos.</p>	<p>Se eliminó el botón con el objetivo de que la información que se desea eliminar se realice desde el teclado y no altere el funcionamiento del sistema.</p>
04	Adaptación del reporte PDF	<p>La creación del reporte de los checklist se almacena en la base de datos en formato pdf este formato se altera al generarlo desde el servidor de Heroku por motivos de diferencias en las versiones de pdf node-creator, en consecuencia, de esto la tabla del reporte no se muestra completa en el pdf.</p>	<p>Se alteró la orientación de la página con el objetivo de que la información se muestre completa en el pdf.</p>
05	Traslado de elemento	<p>En el proceso de trasladar un elemento se registraba este en cualquier inventario, esto generaba un error de consultas, ya que, si el inventario no tiene elementos, significa que el inventario está inactivo, en consecuencia, el sistema se bloqueaba en este proceso.</p>	<p>Se colocó un prerequisite en las consultas donde se valide que el inventario esté activo o con elementos registrados.</p>
06	Peso de las fotos	<p>al guardar un elemento en el inventario no se tuvo en cuenta el peso de las fotos, donde la consulta guarda de igual manera, pero si la imagen era muy pesada esta no era almacenada en base de datos sin informar al usuario.</p>	<p>Se restringe el peso de las fotos a un máximo de 500Kb con el fin de que las funcionalidades del sistema no se alteren.</p>
07	Error en respuesta de consultas a la base de datos	<p>Al momento de realizar consultas a la base de datos a diferentes tablas de este, se presenta el error que no se visualizaba nada en la petición, esto debido a que la información solicitada contaba con un carácter que tuviera una tilde o una ñ.</p>	<p>Se le agregó el cotejamiento utf8 en cada una de las consultas a la base de datos.</p>
08	Cargar lista de elementos de inventarios.	<p>En el momento de realizar una consulta a la base de datos y el dato buscado no se encontraba, se presentaba el error que el Array list encargado de almacenar esta información generaba un error por estar vacío.</p>	<p>Se agregó una condición que cuando la consulta fuera vacía diera una respuesta de que no se encontraba la búsqueda realizada.</p>
09	Cargar foto tomada desde el teléfono móvil	<p>En la opción de añadir y actualizar elementos al momento de dar clic en el botón de imagen y no se tomará una foto, tenía una respuesta de detener la aplicación, esto debido a que en el momento de dar en la opción de foto se</p>	<p>Para corregir este error solo fue necesario llamar la clase sólo cuando el usuario seleccionara la foto indicada.</p>

		llamaba la clase de renderizar imágenes, la clase al no recibir ninguna información generaba el error.	
10	Sesiones en Android	Al iniciar la aplicación en su menú principal de cualquier rol se presentaba un problema con la opción de cerrar sesión, esto debido a que no se estaba finalizando esta opción.	por medio de la función finish de java esta se encarga de restablecer la aplicación a su estado inicial, borrando cualquier rastro de cualquier actividad realizada en otra sesión.

Fuente: Autores del proyecto.

10.4.4. Manuales y documentación

El sistema multiplataforma Udecinv cuenta con dos manuales para su correcto funcionamiento y uso, el primero está enfocado hacia los usuarios del sistema multiplataforma, donde se les brinda una explicación detallada de cada uno de sus módulos con sus distintas opciones de una manera ilustrada con la interfaces de las plataformas de Udecinv, también se cuenta con el manual de programador que evidencia aspectos técnicos de desarrollo en el sitio web y móvil, los cuales son fundamentales para futuros cambios en el sistema o mantenimientos, en este se pueden encontrar ítems importantes de la codificación que se implementaron para su óptimo funcionamiento.

Resultados

Los resultados obtenidos por el cliente para el sistema multiplataforma Udecinv fueron positivos, ya que el sistema en sitio web y móvil cumplió con los requerimientos planteados en las historias de usuario, logrando cumplir con las expectativas de los usuarios, en el transcurso de cada iteración se pudo verificar que todos los módulos realizaron sus funciones requeridas.

En la tercera iteración de pruebas de aceptación se realizó una pequeña simulación de proceso de chequeo haciendo uso de sistema multiplataforma Udecinv, donde los resultados evidenciados se pudieron notar ya que el proceso de verificar cada uno de los elementos se realizó de una manera más eficaz y ágil con la implementación de lector código de barras por medio de la cámara de teléfono móvil, con gran felicidad el cliente se muestra satisfecho con el desarrollo de esta funcionalidad.

en consecuencia los resultados esperados se dieron, gracias a la implementación de una metodología que permite trabajar con el cliente durante todo el transcurso del proyecto, permitiendo que se logre solucionar la problemática que se estaba presentando en la gestión de inventarios, gracias a la versión de desarrollo en Android, mayor a 4.1 se cuenta con una gran cantidad de dispositivos compatibles para acceder a la aplicación por medio de la aplicación de play store con la que cuenta cada teléfono móvil con Android, de modo similar en el sitio web se puede acceder a este teniendo una conexión a internet desde cualquier parte del mundo facilitando su acceso a esta.

Conclusiones

- El proceso de identificación de requerimientos con el cliente permitió recopilar información importante como base principal para el desarrollo del sistema multiplataforma, dándonos una visión acertada del funcionamiento de la gestión de inventarios en la universidad de Cundinamarca seccional Ubaté.
- La fase de diseño fue de gran importancia para las distintas funciones con las que cuenta el sistema multiplataforma, gracias a que en ella se pudo definir aspectos relevantes como la arquitectura del sistema, modelo entidad-relación y las tarjetas crc, los cuales fueron esenciales para llevar a cabo las siguientes fases de la metodología implementada.
- El desarrollo del sistema en dos plataformas permite brindar mayores ventajas para el usuario, debido a el fácil acceso a la información, agilidad en el proceso de entregas de inventario gracias al uso de la cámara del teléfono para escanear el código de barras y la posibilidad de usar diferentes dispositivos para el proceso relacionado a la gestión de inventarios de la universidad.
- Gracias a la efectividad positiva obtenida en el proceso de pruebas se puede evidenciar que el sistema multiplataforma es una herramienta ampliamente útil para la gestión de inventarios permitiendo optimizar y agilizar estos procesos administrativos, de igual forma el sistema puede ampliarse con nuevas funcionalidades que logren servir para solucionar otras problemáticas que se reflejan en varios sectores administrativos de la universidad.

- La implementación de la metodología ágil de desarrollo xp brinda muchas ventajas para la realización de proyectos de software, debido a que el cliente tiene una amplia participación en cada fase, brindando las historias de usuario y sus prioridades dependiendo de la necesidad, dando como resultado una aceptación exitosa en la entrega del proyecto.
- Gracias a la implementación de la encuesta aplicada a las diferentes personas que han realizado el proceso de entregas de inventarios se logró conocer más a fondo la problemática planteada, permitiendo realizar un análisis y planeación del proyecto correcta, los cuales sirven de base fundamental para la realización de las siguientes fases de la metodología.
- El sistema multiplataforma Udecinv está desarrollado con el fin de que sirva como una base para que a futuro se pueda extender a todas las seccionales y extensiones de la universidad de Cundinamarca, permitiendo que el proceso de gestión de inventarios se centralice en un solo sistema donde se pueda realizar un seguimiento de esta información por parte de las directivas de la institución.

Referencias bibliográficas

- ¿Qué es el Código de Barras? – Visión LOGYCA. (n.d.). Retrieved October 23, 2019, from <http://blog.logyca.com/noticias/que-es-el-codigo-de-barras/>
- ¿Qué es inventario? Tipos, utilidad, contabilización y valuación - GestioPolis. (n.d.). Retrieved October 23, 2019, from <https://www.gestiopolis.com/que-es-inventario-tipos-utilidad-contabilizacion-y-valoracion/>
- ¿Qué es una aplicación Web? – Blog Neosoft Sistemas. (n.d.). Retrieved October 23, 2019, from <https://www.neosoft.es/blog/que-es-una-aplicacion-web/>
- ¿Qué son las bases de datos? (n.d.). Retrieved October 23, 2019, from <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- ¿Qué son las Tics o Tecnologías de la Información y la Comunicación? (n.d.). Retrieved October 23, 2019, from <http://tugimnasiacerebral.com/herramientas-de-estudio/que-son-las-tics-tic-o-tecnologias-de-la-informacion-y-la-comunicacion>
- Cátedra: Producción I Capítulo 9. Teoría de Inventarios. (2018.). (1), 1–19.
- Códigos de barras. (n.d.). Retrieved October 23, 2019, from <http://zaloamati.azc.uam.mx/handle/11191/2908>
- Definición de Multiplataforma (informática). (n.d.). Retrieved October 23, 2019, from <http://www.alegsa.com.ar/Dic/multiplataforma.php>
- Definición y cómo funcionan las aplicaciones móviles. (n.d.). Retrieved October 23, 2019, from <https://www.servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>

- Desarrollo de aplicaciones multiplataforma: web-apps vs. nativas | VIU. (n.d.). Retrieved October 23, 2019, from <https://www.universidadviu.com/desarrollo-de-aplicaciones-multiplataforma/>
- Flores, M. M. (2018). Reporte Final de Estadía. 70.
- Gutiérrez, V., & Patricia Jaramillo, D. (2009). Reseña del Software Disponible en Colombia Para la Gestión de Inventarios en Cadenas de Abastecimiento. *Estudios Gerenciales*, 25(110), 125–153. [https://doi.org/10.1016/s0123-5923\(09\)70065-8](https://doi.org/10.1016/s0123-5923(09)70065-8)
- La importancia de las apps móviles en las empresas - Economía TIC. (n.d.). Retrieved October 23, 2019, from <https://economiatic.com/importancia-apps-moviles-empresas/>
- López, S., López, V., Luna, R., Alicia, B., Vásquez, V., Blanca, I., ... Vidal, L. (2011). Sistema de información de control de inventarios del almacén del ITS. *Conciencia Tecnológica* ISSN:1405-5597. Retrieved from <https://www.redalyc.org/articulo.oa?id=94419100007>
- Oña L. (2013). Universidad Politécnica Salesiana Sede Quito. Tesis, 1–100. Retrieved from <http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>.
- ¿Qué es ¿Qué son las plataformas de desarrollo de código bajo y sin código?? - Definición en WhatIs.com. (n.d.). Retrieved March 10, 2020, from <https://searchdatacenter.techtarget.com/es/definicion/Que-son-las-plataformas-de-desarrollo-de-codigo-bajo-y-sin-codigo>
- Metodología XP Programación Extrema (Metodología ágil) - Diego Calvo. (n.d.). Retrieved March 10, 2020, from <http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>
- LUIS MIGUEL, & ELENA, L. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE SOFTWARE. *Ятыатат, вы12у* (235), 245.

- Fetrina, E., Rustamaji, E., Nuraeni, T., & Durrachman, Y. (2017). Inventory management information system development at BPRTIK KEMKOMINFO Jakarta. 2017 5th International Conference on Cyber and IT Service Management, CITSM 2017, 1–4. <https://doi.org/10.1109/CITSM.2017.8089303>
- INICIO - mystock. (n.d.). Retrieved March 28, 2020, from <https://mystock.mybook-mobile.com/>
- Lilia, M. E. J. A. C. T. D. (2017). SISTEMA DE CONTROL DE INVENTARIO MEDIANTE CÓDIGOS QR PARA EL GRUPO DE INVESTIGACIÓN TELETECNO INVENTORY. 28.
- Vázquez, R. O. (2016). Sistema de control de inventario en entorno web y dispositivos móviles con sistema operativo Android para la empresa Rodamientos Bower. Tesis, 1, 82. <http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>
- Vázquez Rubén Ordóñez. (2018). InvetDroid. 58.
- Zheng, Y., & Meng, X. (2010). An inventory management model in mobile commerce. Proceedings - 2010 International Conference of Information Science and Management Engineering, ISME 2010, 2, 11–14. <https://doi.org/10.1109/ISME.2010.35>

- Zheng, Y., & Meng, X. (2010). An inventory management model in mobile commerce. *Proceedings - 2010 International Conference of Information Science and Management Engineering, ISME 2010*, 2, 11–14. <https://doi.org/10.1109/ISME.2010.35>.
- Número máximo de registros en una tabla de base de datos | mysql*. (2019). Retrieved February 23, 2021, from <https://www.iteramos.com/pregunta/19859/numero-maximo-de-registros-en-una-tabla-de-base-de-datos-mysql>
- C-1183-00 Corte Constitucional de Colombia*. (n.d.). Retrieved February 23, 2021, from <https://www.corteconstitucional.gov.co/relatoria/2000/C-1183-00.htm>
- Ley. (n.d.). *CONGRESO DE LA REPÚBLICA*.
- Ley 23 de 1982 Nivel Nacional*. (n.d.). Retrieved February 23, 2021, from <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=3431>
- Normograma del Ministerio de Tecnologías de la Información y las Comunicaciones [LEY_1581_2012]*. (n.d.). Retrieved January 22, 2021, from https://normograma.mintic.gov.co/mintic/docs/ley_1581_2012.htm
- CyTA*. (n.d.). Retrieved March 11, 2021, from <http://www.cyta.com.ar/ta0502/v5n2a1.htm>

Anexos

Anexo 1. Resultado de encuesta de viabilidad del proyecto. (Ver en carpeta).

Anexo 2. Resultado de pruebas de aceptación por el cliente. (Ver en carpeta).

Anexo 3. Actas de reuniones. (Ver en carpeta).

Anexo 4. Manual de usuario. (Ver en carpeta).

Anexo 5. Manual de programador. (Ver en carpeta).

Anexo 6. Video de reuniones. (Ver en carpeta).

Anexo 7. Copia de la base de datos Mysql. (Ver en carpeta).

Manual de usuario para sistema multiplataforma UDECVN



Tabla de contenido

1	Introducción.....	2
2	Sitio Web.....	2
2.1	Especificaciones mínimas requeridas para el sitio web.....	2
2.2	Inicio	2
2.3	Iniciar sesión “Login” sitio web	2
2.4	Acerca de	3
3	Usuario administrador sitio web	3
3.1	Pantalla principal usuario administrador	3
3.2	Acceso a lista de usuarios	4
3.3	Acceso a entregas de inventario	5
3.3.1	Nuevo checklist.	6
3.3.2	Reportes.....	7
3.3.3	Realizar chequeo	7
3.4	Acceso a inventarios.	8
3.4.1	Agregar nuevo inventario.....	9
3.4.2	Modificar inventario.....	10
3.4.3	Agregar nuevo elemento.....	11
3.4.4	Asignar inventario a nuevo encargado.....	12
3.4.5	Estado de los elementos.	12
3.4.6	Información del elemento.....	13
3.4.7	Ver y descargar código de barras del elemento.....	14
3.4.8	Modificar elemento.....	14
3.4.9	Trasladar elemento.	14
3.4.10	Actualizar foto del elemento.....	15
3.5	Acceso a encargados.....	16
3.5.1	Nuevo encargado.....	17
3.5.2	Modificar encargado.....	18
3.6	Acceso a solicitudes.....	18
4	Usuario dependencia sitio web.....	19
4.1	Pantalla principal usuario dependencia	19
4.1.1	Ver lista de inventarios.....	20
4.1.2	Ver información del elemento.....	20

4.1.3	Crear nueva solicitud.....	21
4.1.4	Ver y descargar código de barras del elemento.	21
4.1.5	Ver imagen del elemento	22
4.2	Acceso solicitudes realizadas.	23
5	Aplicativo móvil.....	23
5.1	Instalar aplicación.	23
5.2	Especificaciones mínimas para aplicativo móvil.....	24
5.3	Iniciar sesión “Login” app.....	24
5.4	Acerca de	24
6	Usuario administrador app.	25
6.1	Pantalla de inicio usuario administrador.....	25
6.2	Acceso a entregas de inventario.	26
6.2.1	Chequeo de elementos	26
6.3	Acceso a buscar elementos.	27
6.4	Acceso a artículos inventario.	29
6.4.1	Añadir nuevo elemento.....	29
6.4.2	Actualizar elemento.	30
6.4.3	Cambio de encargado.....	31
6.4.4	Trasladar elemento de inventario.	32
6.4.5	Dar de baja.	33
6.5	Acceso a solicitudes.	34
6.5.1	Responder solicitud.....	34
7	Usuario dependencia app.	35
7.1	Pantalla de inicio usuario dependencia.....	35
7.2	Acceso a buscar elementos.	36
7.3	Acceso a solicitudes.	37
7.3.1	Crear nuevo solicitud.	38
7.4	Solicitudes realizadas	39

1 Introducción.

Dentro de este manual se encuentra un explicativo sobre las distintas funciones con las que cuenta el sistema multiplataforma Udecinv para los dos roles tanto en sitio web como aplicativo móvil, se encontrara una explicación detallada sobre los pasos a realizar sobre las distintas opciones con las que cuenta cada módulo.

2 Sitio Web.

2.1 Especificaciones mínimas requeridas para el sitio web

Para que el sistema multiplataforma en sitio web tenga un funcionamiento en óptimas condiciones se recomienda contar con un computador que tenga instalado un sistema operativo con una versión superior a windows 7, así mismo tener una buena conexión a internet y navegadores como Google Chrome, Mozilla Firefox, Microsoft Edge, opera y zafarí.

2.2 Inicio

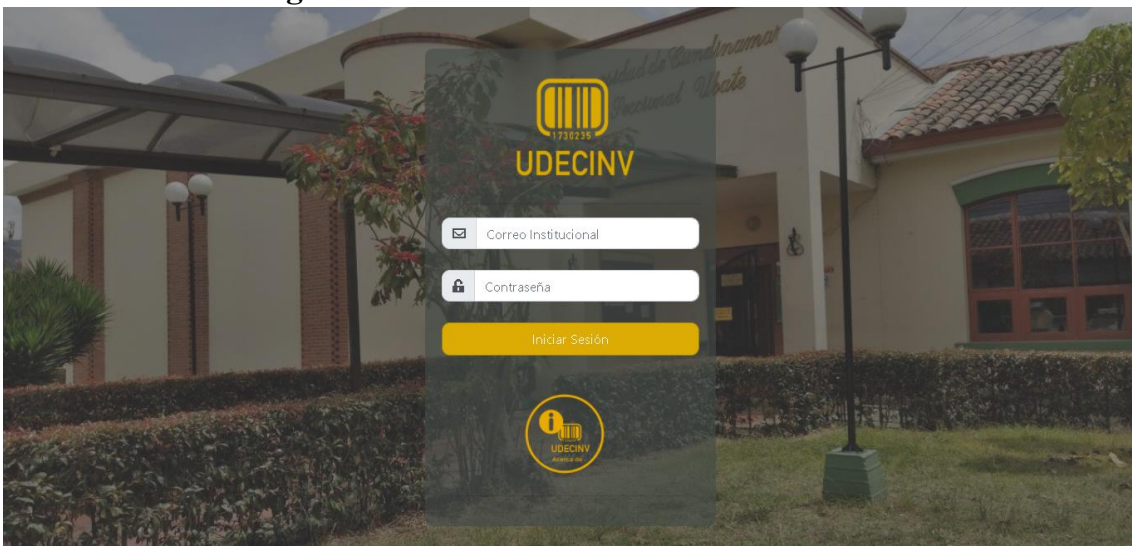
Para el acceso al sistema multiplataforma en sitio web es fundamental tener acceso a internet, se puede acceder mediante el escaneo del siguiente código QR o con la siguiente URL:

<https://udecinv.herokuapp.com>



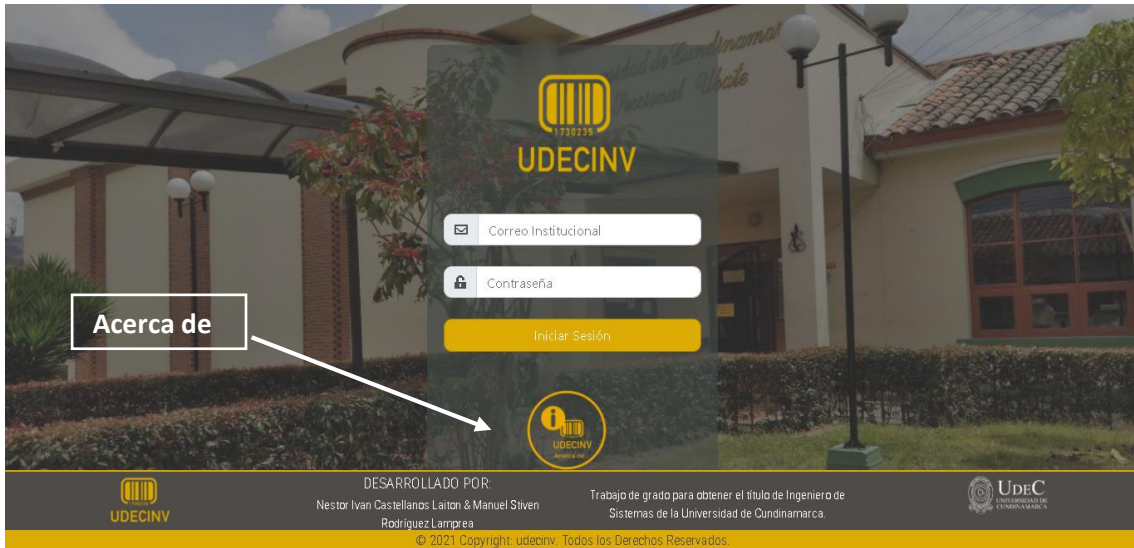
Qr para acceder al sitio web.

2.3 Iniciar sesión “Login” sitio web



Para acceder al sitio web es obligatorio identificarse, para ello se deberá ingresar el correo institucional y la contraseña dada por el administrador, en el caso que los datos sean erróneos se visualizará un mensaje con la información que es errónea.

2.4 Acerca de

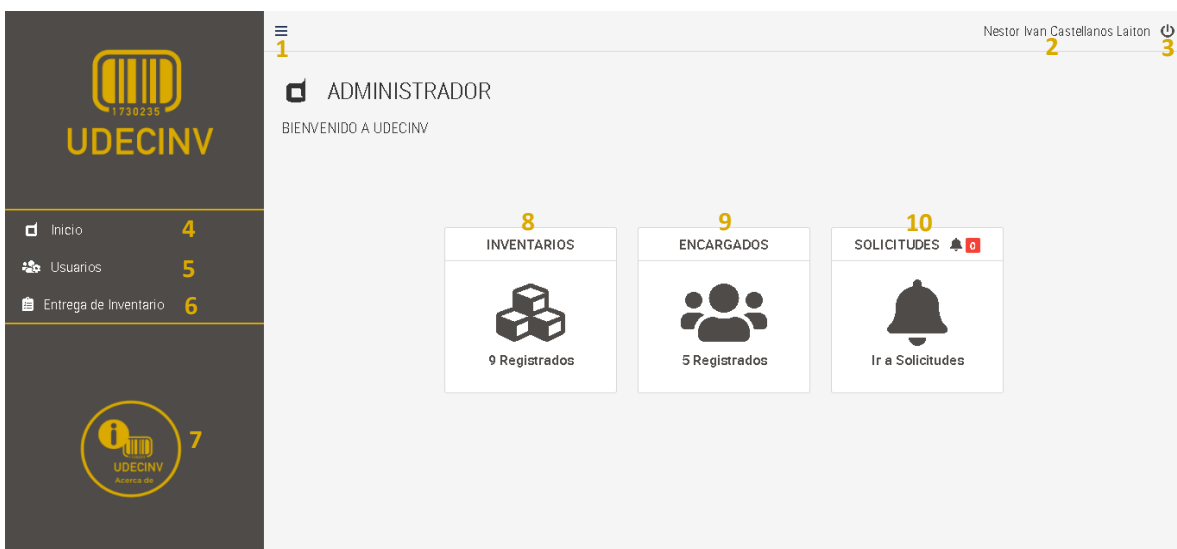


En la parte inferior del sistema en sitio web se encuentra un botón donde se puede visualizar la información sobre el sistema multiplataforma y sus desarrolladores.

3 Usuario administrador sitio web

3.1 Pantalla principal usuario administrador

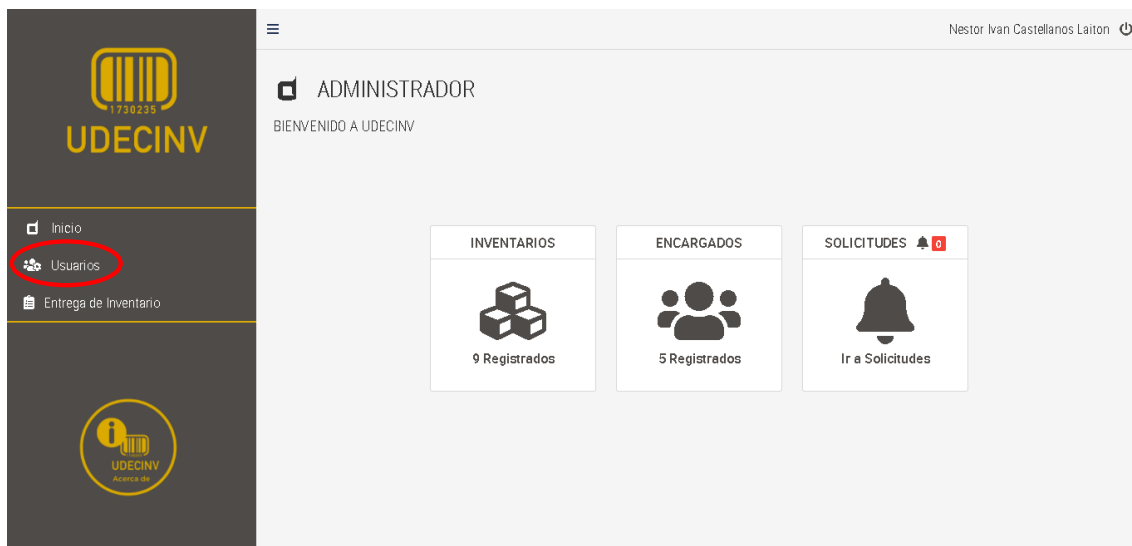
Esta pantalla permite al administrador acceder a las diferentes funcionalidades que el sistema suministra para su uso en este rol.

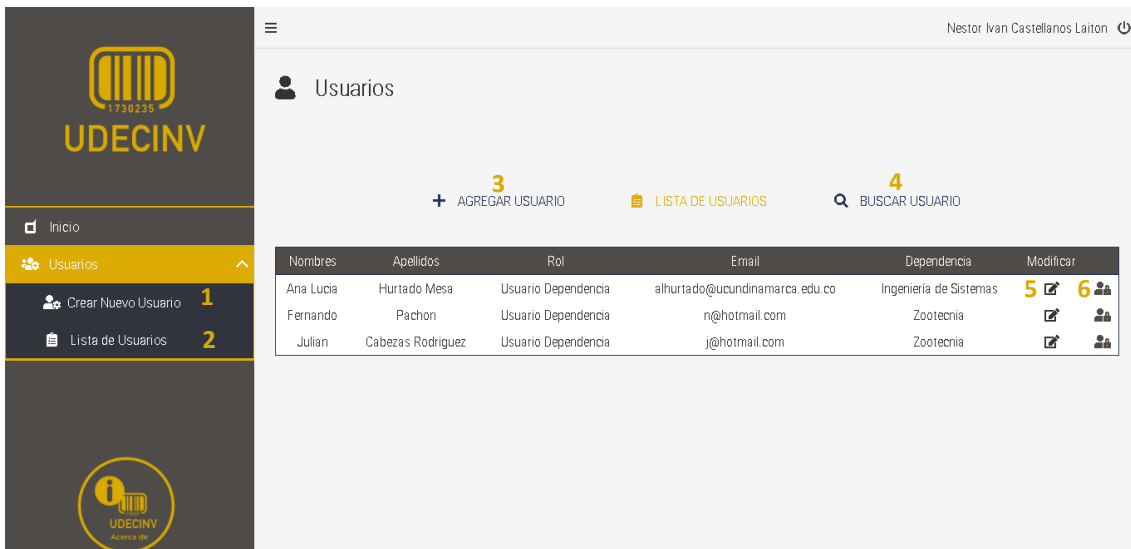


Al iniciar la sesión en el usuario administrador se visualiza la primera interfaz la cual cuenta con una serie de opciones que se pueden evidenciar a continuación:

1. Menú lateral izquierdo.
2. Identificador de usuario.
3. Botón para cerrar sesión.
4. Botón Inicio.
5. Botón acceso a lista de usuarios.
6. Botón para entregas de inventarios.
7. Botón acerca de.
8. Botón acceso a inventarios.
9. Botón acceso a encargados.
10. Botón acceso a solicitudes.

3.2 Acceso a lista de usuarios

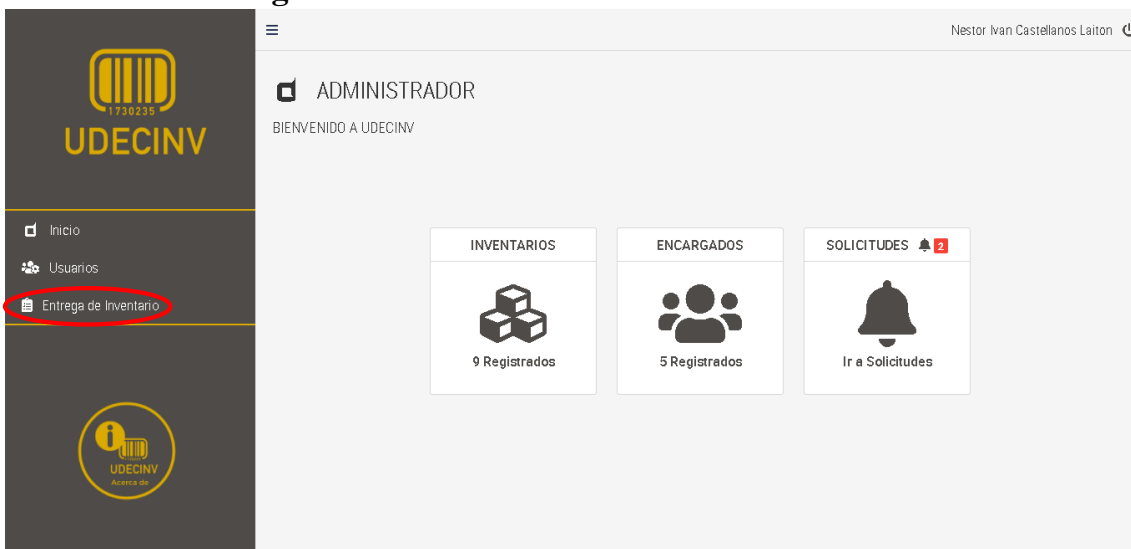


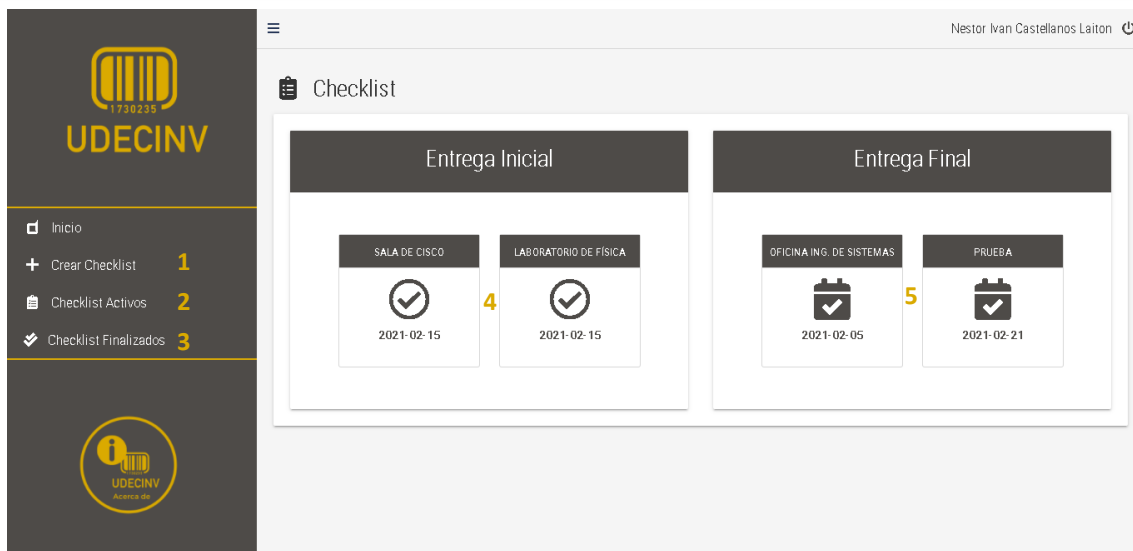


Menú de lista de usuarios

1. Botón crear nuevo usuario: este permite abrir un formulario en modal que requiere la información de un nuevo usuario para ser almacenado.
2. Botón listo de usuarios: al presionar este botón se recarga la página actual.
3. Botón agregar usuario permite crear un nuevo usuario.
4. Botón buscar usuario: al presionar en este botón se abrirá un buscador que permite filtrar la información de los usuarios.
5. Botón Modificar: permite modificar la información de los usuarios.
6. Botón hacer administrador: este botón permite hacer usuario administrador a un usuario dependencia.

3.3 Acceso a entregas de inventario

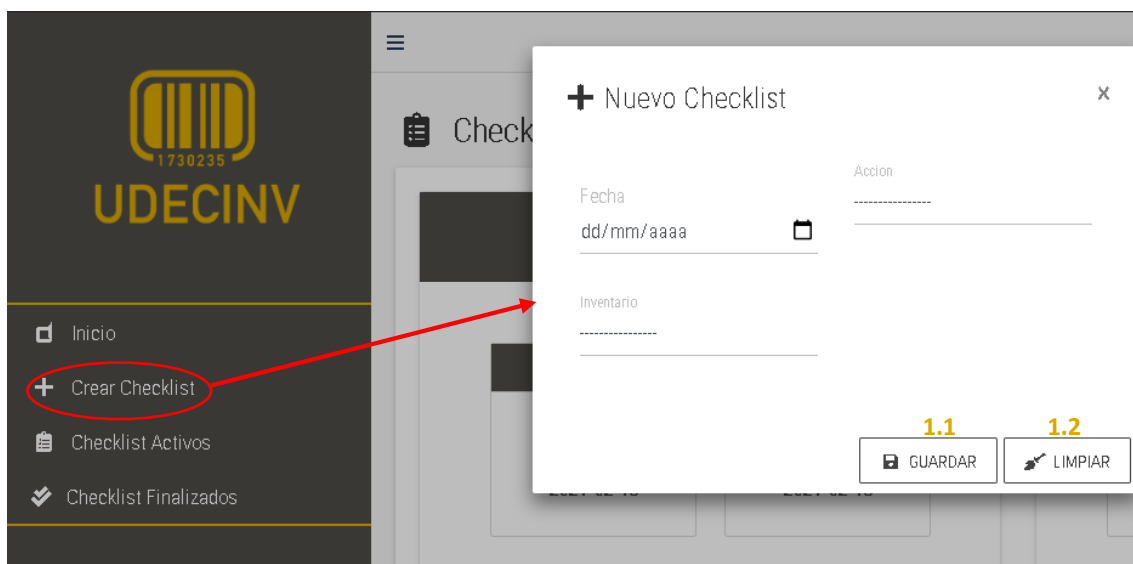




Menú de entregas de inventarios

1. Botón de crear checklist: este sirve para crear un nuevo checklist, para realizar el proceso de chequeo de los elementos de los inventarios.
2. Botón para ver checklist activos: este sirve para visualizar que inventarios están activos para realizar el proceso de entregas, de igual manera al dar clic sobre el checklist se abre la pantalla de chequeo.
3. Botón checklist finalizados: permite visualizar los inventarios que van han sido chequeados y abrir la opción de ver el reporte.
4. Botones de checklist activos de entrega inicial: estos permiten abrir cada inventario para realizar la entrega inicial de los elementos.
5. Botones de checklist activos de entrega Final: estos permiten abrir cada inventario para realizar la entrega final de los elementos.

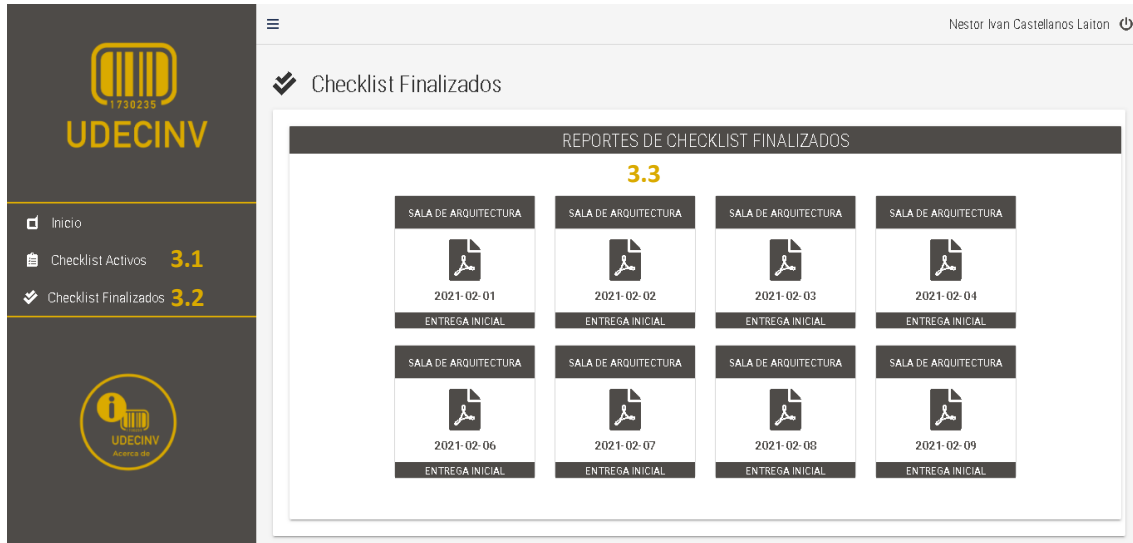
3.3.1 Nuevo checklist.



- 1.1. Botón guardar: permite guardar la información suministrada en el formulario.

1.2. Botón limpiar: permite borrar los datos ingresados en los campos del formulario.

3.3.2 Reportes.

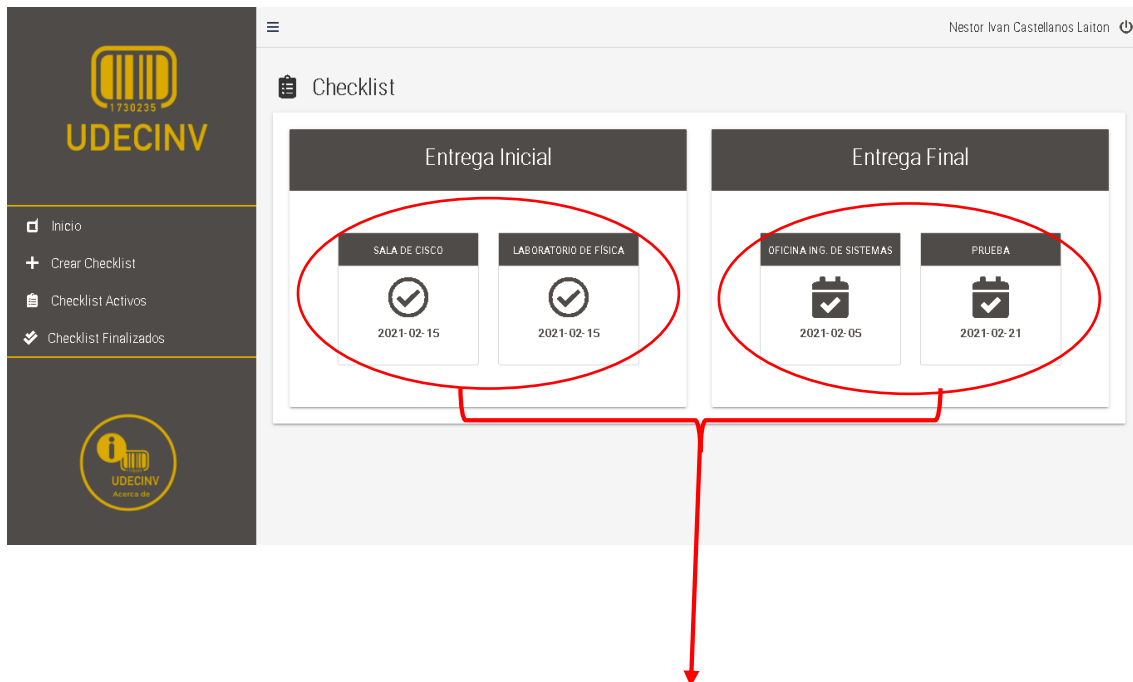


3.1. Botón checklist activos: permite regresar a la página anterior.

3.2. Botón checklist finalizados: permite recargar la página actual.

3.3. Botones de los checklist finalizados: permite seleccionar e ingresar al pdf con los reportes de cada checklist.

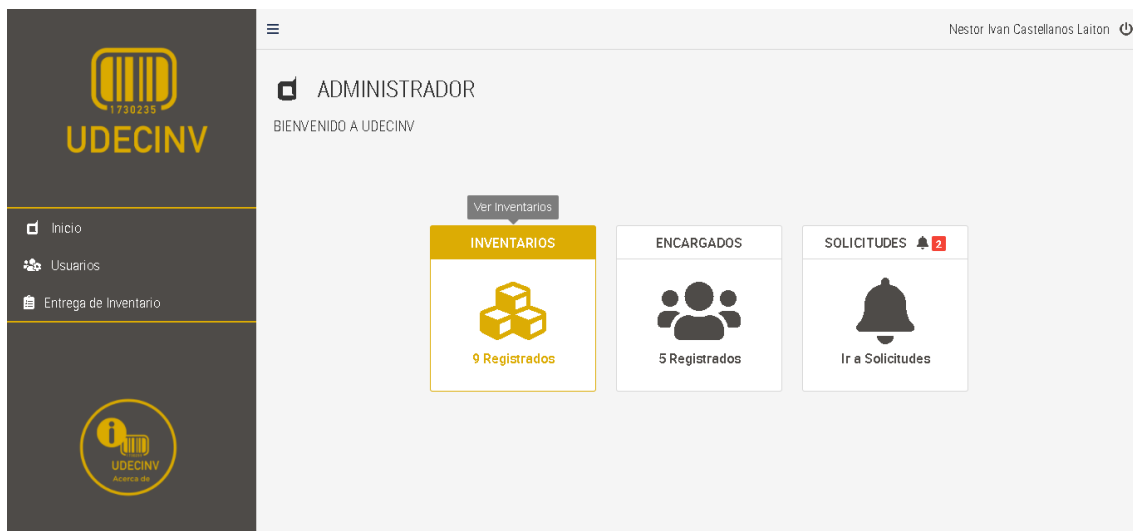
3.3.3 Realizar chequeo.

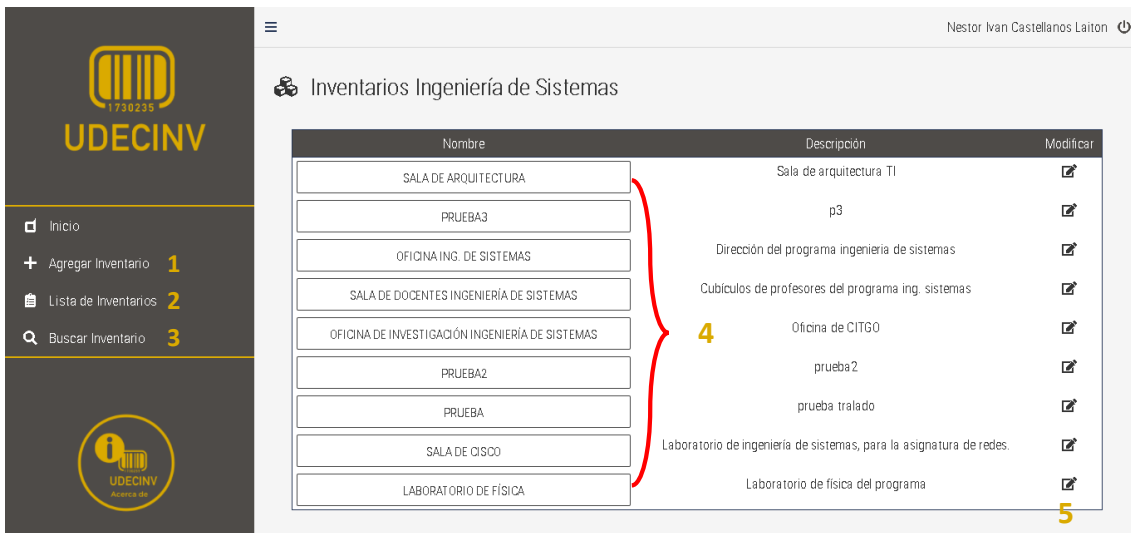




- 4.1. Botón checklist: permite regresar a la página anterior donde están todos los checklist activos.
- 4.2. Botón chequear: permite chequear la placa ingresada en el formulario.
- 4.3. Botones limpiar: al darle en este botón se eliminará la información ingresada en el formulario.
- 4.4. Botón Finalizar chequeo: permite finalizar el chequeo activo.
- 4.5. Botón ver: permite visualizar la imagen e información específica de cada elemento.

3.4 Acceso a inventarios.

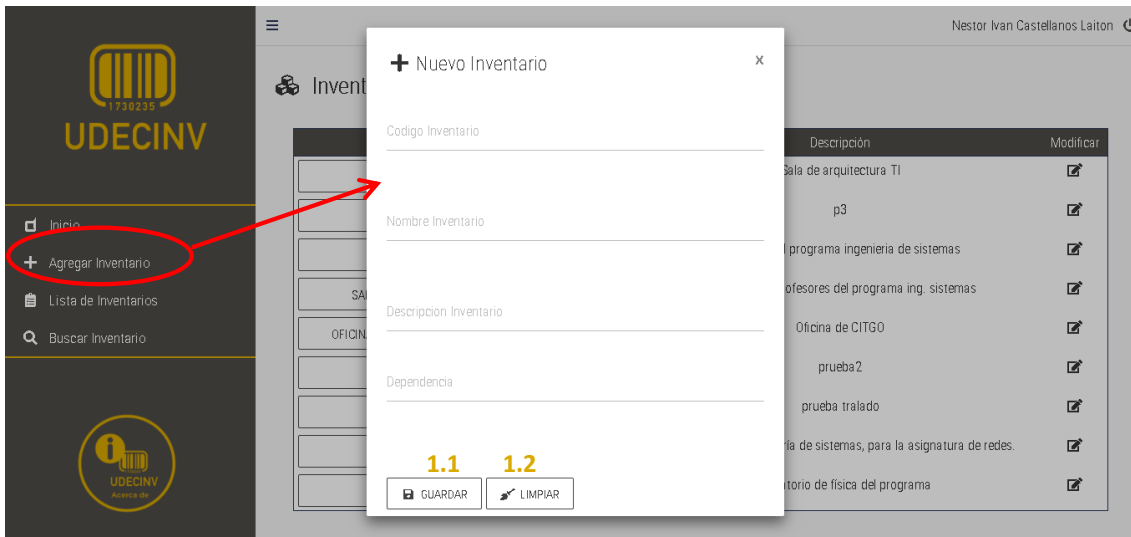




Menú de inventarios

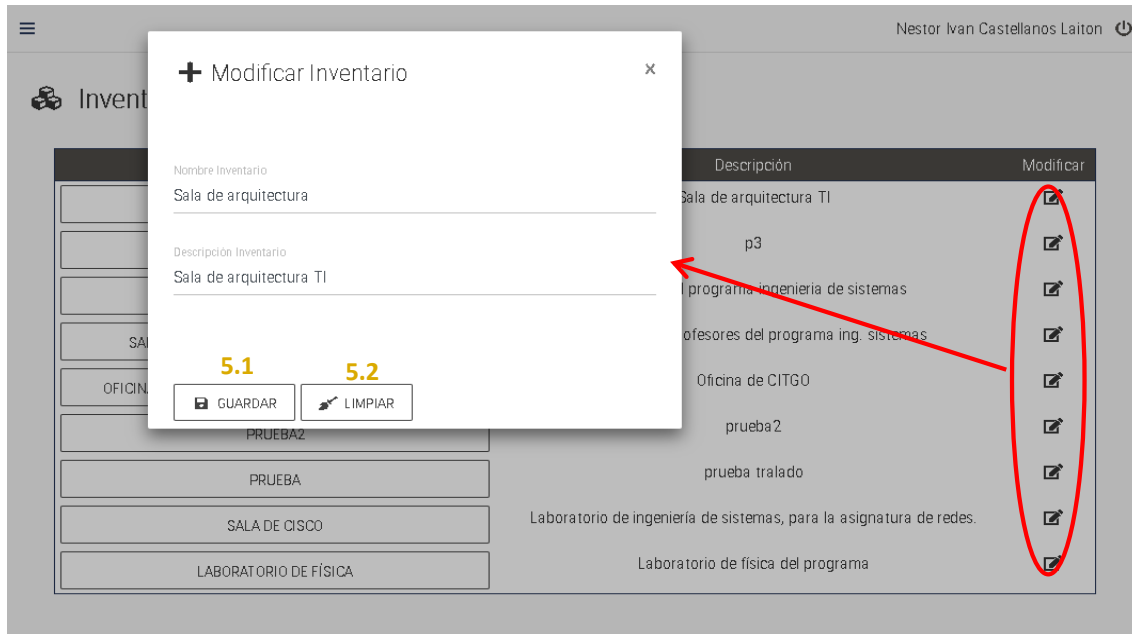
1. Botón agregar inventario: este botón permite abrir un formulario para ingresar un nuevo inventario.
2. Botón de lista de inventarios: este botón recarga la página actual.
3. Botón buscar inventario: este botón permite abrir un buscador para filtrar los diferentes inventarios.
4. Botones de inventarios: estos permiten abrir cada inventario disponible.
5. Botón modificar inventario: estos modificar datos de los inventarios.

3.4.1 Agregar nuevo inventario.



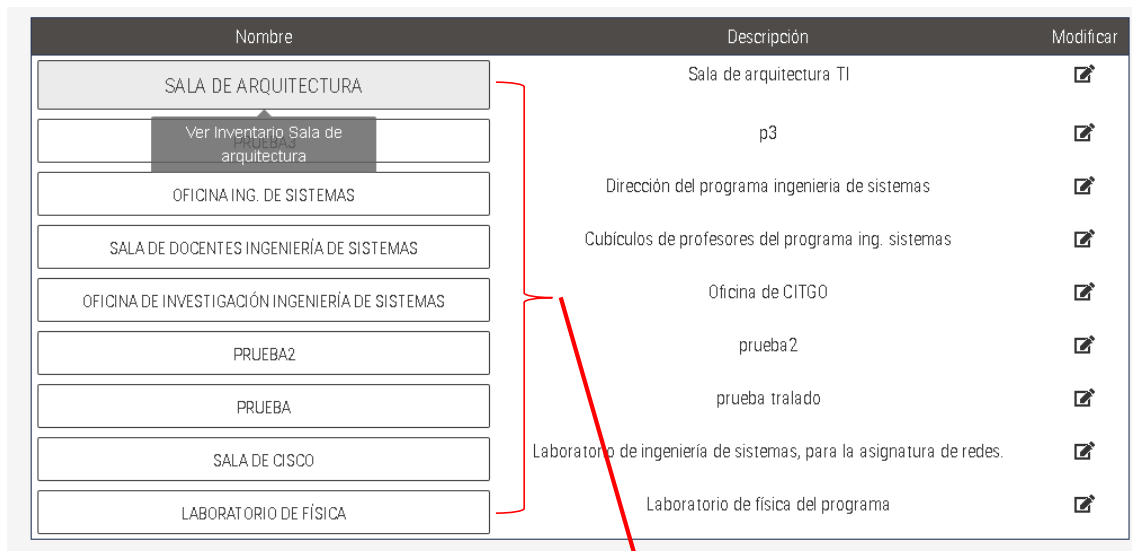
- 1.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.
- 1.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

3.4.2 Modificar inventario.



5.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

5.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

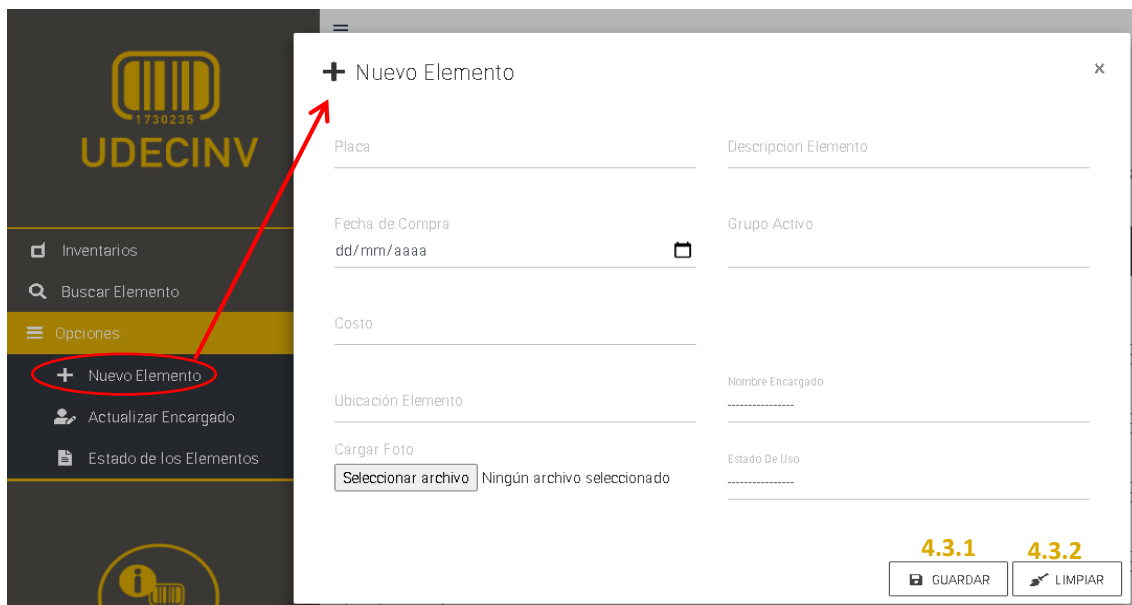




Menú elementos

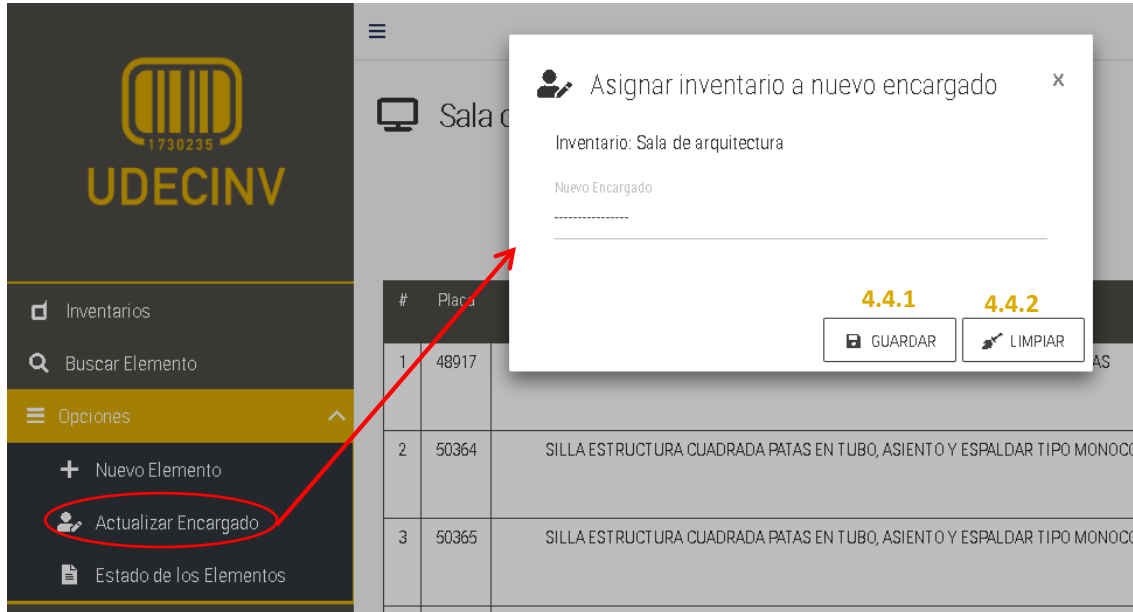
- 4.1. Botón Inventarios: Permite volver a la página anterior.
- 4.2. Botón buscar elemento: este botón permite acceder al buscador de elementos
- 4.3. Botón nuevo elemento: permite ingresar al formulario para registrar nuevos elementos.
- 4.4. Botón Actualizar encargado: este botón accede a asignar el inventario a un encargado.
- 4.5. Botón estado de los elementos: permite acceder a la página de lista del estado de los elementos.
- 4.6. Botón Ver: permite ver la información mas detallada de cada elemento.

3.4.3 Agregar nuevo elemento.



- 4.3.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.
- 4.3.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

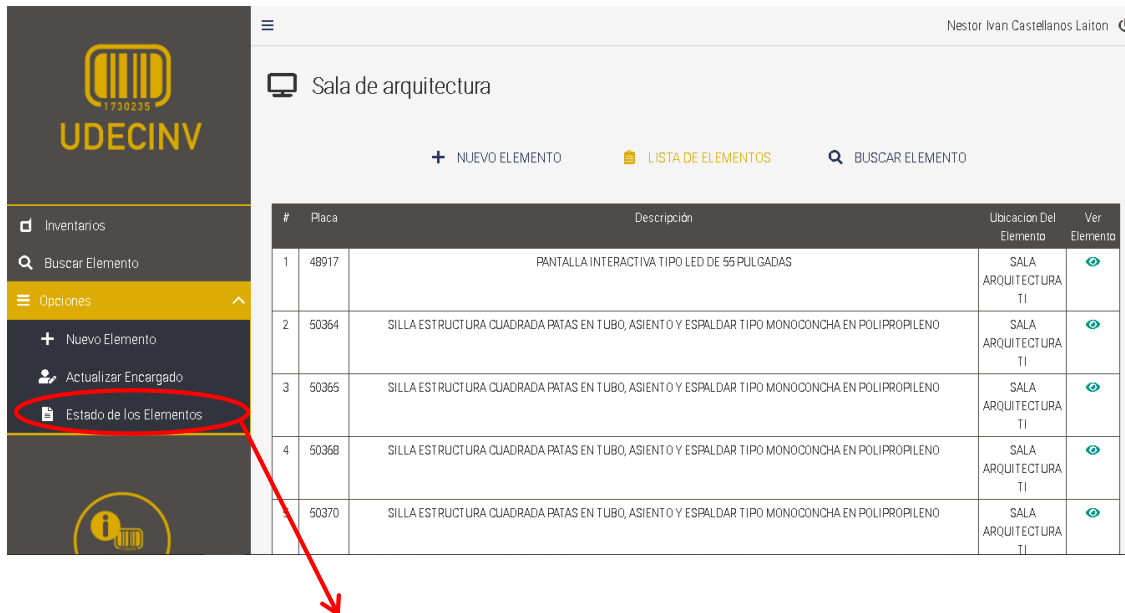
3.4.4 Asignar inventario a nuevo encargado.



4.4.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

4.4.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

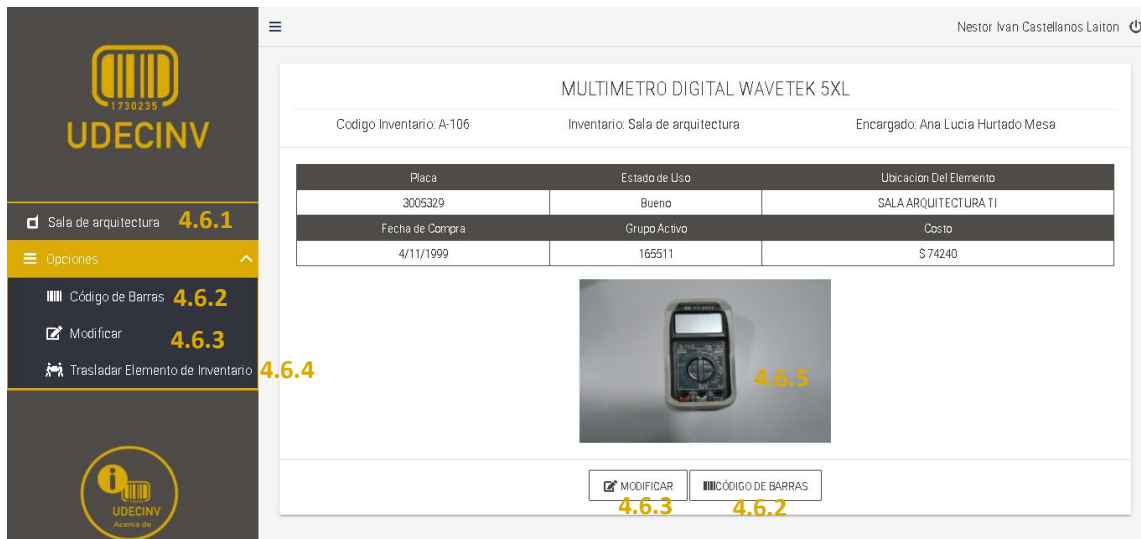
3.4.5 Estado de los elementos.





- 4.5.1. Botón Inventario: permite volver a la página anterior.
- 4.5.2. Botón Estado: permite ocultar o expandir tablas de elementos en estado bueno.
- 4.5.3. Botón Estado: permite ocultar o expandir tablas de elementos en estado malo.
- 4.5.4. Botón Estado: permite ocultar o expandir tablas de elementos en estado regular.
- 4.5.5. Botón Estado: permite ocultar o expandir tablas de elementos en mantenimiento.

3.4.6 Información del elemento.

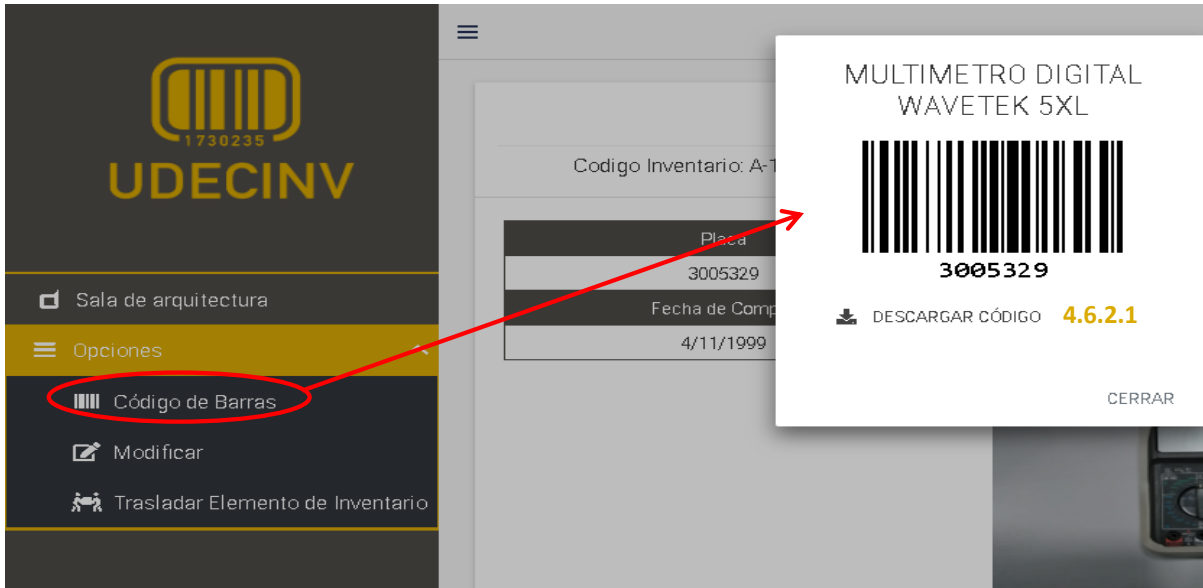


Menú elementos

- 4.6.1. Botón inventario: volver a página anterior.
- 4.6.2. Botón código de barras: este botón permite generar un código de barras que podrá ser descargado.
- 4.6.3. Botón modificar: permite abrir un formulario para modificar la información del elemento.
- 4.6.4. Botón trasladar elemento de inventario: permite abrir un formulario para asignar elemento a otro inventario.

4.6.5. Botón imagen: permite abrir las opciones para ampliar imagen y modificar imagen.

3.4.7 Ver y descargar código de barras del elemento.



4.6.2.1. Botón descargar código: este botón permite descargar el código de barras en una imagen.

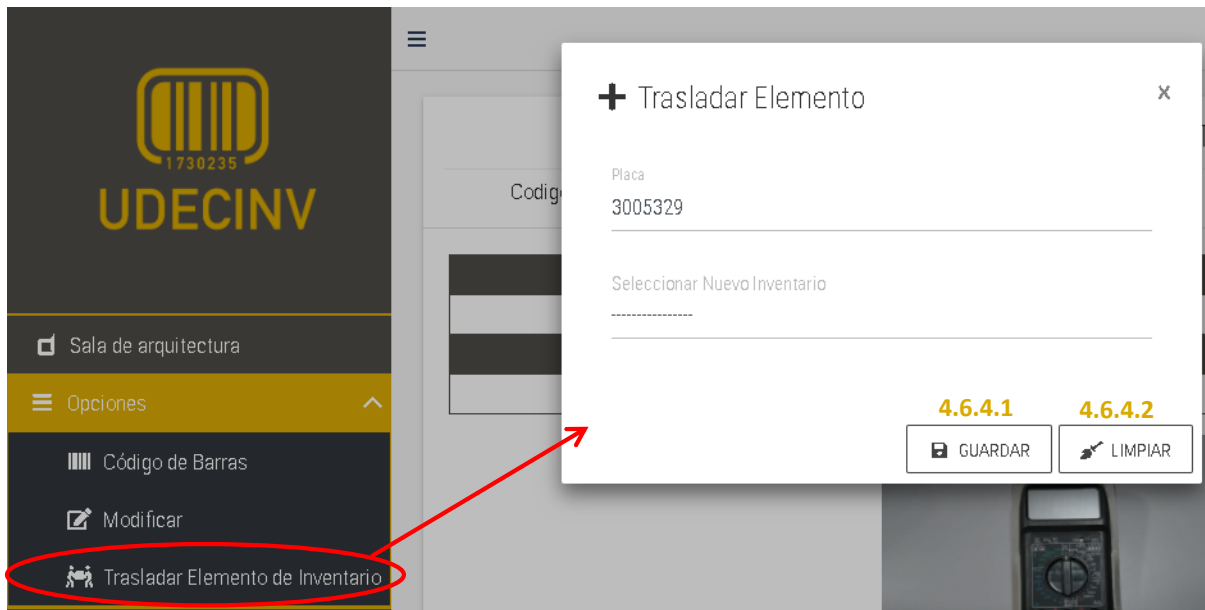
3.4.8 Modificar elemento.



4.6.3.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

4.6.3.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

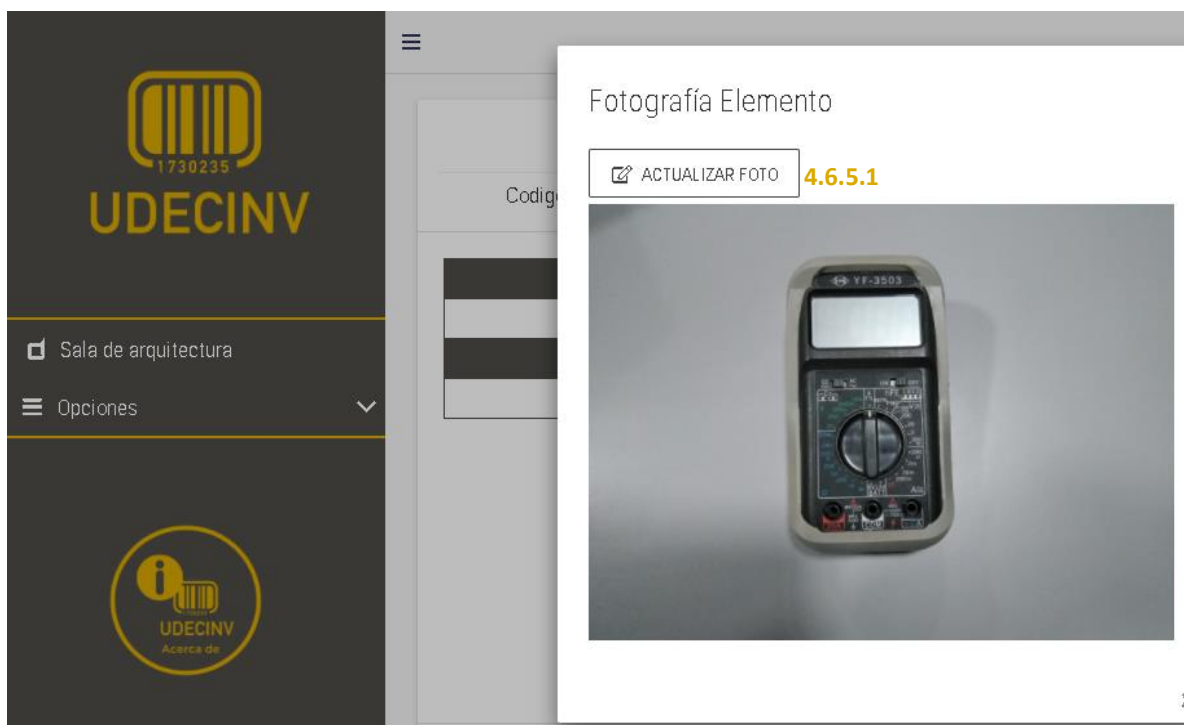
3.4.9 Trasladar elemento.



4.6.4.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

4.6.4.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

3.4.10 Actualizar foto del elemento.



4.6.5.1. Actualizar foto: permite dar opciones para actualizar la foto.

Fotografía Elemento

 ACTUALIZAR FOTO

Placa
3005329

 Seleccionar archivo Ningún archivo seleccionado

4.6.5.1.1

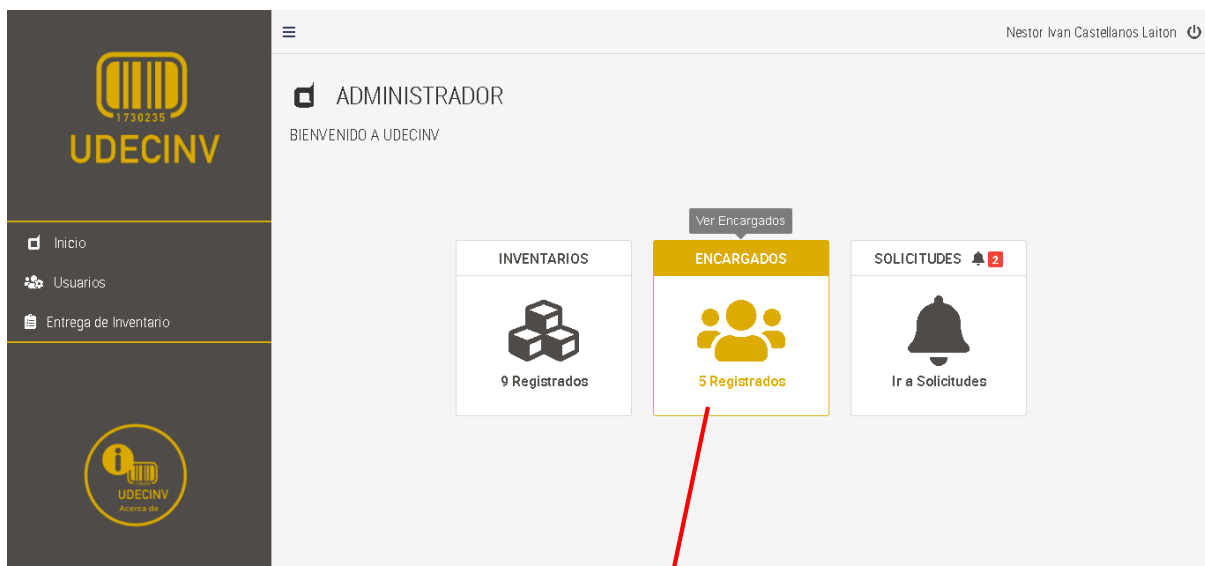


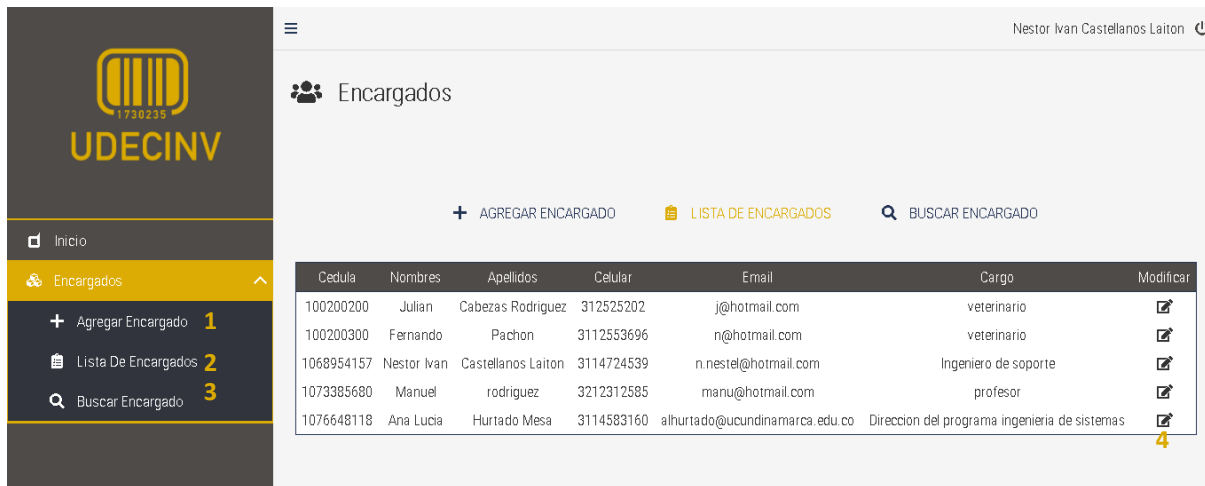
 GUARDAR **4.6.5.1.2**

4.6.5.1.1. Botón seleccionar archivo: permite elegir un archivo de imagen para cargarlo.

4.6.5.1.2. Botón guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

3.5 Acceso a encargados.

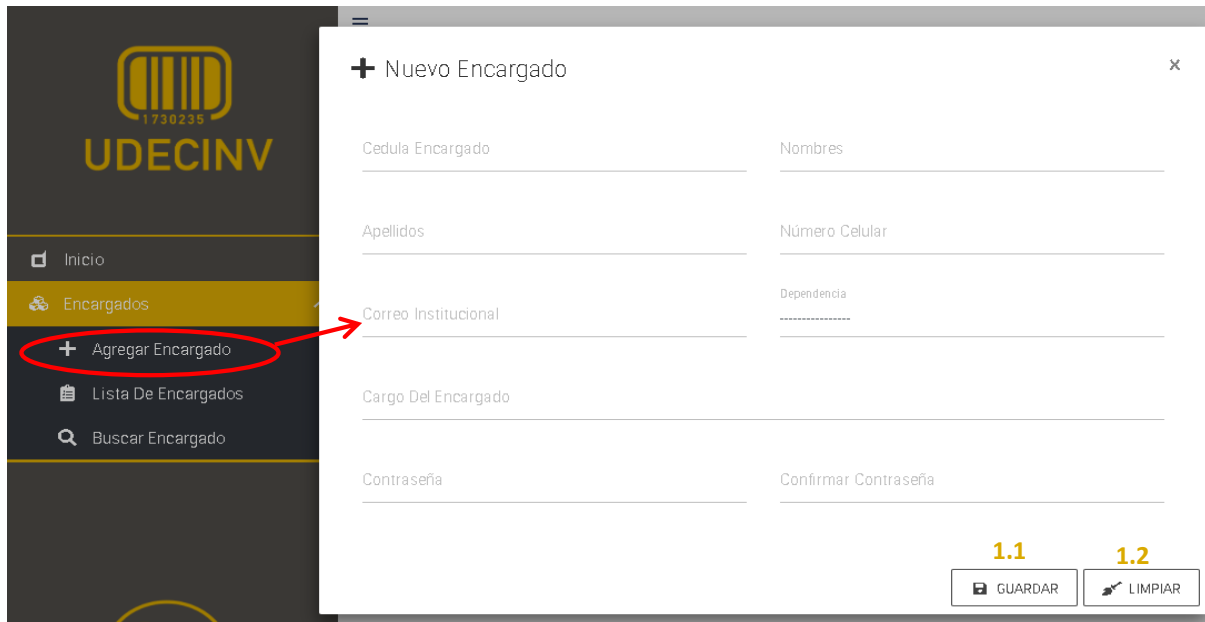




Menús encargados

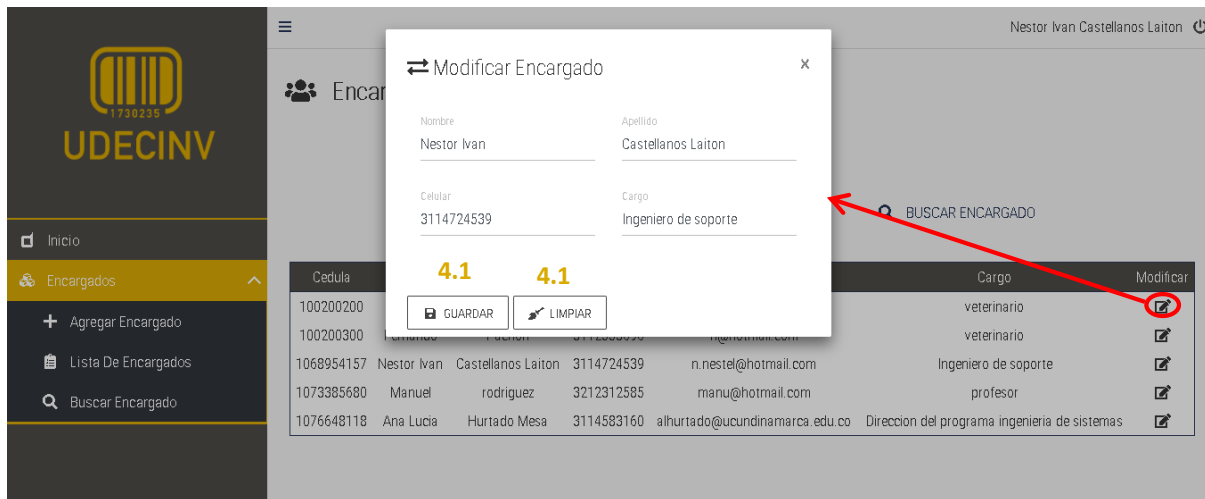
1. Botón agregar encargado: permite acceder al formulario de nuevo encargado.
2. Botón de lista de encargados: este botón recarga la página actual.
3. Botón buscar encargado: permite ingresar al formulario para buscar por filtros el encargado.
4. Botón modificar: este botón accede a un formulario para modificar el encargado.

3.5.1 Nuevo encargado.



- 1.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.
- 1.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

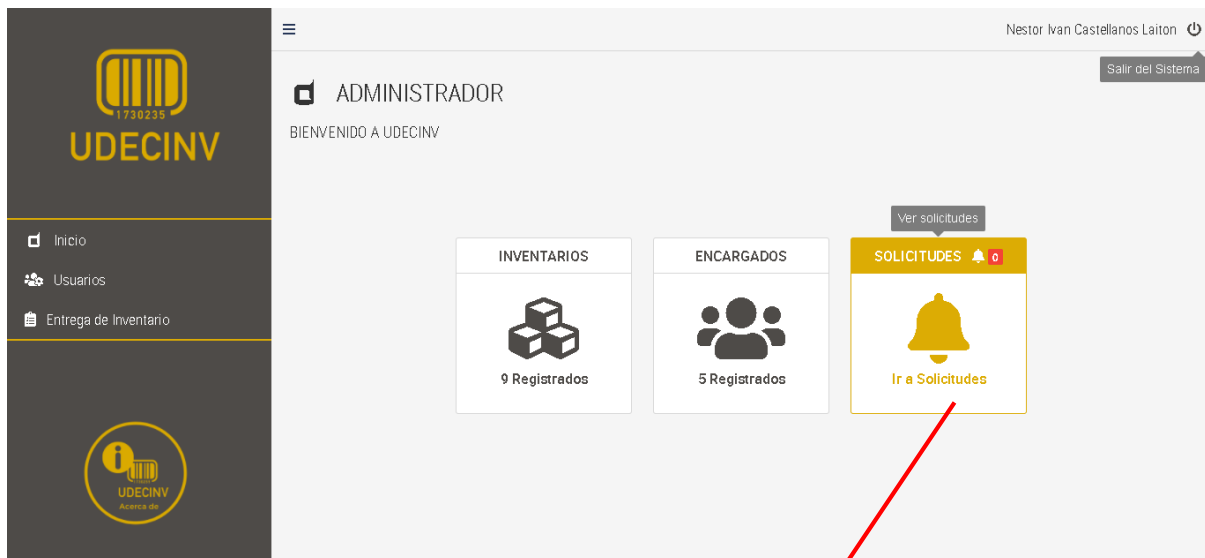
3.5.2 Modificar encargado.



4.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario.

4.2. Botón limpiar: al darle en este botón se eliminará la información ingresada en el formulario.

3.6 Acceso a solicitudes.





Menú solicitudes

1. Botón de lista de solicitudes: permite recargar la página.
2. Botón buscar solicitud: permite abrir el buscador y filtrar las solicitudes con una palabra clave.
3. Botón solucionar: permite ingresar al formulario para responder la solicitud.
4. Botón eliminar: este botón elimina la solicitud seleccionada.

4 Usuario dependencia sitio web.

4.1 Pantalla principal usuario dependencia

Esta pantalla permite al usuario dependencia acceder a las diferentes funcionalidades que el sistema suministra para su uso en este rol.





Al iniciar la sesión en el usuario dependencia se visualiza la primera interfaz la cual cuenta con una serie de opciones que se pueden evidenciar a continuación:

1. Botón Inicio.
2. Botón solicitudes realizadas.
3. Botón inventarios asignados.

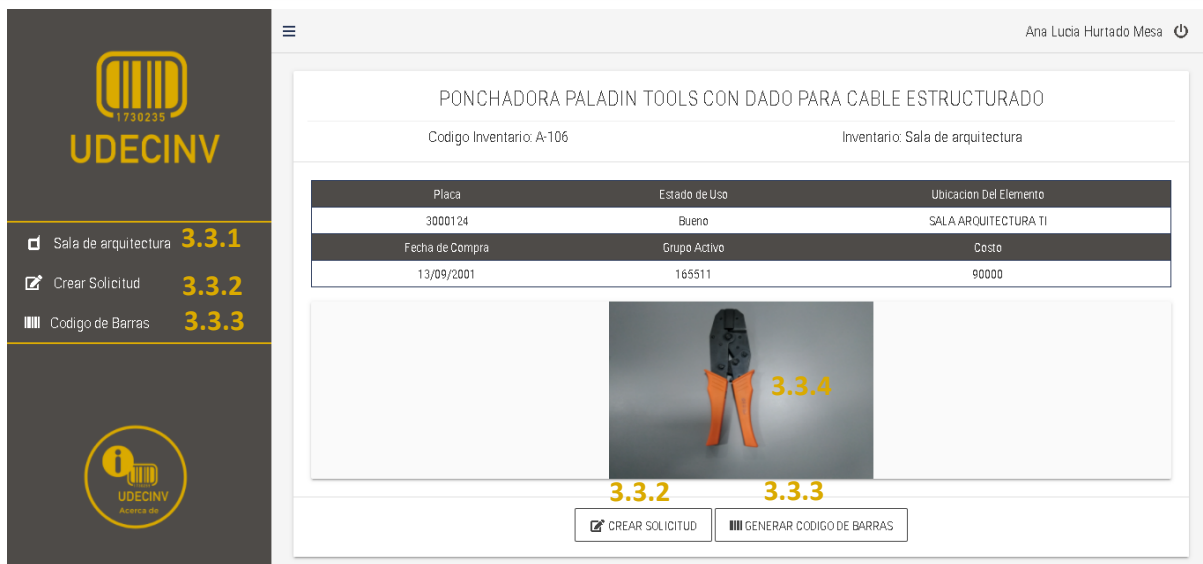
4.1.1 Ver lista de inventarios.



#	Placa	Descripción	Ubicación Del Elemento	Ver Elemento
1	48917	PANTALLA INTERACTIVA TIPO LED DE 55 PULGADAS	SALA ARQUITECTURA TI	 3.3
2	50364	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	
3	50365	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	
4	50368	SILLA ESTRUCTURA CUADRADA PATAS EN TUBO, ASIENTO Y ESPALDAR TIPO MONOCONCHA EN POLIPROPILENO	SALA ARQUITECTURA TI	

- 3.1. Botón mis inventarios: permite retornar a la página anterior.
- 3.2. Botón Buscar elemento: permite filtrar con una palabra clave los elementos.
- 3.3. Botón ver: permite ver detalladamente cada elemento.

4.1.2 Ver información del elemento.



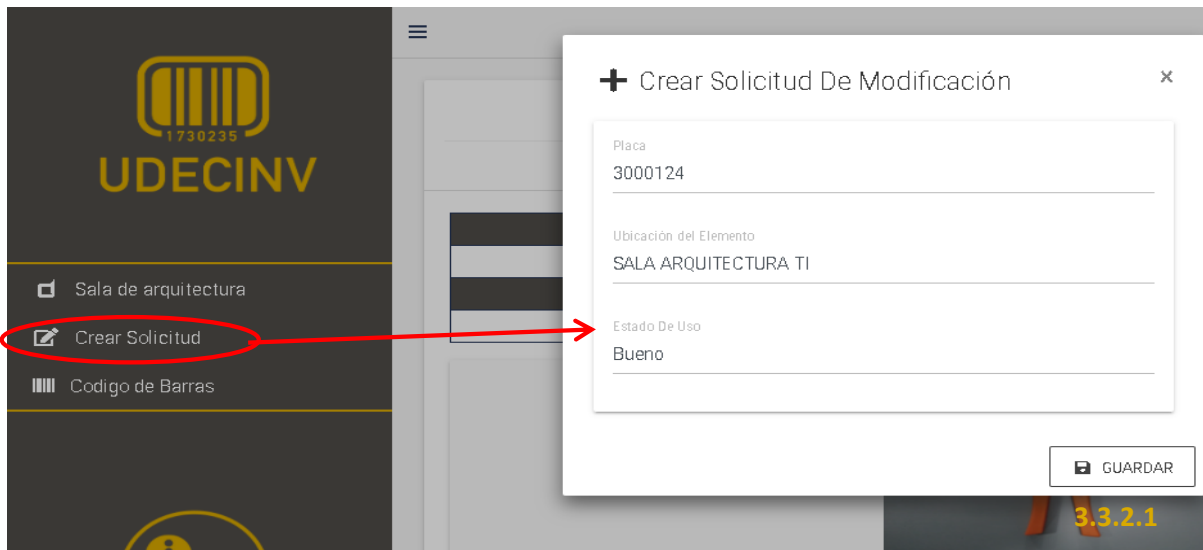
3.3.1. Botón inventario: permite volver a la página anterior.

3.3.2. Botón crear solicitud: permite abrir el formulario para hacer la solicitud.

3.3.3. Botón código de barras: permite generar el código de barras para después descargarlo.

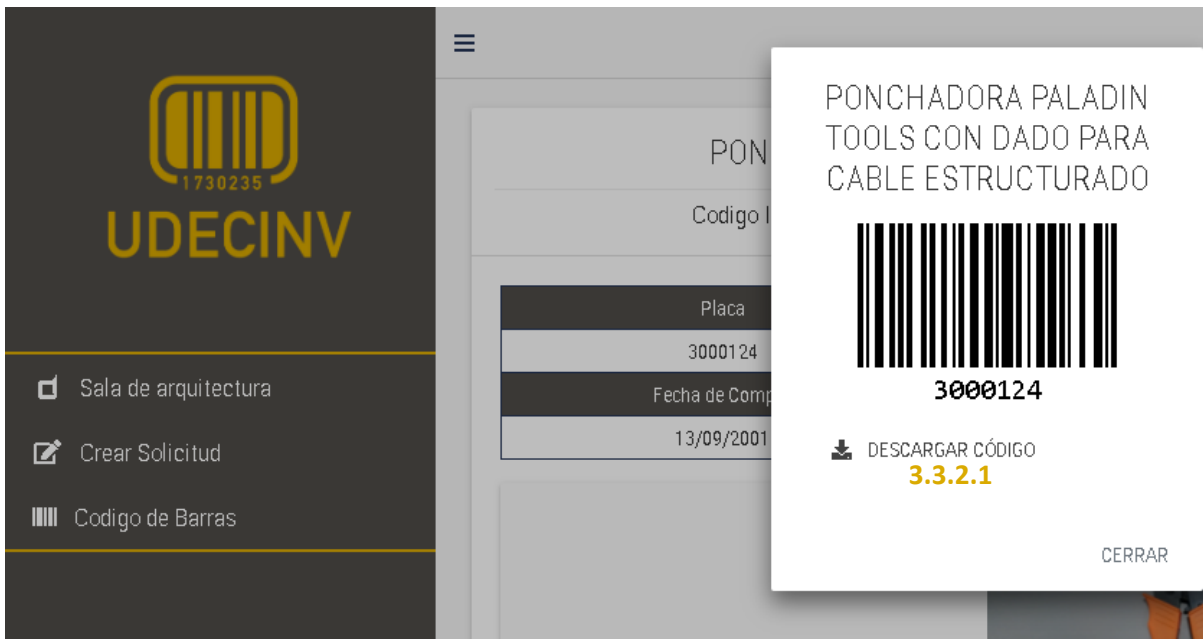
3.3.4. Botón imagen: permite ampliar la imagen.

4.1.3 Crear nueva solicitud.



3.3.2.1. Botón Guardar: permite guardar en la base de datos la información suministrada por el usuario en el formulario de crear solicitud.

4.1.4 Ver y descargar código de barras del elemento.



6.3.3.1. Botón descargar código: este botón permite descargar el código de barras en una imagen.

4.1.5 Ver imagen del elemento



En la imagen anterior se evidencia como al dar click en la opción 3.3.4 se amplía la imagen.

4.2 Acceso solicitudes realizadas.



Solicitudes Pendientes

Fecha de Realización	Tipo de Solicitud	Descripción	Eliminar
2021-02-25	Cambio de Lugar Físico	Cambio de Lugar Físico de la placa 48917 de SALA ARQUITECTURA TI a SALA ARQUITECTURA TI	 2.3
2021-02-25	Cambio de estado de uso	Cambio de estado de uso de placa 48917 de Malo a Bueno	

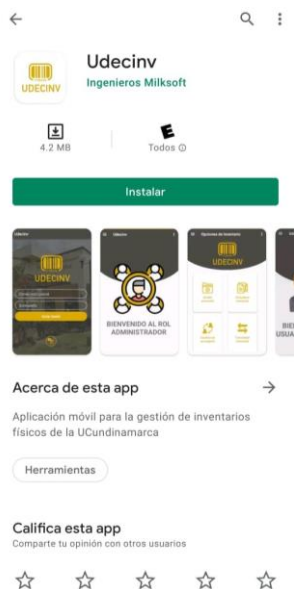
Menú solicitudes realizadas

- 2.1. Botón inicio: este botón permite retornar a la pagina anterior.
- 2.2. Botón solicitudes realizadas recarga la página.
- 2.3. Botón eliminar: permite eliminar la solicitud realizada.

5 Aplicativo móvil.

5.1 Instalar aplicación.

La app de UDECINV se encuentra disponible en las tiendas de aplicación de play store esta para los dispositivos con sistema operativo Android.



Qr para descargar aplicación móvil

5.2 Especificaciones mínimas para aplicativo móvil.

Para descargar la aplicación móvil de la play store se debe de tener una versión de Android mayor a 4.1, almacenamiento disponible de 10mb para su instalación, la resolución de cámara del dispositivo deberá ser mínimo de 5 megapíxeles y deberá contar una conexión a internet con una mínima velocidad simétrica de 2mb para su óptimo funcionamiento.

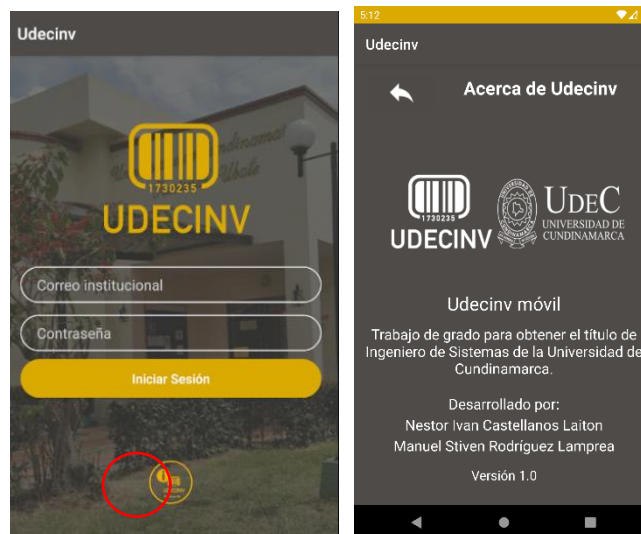
5.3 Iniciar sesión “Login” app



Login

Para acceder a la aplicación es obligatorio identificarse, para ello se deberá ingresar el correo institucional y la contraseña dada por el administrador, en el caso que lo datos sean erróneos se visualizará un mensaje con la informa que es errónea.

5.4 Acerca de

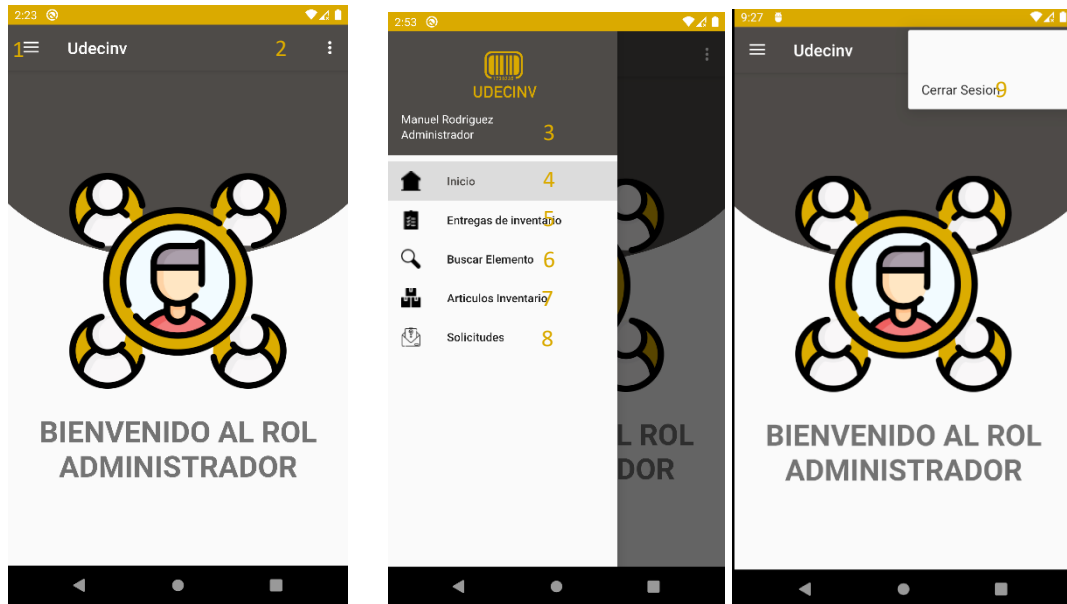


Acerca de.

En la parte inferior del sistema se encuentra un botón donde se puede visualizar la información sobre el sistema multiplataforma.

6 Usuario administrador app.

6.1 Pantalla de inicio usuario administrador



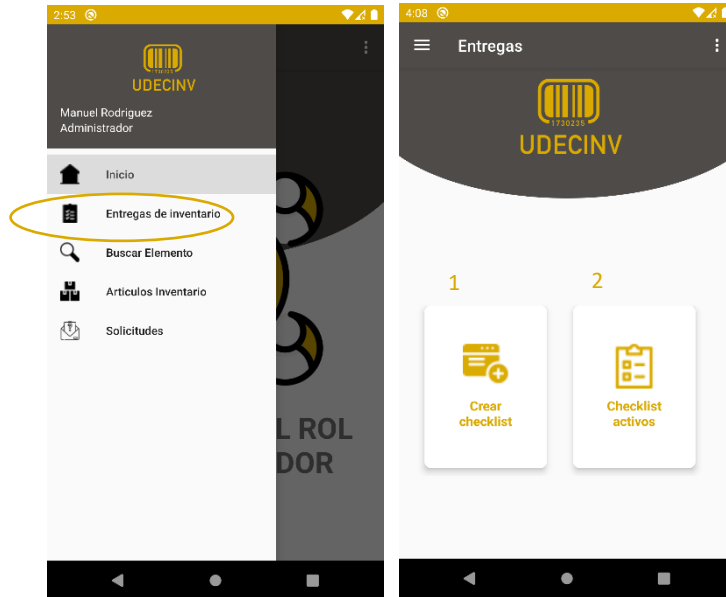
Interfaz de inicio.

Menú desplegable lateral.

Al iniciar la sesión se visualiza la primera interfaz donde cuenta con una serie de opciones que son las siguientes:

11. Menú lateral izquierdo.
12. Menú superior derecho.
13. Identificador de usuario.
14. Botón para pantalla inicio de sesión.
15. Botón para entregas de inventarios.
16. Botón para buscar elemento.
17. Botón para artículos inventario.
18. Botón para solicitudes.
19. Botón para cerrar sesión.

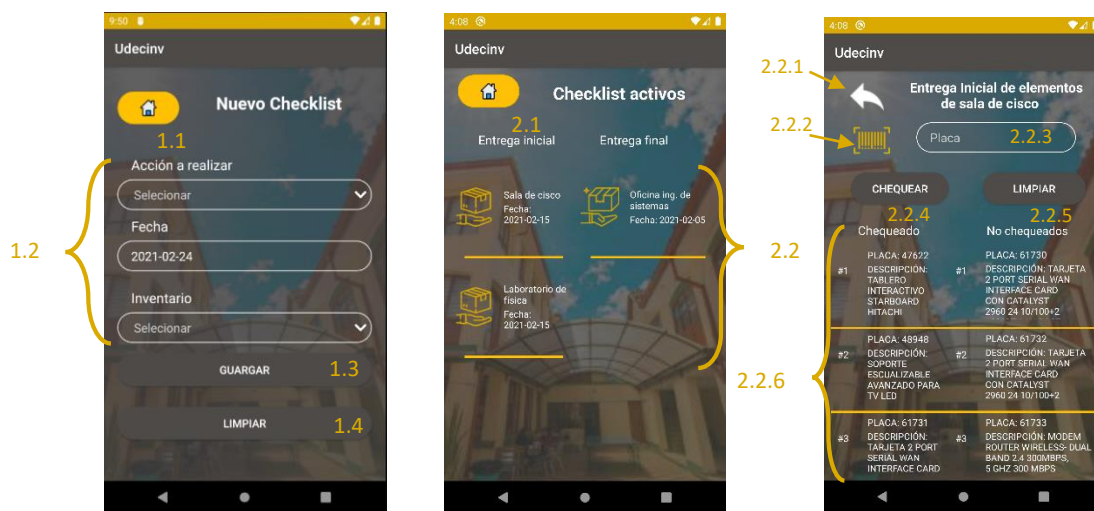
6.2 Acceso a entregas de inventario.



Menú de entregas de inventarios.

7. Botón de crear checklist: este sirve para crear un nuevo checklist para realizar el proceso de chequeo de los elementos de los inventarios.
8. Botón para ver checklist activos: este sirve para visualizar que elementos están activos para realizar el proceso de entregas, de igual manera al dar clic sobre el checklist se abre la pantalla de chequeo.

6.2.1 Chequeo de elementos



Nuevo checklist.

Checklist activos.

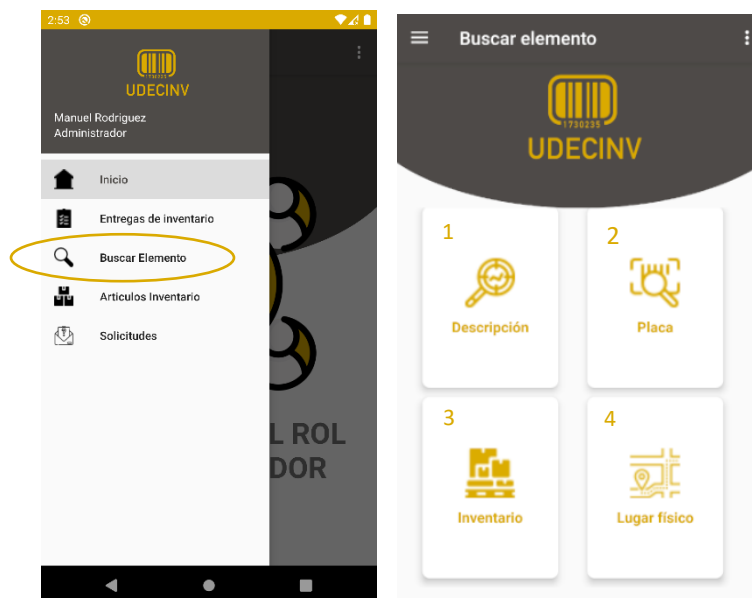
Chequeo.

- 1.3. Botón para volver a la pantalla principal.
- 1.4. Campos para ingresar y seleccionar la información del checklist.
- 1.5. Botón para guardar y crear checklist.
- 1.6. Botón para limpiar todos los campos.
- 2.1. Botón para volver a la pantalla principal.
- 2.2. Lista de checklist activos clasificados en entrega inicial o final.
 - 2.2.1. Botón atrás para volver a la pantalla de checklist activos.
 - 2.2.2. Botón para abrir lector de código de barras. Si es la primera vez que ingresa al sistema saldrá una pestaña para pedir permiso al usuario del teléfono móvil.
 - 2.2.3. Campo de texto para digitar placa de manera manual.
 - 2.2.4. Botón para chequear el elemento de manera manual.
 - 2.2.5. Botón para limpiar el campo de texto de la placa.
 - 2.2.6. Lista de elementos chequeados y no chequeados.

Para el proceso de crear un nuevo checklist se debe de seleccionar la acción a realizar, después la fecha del día del chequeo en seguida seleccionar el inventario y por ultimo guardar el checklist.

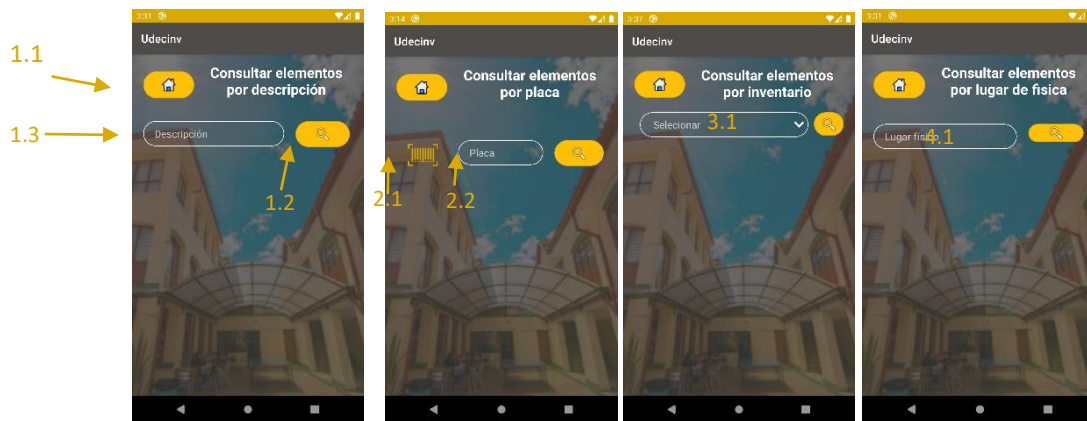
Al realizar el proceso de chequeo se debe seleccionar el checklist activo del inventario, se visualizará la pantalla de chequeo, donde muestra los todos los elementos de ese inventario clasificados en chequeados y no chequeados, para realizar el proceso de chequear un elemento se puede ingresando manual mente el número de placa y dar en chequear o por medio de lector de código de barras.

6.3 Acceso a buscar elementos.



Menú de búsquedas.

- 1. Botón para realizar búsquedas descripción de placa del elemento.
- 2. Botón para realizar búsquedas por número de placa del elemento.
- 3. Botón para realizar búsqueda por inventario.
- 4. Botón para realizar búsquedas por lugar físico del elemento.



Búsqueda por Descripción.

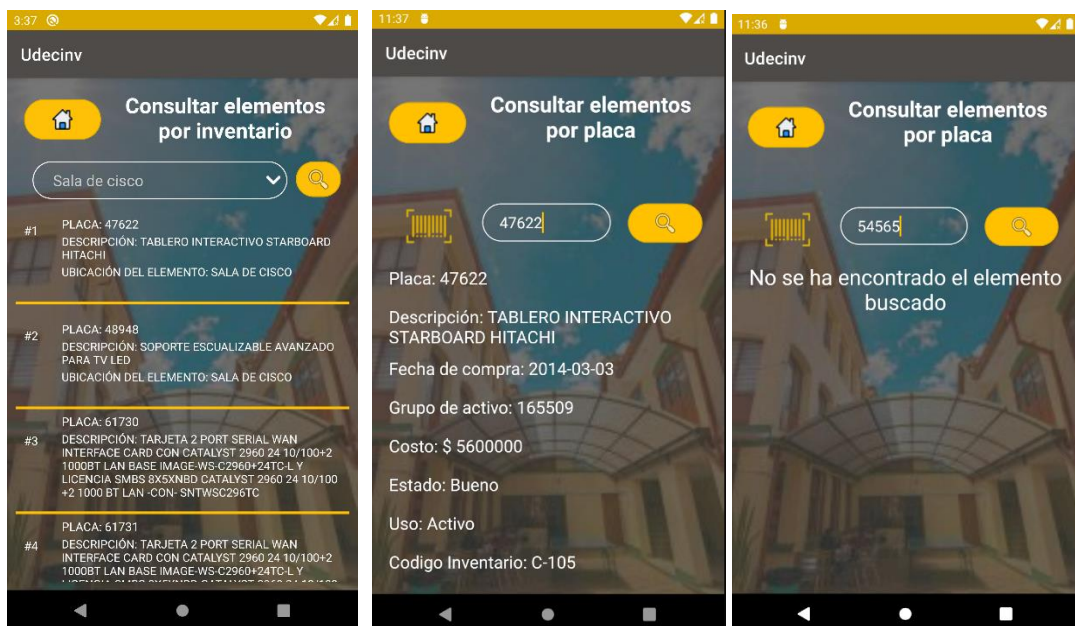
Búsqueda por Placa.

Búsqueda por Inventario.

Búsqueda por lugar físico.

- 1.1. Botón para volver a la pantalla principal.
- 1.2. Botón para realizar la búsqueda.
- 1.3. Campo de texto para escribir descripción.
- 2.1. Botón para abrir lector de código de barra. Si es la primera vez que ingresa al sistema saldrá una pestaña para pedir permiso al usuario del teléfono móvil.
- 2.2. Campo de texto para ingresar el número de placa del elemento a buscar.
- 3.1. Menú desplegable para seleccionar inventario a buscar.
- 4.1. Campo de texto para ingresar la ubicación física de los elementos a buscar.

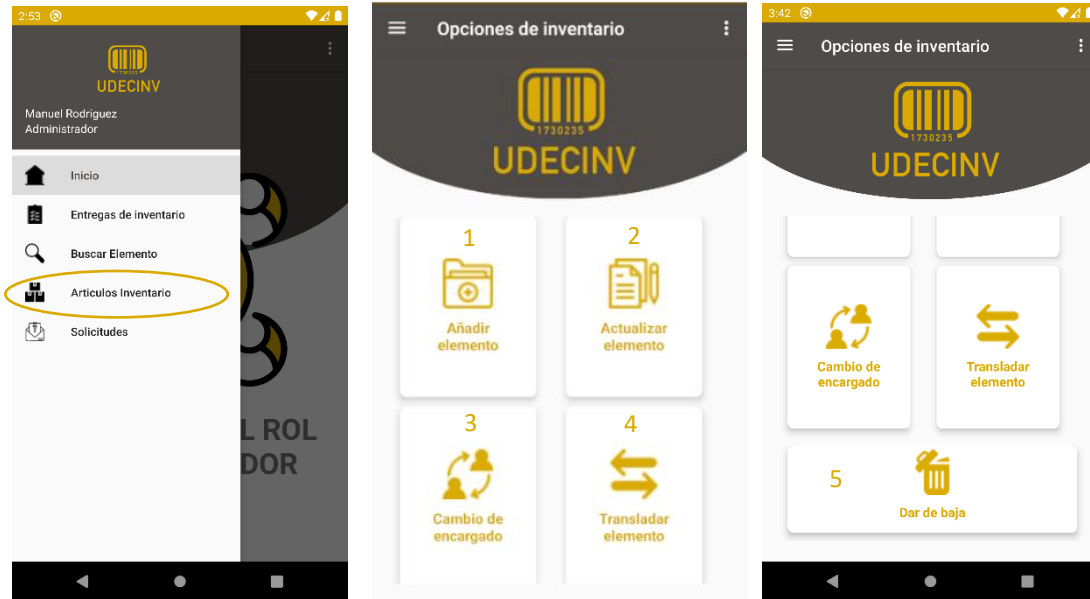
Para el proceso de búsquedas se realiza de la misma manera para todas las opciones, se debe ingresar o seleccionar la información del elemento a buscar y dar en el icono de la lupa, la aplicación empezara el proceso de búsqueda, si fue exitosa se visualizará la información de lo contrario saldrá un mensaje que no se encontró la búsqueda.



Búsquedas exitosas

Búsqueda fallida

6.4 Acceso a artículos inventario.



Menu de articulos de inventario.

1. Botón para ingresar un nuevo elemento al sistema multiplataforma.
2. Botón para actualizar información del elemento.
3. Botón para cambiar el encargado del inventario.
4. Botón para trasladar elemento de un inventario a otro.
5. Botón para dar de baja a un elemento.

6.4.1 Añadir nuevo elemento

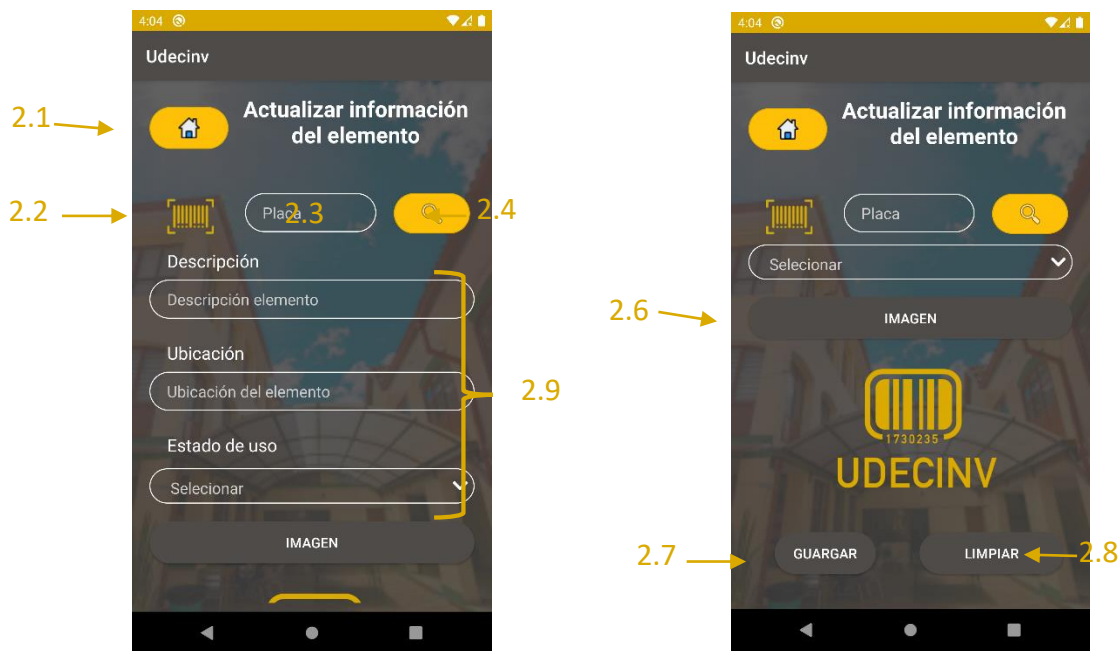


Formulario de añadir elemento.

- 1.1. Botón para volver a la pantalla principal.
- 1.2. Botón para abrir lector de código de barra. Si es la primera vez que ingresa al sistema saldrá una pestaña para pedir permiso al usuario del teléfono móvil.
- 1.3. Campos de texto para ingresar la información del elemento a ingresar.
- 1.4. Botón para abrir la cámara del teléfono móvil y tomar la foto.
- 1.5. Botón para guardar el nuevo elemento.
- 1.6. Botón para limpiar todos los campos.

Para agregar un nuevo elemento al inventario se deben de llenar todos campos de texto con su correspondiente información, en el caso del campo de la placa se puede ingresar manualmente o por medio le lector código de barras, para agregar la foto al formulario se debe dar clic en el botón este abrirá la cámara para que el usuario tomo la foto del elemento “se aconseja tomar la foto de manera horizontal”, después de haber llenado todos los campos de se debe dar en el botón guardar si el elemento ya está creado saldrá una alerta que informe sobre este.

6.4.2 Actualizar elemento.

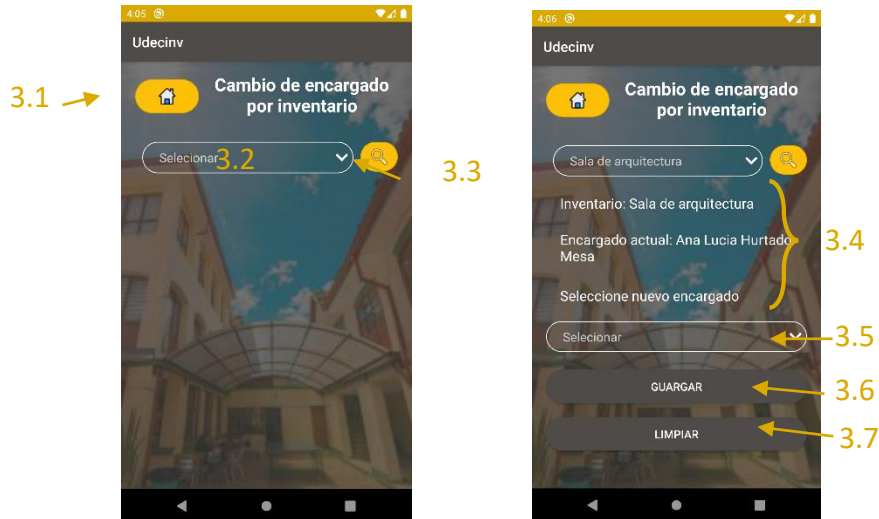


Formulario de actualizar información del elemento.

- 2.1. Botón para volver a la pantalla principal.
- 2.2. Botón para abrir lector de código de barra. Si es la primera vez que ingresa al sistema saldrá una pestaña para pedir permiso al usuario del teléfono móvil.
- 2.3. Campo de texto para ingresar
- 2.4. Botón para buscar información actual del elemento.
- 2.5. Campos de texto sobre información posible a modificar.
- 2.6. Botón para actualizar imagen del elemento
- 2.7. Botón para guardar los cambios realizados.
- 2.8. Botón para limpiar todos los campos.

Para actualizar la información del elemento primero se debe de buscar el elemento por medio de la placa en el campo en seguida se debe dar en el botón de lupa 2.4 para cargar la información actual del elemento, después de actualizar la información del elemento se debe dar en el botón de guardar, si el usuario no ha modificado ningún campo saldrá un mensaje.

6.4.3 Cambio de encargado.

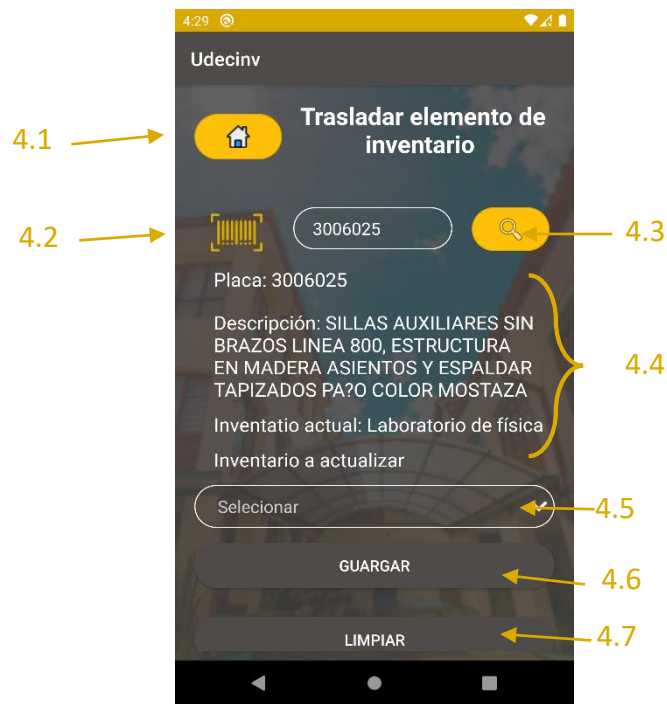


Formulario de cambio de encargado.

- 3.1. Botón para volver a la pantalla principal.
- 3.2. Menú desplegable con lista de inventarios.
- 3.3. Botón para buscar información del inventario.
- 3.4. Información del inventario seleccionado.
- 3.5. Menú desplegable con lista de encargados activos.
- 3.6. Botón para guardar el cambio de encargado.
- 3.7. Botón para limpiar la información del inventario.

Para el cambio de encargado se debe primero seleccionar el inventario a cambiar de encargado, de manera simultánea se consultará la información actual del inventario, siguiente se debe de seleccionar el nuevo encargado del inventario, seguido de esto se debe dar en guardar, si el usuario selecciona el mismo encargado saldrá un mensaje.

6.4.4 Trasladar elemento de inventario.

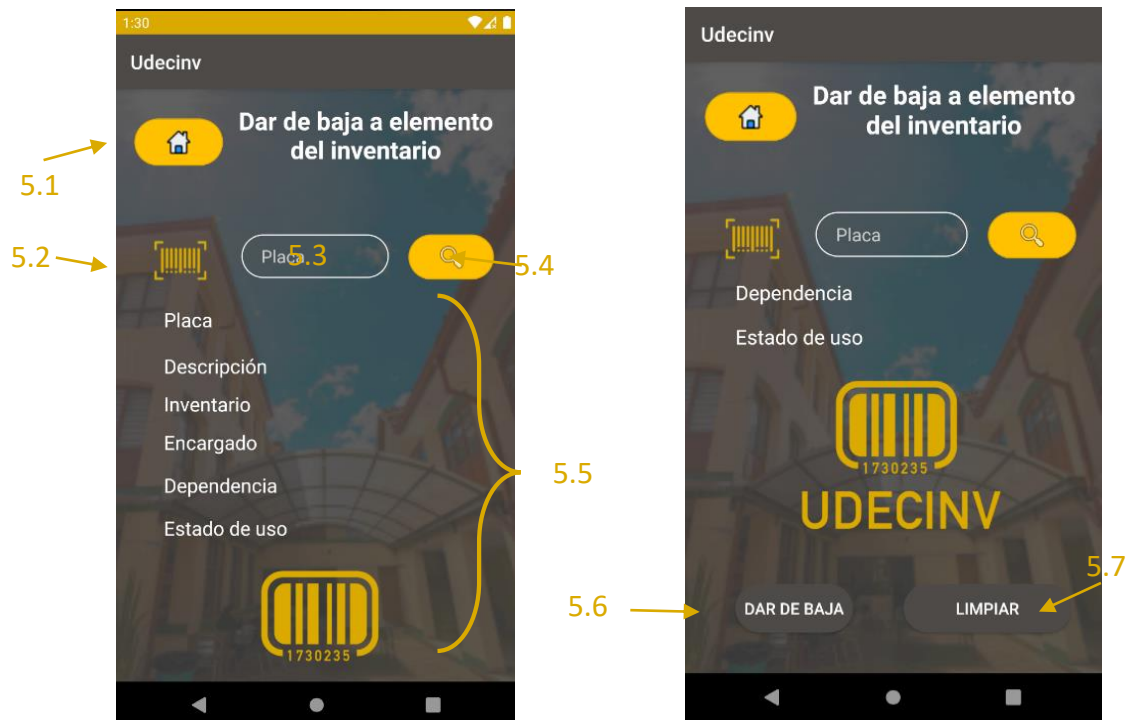


Formulario de traslado de elemento.

- 4.1. Botón para volver a la pantalla principal.
- 4.2. Botón para abrir lector de código de barras.
- 4.3. Botón para buscar información del elemento.
- 4.4. Información del elemento a trasladar.
- 4.5. Menú desplegable con lista de inventarios.
- 4.6. Botón para guardar el traslado de elemento.
- 4.7. Botón para limpiar información del elemento.

Para trasladar un elemento de un inventario a otro se debe primero buscar el elemento para cargar la información y conocer el inventario actual al que pertenece, seguido a esto debe seleccionar el inventario a trasladar y para finalizar se debe guardar el proceso.

6.4.5 Dar de baja.

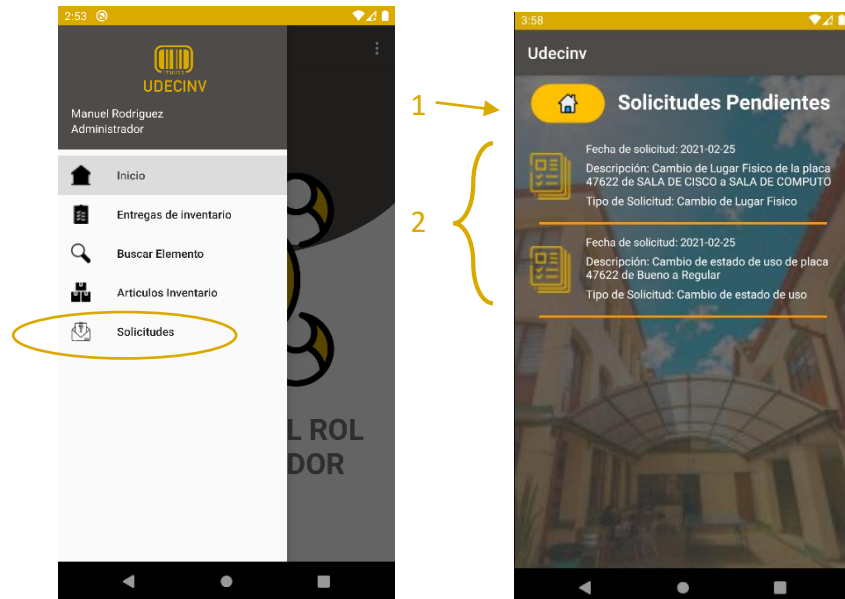


Formulario dar de baja a un elemento.

- 5.1. Botón para volver a la pantalla principal.
- 5.2. Botón para abrir lector de código de barras.
- 5.3. Campo de texto para agregar placa del elemento a dar de baja.
- 5.4. Botón para buscar la información del elemento.
- 5.5. Información del elemento.
- 5.6. Botón para dar de baja.
- 5.7. Botón para limpiar la información del elemento.

Para dar de baja a un elemento se debe ingresar el número de placa y dar en el botón dar de baja saldrá un aviso de confirmar la acción, si el usuario desea estar seguro del elemento a dar de baja puede dar el botón de la lupa para cargar la información del elemento.

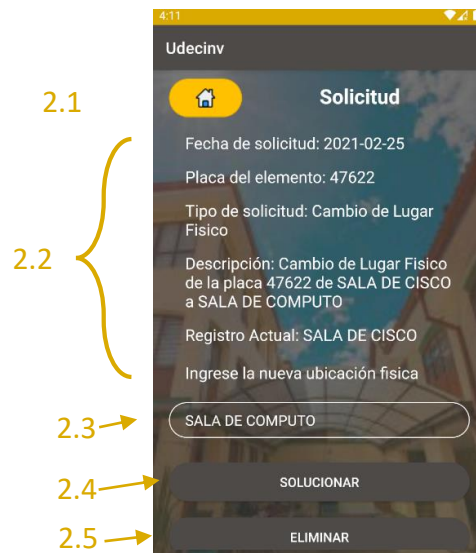
6.5 Acceso a solicitudes.



Opciones solicitudes

1. Botón para volver a la pantalla principal.
2. Lista de solicitudes pendientes.

6.5.1 Responder solicitud



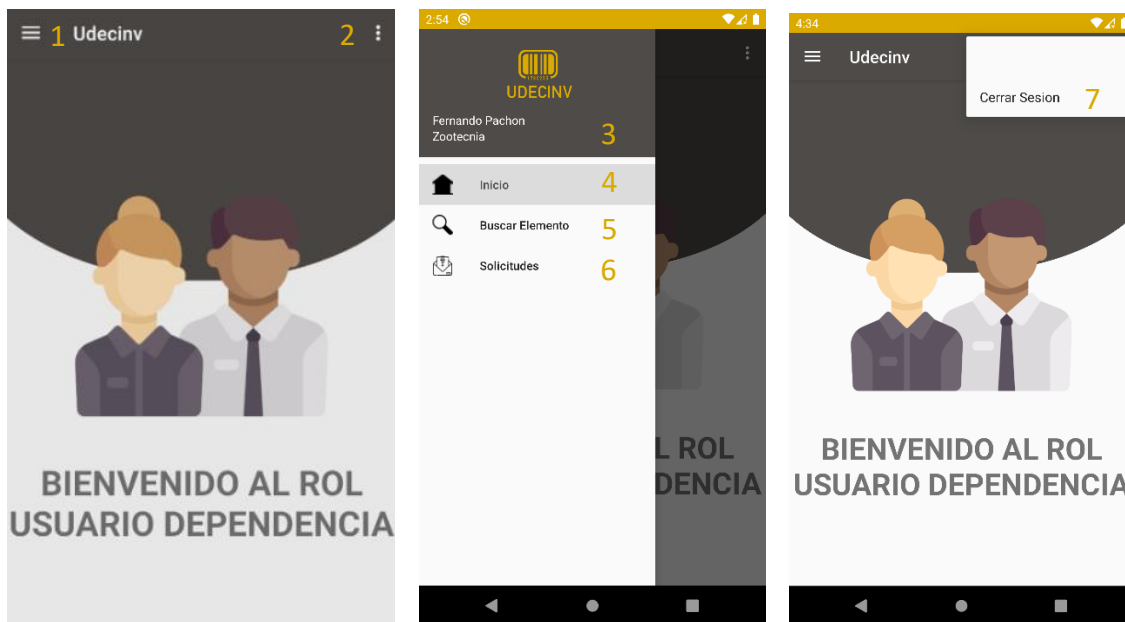
Formulario de acción de solicitud.

- 2.1. Botón para volver a la pantalla principal.
- 2.2. Información sobre la solicitud dada por el usuario.
- 2.3. Dato a actualizar sobre la solicitud.
- 2.4. Botón para solucionar solicitud.
- 2.5. Eliminar solicitud pendiente.

En las opciones de solicitudes se visualiza una lista con las solicitudes pendientes que ha generado el usuario administrador, para dar solución a esta se debe dar clic sobre la solicitud que quiere dar respuesta, allí se visualizara la información de la solicitud y un campo de texto para agregar el cambio a realizar que pidió el usuario.

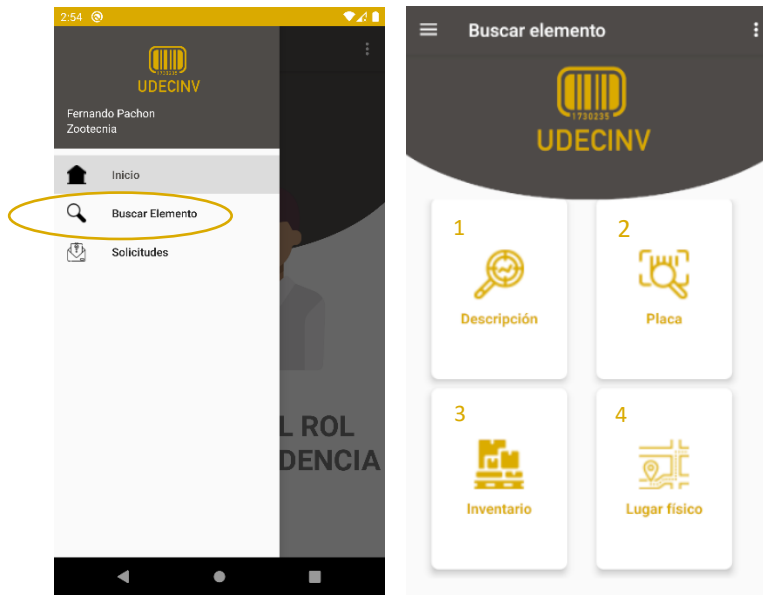
7 Usuario dependencia app.

7.1 Pantalla de inicio usuario dependencia.



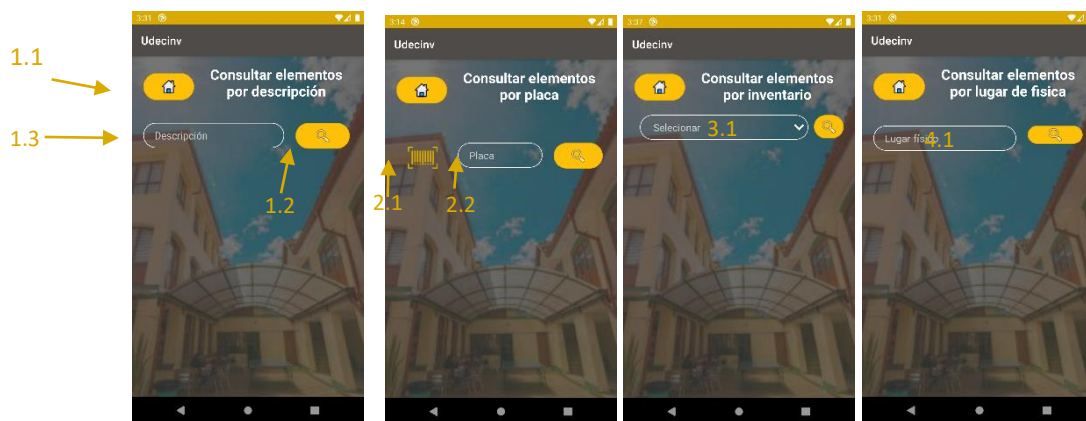
1. Menú lateral izquierdo.
2. Menú superior derecho.
3. Identificador de usuario.
4. Botón para pantalla inicio de sesión.
5. Botón para buscar elemento.
6. Botón para solicitudes.
7. Botón para cerrar sesión.

7.2 Acceso a buscar elementos.



Menú de búsquedas.

1. Botón para realizar búsquedas descripción de placa del elemento.
2. Botón para realizar búsquedas por número de placa del elemento.
3. Botón para realizar búsqueda por inventario.
4. Botón para realizar búsquedas por lugar físico del elemento.



Búsqueda por Descripción.

Búsqueda por Placa.

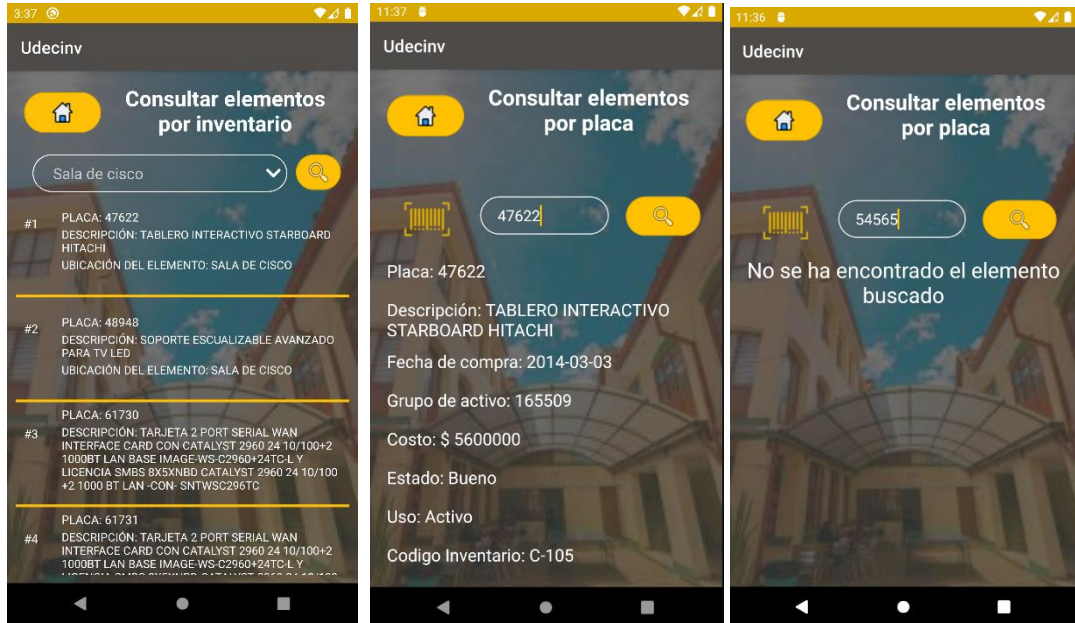
Búsqueda por Inventario.

Búsqueda por lugar físico.

- 1.1. Botón para volver a la pantalla principal.
- 1.2. Botón para realizar la búsqueda.
- 1.3. Campo de texto para escribir descripción.
- 2.1. Botón para abrir lector de código de barra. Si es la primera vez que ingresa al sistema saldrá una pestaña para pedir permiso al usuario del teléfono móvil.
- 2.2. Campo de texto para ingresar el número de placa del elemento a buscar.
- 3.1. Menú desplegable para seleccionar inventario a buscar.

4.1. Campo de texto para ingresar la ubicación física de los elementos a buscar.

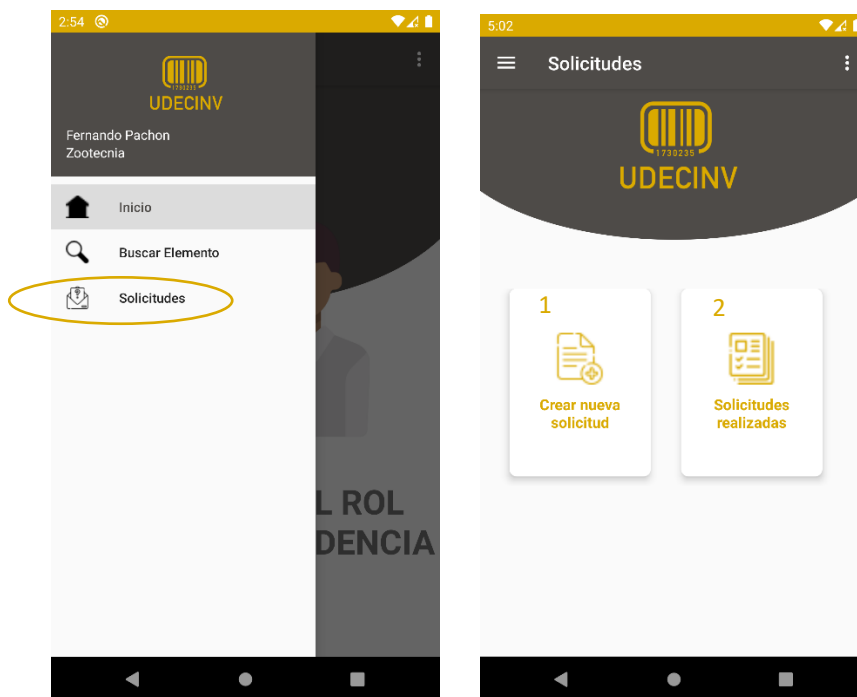
Para el proceso de búsquedas se realiza de la misma manera para todas las opciones, se debe ingresar o seleccionar la información del elemento a buscar y dar en el icono de la lupa, la aplicación empezara el proceso de búsqueda, si fue exitosa se visualizará la información de lo contrario saldrá un mensaje que no se encontró la búsqueda.



Búsquedas exitosas

Búsqueda fallida

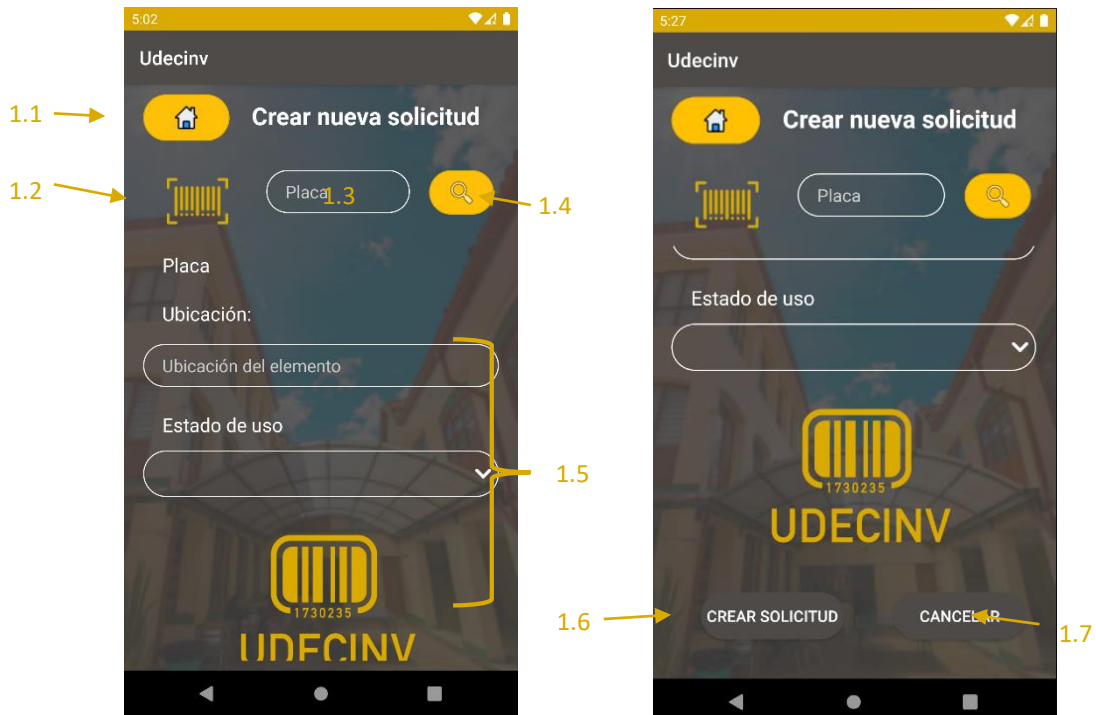
7.3 Acceso a solicitudes.



Menu solicitudes

1. Crear nueva solicitud.
2. Ver solicitudes realizadas.

7.3.1 Crear nuevo solicitud.

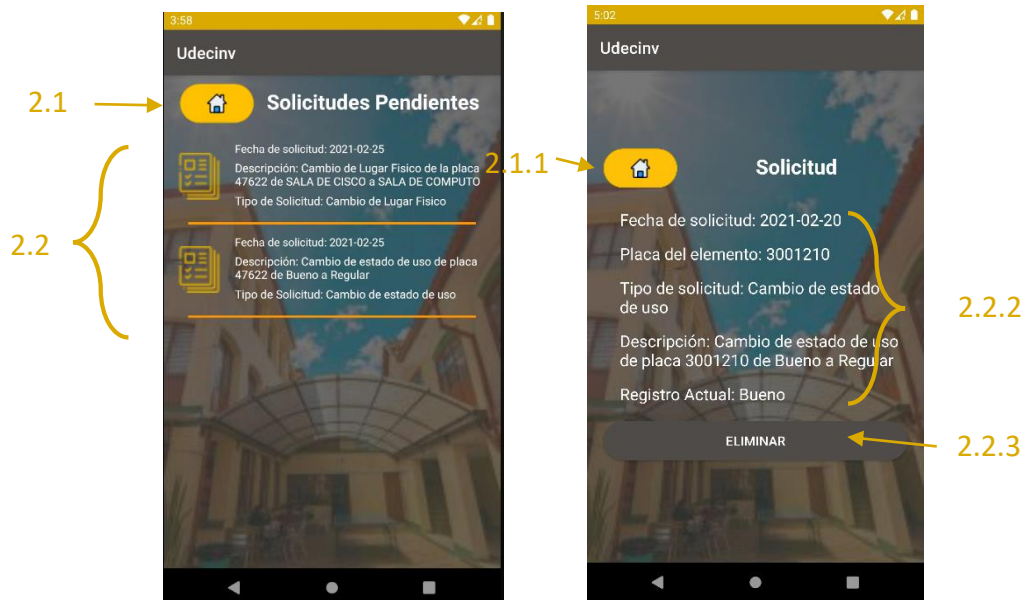


Crear nueva solicitud

- 1.1. Botón para volver a la pantalla principal.
- 1.2. Botón para abrir lector de código de barras.
- 1.3. Campo de texto para ingresar placa a elemento de realizar solicitud
- 1.4. Buscar información del elemento.
- 1.5. Campos de texto de información.
- 1.6. Botón para crear solicitud.
- 1.7. Botón para cancelar solicitud.

Para crear una solicitud se debe de primero ingresar el numero placa del elemento a crear solicitud, siguiente a esto se debe buscar la información actual de este, seguido de esto se debe modificar los campos y por ultimo se debe dar en crear solicitud.

7.4 Solicitudes realizadas.



Solicitudes pendientes a solucionar.

- 2.1. Botón para volver a la pantalla principal
- 2.2. Lista de solicitudes pendientes de solucionar por el administrador.
- 2.2.1. Botón para volver a la pantalla principal.
- 2.2.2. Información sobre la solicitud requerida.
- 2.2.3. Botón para eliminar solicitud.

En solicitudes realizadas se visualizan todas las solicitudes generadas y a espera que el usuario administrador de solución a esta o la elimine por no poder aplicarse.

**MANUAL DE PROGRAMADOR PARA SISTEMA
MULTIPLATAFORMA UDECV**



AUTORES:

NESTOR IVAN CASTELLANOS LAITON

MANUEL STIVEN RODRÍGUEZ LAMPREA

Tabla de contenido

1	Introducción.....	6
2	Objetivo.....	6
3	Herramientas de desarrollo	7
3.1	Node js.	7
3.2	JavaScript	7
3.3	Expressjs.....	7
3.4	Mysql.....	7
3.5	Android studios.....	7
3.6	Php 7.	8
3.7	Rest Api.	8
4	Integración de sistema multiplataforma.	9
5	Funciones principales de Sitio web.....	11
5.1	Generar código de barras	11
5.2	Validación de usuarios	11
5.3	Procesamiento de imágenes.....	12
5.4	Validar si existe imagen	13
5.5	Mostrar imagen en modal	13
5.6	Mostrar imagen almacenada en base 64.....	14
5.7	Generación de reportes en pdf.....	14
5.8	Búsquedas.....	15
5.9	Opciones CRUD	16
5.9.1	Guardar	16
5.9.2	Actualizar	17
5.9.3	Seleccionar	18
5.9.4	Eliminar	18
6	Funciones principales aplicación móvil.....	19
6.1	Validación de usuarios.	19
6.2	Lector de código barras.....	20
6.3	Capturar foto con el teléfono móvil.....	22
6.4	Procesamiento de imágenes.....	23
6.5	Opciones CRUD	25

6.5.1	Añadir elemento	26
6.5.2	Actualizar elementos	28
6.5.3	Consultar elementos	29
6.5.4	Eliminar solicitud.....	30
6.6	Cambio de encargado de inventario.....	30
6.7	Notificaciones.	31

Tabla de figuras

Figura 1 <i>Conexión a base de datos sitio web</i>	9
Figura 2 <i>Conexión a base de datos Aplicativo móvil</i>	10
Figura 3 <i>Generar código de barras</i>	11
Figura 4 <i>Validar usuarios sitio web</i>	12
Figura 5 <i>Validar imágenes sitio web</i>	13
Figura 6 <i>Mostrar imagen en modal</i>	13
Figura 7 <i>Mostrar imagen en base64</i>	14
Figura 8 <i>Generar pdf parte1</i>	14
Figura 9 <i>Generar pdf parte2</i>	15
Figura 10 <i>Búsquedas sitio web</i>	16
Figura 11 <i>Guardar en sitio web</i>	17
Figura 12 <i>Actualizar en sitio web</i>	17
Figura 13 <i>Seleccionar en sitio web</i>	18
Figura 14 <i>Eliminar en sitio web</i>	18
Figura 15 <i>Consulta de usuario y validación de contraseña</i>	19
Figura 16 <i>Direccionamiento de pantalla según el rol</i>	20
Figura 17 <i>Código funcionamiento lector de código de barras</i>	21
Figura 18 <i>Permisos de cámara y vibrador del dispositivo</i>	22
Figura 19 <i>Permisos para abrir la cámara del dispositivo</i>	22
Figura 20 <i>Código para tomar la foto desde la aplicación</i>	22
Figura 21 <i>Procesamiento de la foto</i>	23
Figura 22 <i>Código para redimensionar foto</i>	23
Figura 23 <i>Código para convertir la imagen en base 64 Android</i>	24
Figura 24 <i>Código para convertir de base64 a imagen en php</i>	24
Figura 25 <i>Código para convertir la imagen en base 64</i>	24
Figura 26 <i>Código para convertir de base64 a imagen Android</i>	25
Figura 27 <i>Librería para acceso a internet</i>	25
Figura 28 <i>Envío de información por medio de la Rest api</i>	26
Figura 29 <i>Añadir elemento aplicativo móvil</i>	26
Figura 30 <i>Almacenar la información por method post</i>	27
Figura 31 <i>Ingresar datos a la base de datos</i>	27
Figura 32 <i>Organizar información de búsqueda</i>	28

Figura 33 <i>Mostrar información en pantalla de la búsqueda.</i>	28
Figura 34 <i>Modificar datos en la base de datos.</i>	29
Figura 35 <i>Búsquedas por el method Get.</i>	29
Figura 36 <i>Consulta de la búsqueda a la base de datos.</i>	29
Figura 37 <i>Clasificar información de la búsqueda.</i>	30
Figura 38 <i>Mostrar recycler view en la pantalla.</i>	30
Figura 39 <i>Eliminar solicitud en base de datos.</i>	30
Figura 40 <i>Cambio de encargado consulta mysql.</i>	31
Figura 41 <i>Librerías para generar notificaciones.</i>	31
Figura 42 <i>Código para visualizar notificaciones.</i>	32

1 Introducción

La temática que se abordará en este apartado está enfocada hacia la comprensión del desarrollo del código de la Aplicación móvil y sistema de información como plataformas de desarrollo para la gestión de inventarios de la universidad de Cundinamarca seccional Ubaté (Udecinv), caso de estudio programa ingeniería de sistemas el cual es desarrollado con el propósito de servir como un manual técnico o guía para los desarrolladores.

2 Objetivo

Mostrar la estructura lógica y de funcionamiento del sistema multiplataforma Udecinv con el fin de dar a comprender el proceso de integración de las diferentes plataformas utilizadas en el sistema.

3 Herramientas de desarrollo

3.1 Node js.

nodejs, entorno de ejecución para JavaScript del lado del servidor construido en el motor de JavaScript v8 de Chrome, es una de las principales herramientas de desarrollo para sistema multiplataforma, ya que permite la manipulación de la información de las bases de datos con las vistas en el sitio web, además de ser el gestor de toda la estructura lógica del sistema en el sitio web desde el backend.

3.2 JavaScript

JavaScript, lenguaje de programación interpretado que se implementa normalmente para la visualización en navegadores web del lado del cliente, pero que en la actualidad también puede darse del lado del servidor como en el entorno de programación nodejs, en el sistema multiplataforma este lenguaje se encuentra del lado del cliente y del servidor en diferentes funcionalidades gracias al uso de nodejs como lenguaje de programación para el backend.

3.3 Expressjs.

Express js, es un marco de aplicación web de back-End para nodejs de código abierto que permite facilitar la interacción de las diferentes clases utilizadas durante el desarrollo del sistema multiplataforma.

3.4 Mysql.

MySQL, Gestor de base de datos relacional, multihilo y multiusuario de código abierto, siendo uno de los más usados en el mundo, se usó para el almacenamiento de los datos que se utilizan para la gestión de inventarios en la Universidad de Cundinamarca seccional Ubaté.

3.5 Android studios.

Android Studio, entorno de desarrollo oficial integrado oficial para plataformas de Android, se usó como entorno de codificación para todo el aspecto lógico y funcional y así mismo sus interfaces gráficas de la aplicación para teléfonos móviles, de igual manera el uso de los emuladores de Android con distintas Api para verificar su compatibilidad.

3.6 Php 7.

Php 7, es un lenguaje de programación adaptado especialmente para el desarrollo web, se usó como intermediario entre el servidor y la base de datos realizando consultas y emitiendo una respuesta que será procesada por la aplicación móvil.

3.7 Rest Api.

Rest Api, Conjunto de reglas específicas que son utilizadas para comunicarse entre otras aplicaciones, es un mecanismo útil para conectar dos softwares entre sí, de esta manera garantizar el intercambio de información, se utilizó como puente de comunicación entre la aplicación móvil y el servidor donde se encuentra alojado las consultas a la base de datos por los métodos Get y Post.

4 Integración de sistema multiplataforma.

Udecinv cuenta con plataformas que es sitio web y aplicación móvil para Android, estas cuentan con una característica principal que es la sincronización en tiempo real permitiendo que las dos plataformas tengan los datos actualizados, de esta manera lo que se realiza en alguna de las dos plataformas se verá reflejado en la otra.

La conexión del sistema multiplataforma con la base de datos se realiza mediante el nombre del servidor a conectar, un nombre de usuario, una contraseña si es el caso y el nombre de la base de datos a la que se quiere conectar, en este caso esos datos se ven reflejados en la imagen después de este apartado. Donde mediante nodejs se hace un archivo llamado conexión y este contiene el siguiente código:

Figura 1

Conexión a base de datos sitio web

```

module.exports = {
  database: {
    //servidor
    host: 'host25.latinoamericahosting.com',
    //usuario
    user: 'molinodo_wp1409',
    //contraseña base de datos
    password: 'zqE~u-UaX9P',
    // nombre de base de datos
    database: 'molinodo_udecinvok'
  }
};

```

La conexión a la base de datos para la aplicación móvil se debe de hacer uso del funcionamiento de la Rest Api esta se encarga de hacer la conexión entre la aplicación móvil y el servidor, en el servidor se encuentra un archivo con extensión .php que es el encargo de hacer la conexión con la base de datos.

Figura 2

Conexión a base de datos Aplicativo móvil

```

1  <?php
2
3  //Servidor
4  $hostname='host25.latinoamericahosting.com';
5  //nombre de base de datos
6  $database='molinodo_udecinok';
7  //usuario
8  $username='molinodo_wp1409';
9  //contraseña
10 $password='zqE~u-UaX9P';
11
12
13 $conexion=mysqli_connect($hostname,$username,$password);
14
15 //Verificación de la conexión a la base de datos
16 if (!$conexion) {
17     echo "no se pudo conectar con la base de datos";
18 } else {
19     $base = mysqli_select_db($conexion, $database);
20     if (!$base) {
21         echo "no se conecto a ninguna base de datos";
22     } else {
23
24     }
25 }
26 ?>

```

5 Funciones principales de Sitio web

5.1 Generar código de barras

El sistema multiplataforma en el sitio web permite la generación de código de barras, este proceso es realizado gracias a la implementación de JavaScript mediante un código que permite recibir un parámetro, donde es recibido por una función que utiliza la librería JsBarcode la cual hace que se genere un código de barras, el cual se mostrará en un modal en forma de imagen donde se podrá descargar para su posterior impresión.

Figura 3 Generar código de barras

```
function generarcodigobarras(placa) {
    d=placa.split('||');
    var x = d[0];

    var titulo=d[1];
    document.getElementById('nombre').innerHTML=titulo;
    $('#cod').val(d[0]);

    JsBarcode("#codigobarras",x);
}
```

5.2 Validación de usuarios

En el sitio web para validar el usuario según el rol asignado se realiza por medio de una función llamada local-signin que permite consultar la información del usuario en la base de datos y comparar las contraseñas almacenadas en esta, mediante un proceso donde se descripta la contraseña almacenada en la base de datos y se compara con la registrada en el formulario, si el usuario o la contraseña no existen se mostrará una alerta informando lo anterior y si las contraseñas son diferentes de igual manera mostrará un mensaje informando que la contraseña es incorrecta. Finalmente, si los datos ingresados en el usuario son iguales a los registrados en la base de datos se dará acceso a el usuario a su respectivo rol.

Figura 4

Validar usuarios sitio web.

```
const passport = require('passport');
const LocalStrategy = require('passport-local').Strategy;

const helpers = require('../funciones/contra');
const pool = require('../basedatos');

passport.serializeUser((user, done) => {
  done(null, user.Id_Sesion);
});

passport.deserializeUser(async (id, done) => {
  const rows = await pool.query('SELECT * FROM sesiones WHERE Id_Sesion = ?', [id]);
  done(null, rows[0]);
});

passport.use('local-signin', new LocalStrategy({
  usernameField: 'email',
  passwordField: 'contra',
  passReqToCallback: true
}, async (req, email, contra, done) => {

  const rows = await pool.query('SELECT * FROM sesiones WHERE Email = ?', [email]);
  if (rows.length > 0) {
    const user = rows[0];
    //console.log(user)
    const validPassword = await helpers.matchPassword(contra, user.Password)
    if (validPassword) {
      done(null, user, req.flash('success', 'bienvenido ' + user.Usuario));
    } else {
      done(null, false, req.flash('message2', 'contraseña incorrecta'));
    }
  } else {
    return done(null, false, req.flash('message', 'el usuario no existe.'));
  }
}
));
```

5.3 Procesamiento de imágenes

Para mejorar la interacción del usuario con el sistema multiplataforma en el sitio web es necesario realizar varias opciones con la renderización de las imágenes, en primera instancia se debe validar si existe imagen, mostrar imágenes en un modal y mostrar imágenes encriptadas en base 64.

5.4 Validar si existe imagen

En este caso se debe validar si existe la imagen almacenada en la base de datos de lo contrario se mostrará el logo implementado para el sistema multiplataforma para mostrar las fotos de cada uno de los elementos almacenados en los diferentes inventarios, esto se realiza gracias a una función de JavaScript que verifica la información de la foto de cada elemento y si esta es vacía o tiene error mostrará por defecto el logo Udecinv.

Figura 5

Validar imágenes sitio web

```

var images = $(".image");

$(images).on("error", function(event) {
    $(event.target).attr("src", "img/logo.png");
});

```

5.5 Mostrar imagen en modal

Cuando se muestra la foto de cada elemento esta se mostrará en miniatura y es necesario dar clic sobre ella para que se pueda ver en una forma más grande, para ello es necesario mostrar la imagen en modal donde no se altere la ventana actual, con una función de JavaScript donde se carga el contenido de la imagen y se redimensiona a una versión más grande como se puede reflejar en la siguiente imagen.

Figura 6

Mostrar imagen en modal

```

var modal = document.getElementById('myModal');

var modalImg = document.getElementById("img01");

// buscamos todas las imagenes
var imgs = document.getElementsByTagName('img');
for(var i = 0; i < imgs.length;i++){
    // por cada imagen le asignamos el evento clic que mostrara la imagen en la modal
    var img = imgs[i];
    img.onclick = function(){
        modal.style.display = "block";
        modalImg.src = this.src;
    }

    // al hacer clic afuera, se cierra el modal
    window.onclick = function(event) {
        if (event.target == modal) {
            modal.style.display = "none";
        }
    }
}
}

```

5.6 Mostrar imagen almacenada en base 64

Para el almacenamiento de las fotos en la base de datos es necesario encriptar la información binaria de esta a base64 con el objetivo que se almacene en un formato blob, para mostrar estas fotos en el sitio web se requiere especificar en qué formato se encuentra la imagen y toda la información binaria de esta almacenada en un vector.

Figura 7

Mostrar imagen en base64

```
<div class="body">
  <a data-toggle="modal" data-target="#myModal" onclick="updateFotoElemento('<%=elemento%>')">
  </a>
</div>
```

5.7 Generación de reportes en pdf

Para el proceso de checklist, es necesario generar un reporte en formato pdf donde se pueda verificar la información de los elementos chequeados y no chequeados, para ello se utilizó la librería pdf node-creator la cual permite renderizar una plantilla elaborada con html con la información a mostrar en el pdf que finalmente es almacenada en la tablas checklist de la base de datos en la columna reportes, en este proceso también se cambia el estado del checklist de activo a finalizado para no alterar la información suministrada.

Figura 8

Generar pdf parte 1

```
var na = nombre2+' '+app;
//se guarda la ruta de la plantilla html en una variable
var html = fs.readFileSync('./src/vistas/template.html', 'utf8');

//le doy formato y opciones a el archivo pdf
var options = {
  format: "A3",
  orientation: "landscape",
  border: "15mm"
};

/*
envio los parametros obtenidos en mysql y creo la ruta para guardar el archivo
temporalmente en el servidor.
*/
var document = {
  html: html,
  data: {
    arr: arr, arr2:arr2, fecha:fecha, fecha2:fecha2, nombre:nombre, encargado:na, accion:accion, idven:idven
  },
  path: "./src/vistas/checklist.pdf"
};
```

Figura 9

Generar pdf parte2

```

//creo el archivo pdf y lo guardo en mysql.
pdf.create(document, options)
  .then(res => {

var pd = fs.readFileSync("./src/vistas/checklist.pdf");

pool.query('UPDATE checklist set ? where Id_Checklist = ?', [{
  Estado:'Finalizado',
  Reporte: pd
}], idcheck],
(err, rows) => {

});

  })
  .catch(error => {
    console.error(error)
  });

/*
mientras se guarda el archivo en mysql, se redirecciona la pagina para que
se continúe trabajando en otros procesos.
*/
pool.query('UPDATE checklist set ? where Id_Checklist = ?', [{
  Estado:'Finalizado'
}], idcheck],
(err, rows) => {
  res.redirect('/checklist');
});
});
});
});
});
});
});
});
});

```

5.8 Búsquedas

Las búsquedas es una de las funciones más importantes en el sistema multiplataforma en el sitio web, donde se filtra la información de los elementos por diferentes parámetros, para este proceso se implementó una fracción de código de JavaScript con Ajax que realiza el proceso de recopilar la información de una tabla y mostrar los atributos obtenidos de un campo de texto o buscador que se especifican en este. reduciendo la información que no es solicitada en la tabla.

Figura 10

Búsquedas sitio web

```

<script>
$(document).ready(function(){
  $("#buscar").on("keyup", function() {
    var value = $(this).val().toLowerCase();
    $("#tabla tr").filter(function() {
      $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
    });
  });
});
</script>

```

5.9 Opciones CRUD

Uno de los procesos más importantes, relevantes y que más se implementan en la ejecución del proyecto de grado son las opciones CRUD donde para todas las funciones se requiere de una de ellas, a continuación, se explica brevemente cada una de ellas y como es realizada con nodejs.

5.9.1 Guardar

La primera opción es guardar, donde se implementa la estructura de SQL para este proceso, recibiendo parámetros del formulario mediante el método POST y estos parámetros se almacenan en variables que luego son almacenadas en la base de datos.

Figura 11

Guardar en sitio web

```
//guardar inventarios
controller.saveInventario = (req, res) => {

const {codigoI, nombreI, descripcionI, dependencia}= req.body;

const rows = pool.query('SELECT * FROM inventarios where Id_Inventario = ?' , [codigoI], (err, rows) => {
if (rows.length > 0) {

req.flash('message2', 'ya existe este Inventario registrado');
res.redirect('/inventarios');
}else{

const insUser = pool.query('INSERT INTO inventarios set ?',{
Id_Inventario: codigoI,
Nombre: nombreI,
Descripcion: descripcionI,
Dependencia: dependencia

} ,(err, datos) => {

req.flash('success', 'Inventario registrado correctamente. ');
res.redirect('/inventarios');

})
}
});
};
```

5.9.2 Actualizar

En la opción de actualizar se requieren datos de los formularios con el método POST y se especifica qué campos se quieren actualizar y finalmente con la consulta sql se realiza este proceso.

Figura 12

Actualizar en sitio web

```
//actualizar inventarios
controller.updateinventarios = (req, res) => {
const {codigoI, nombreI, descripcionI} = req.body;
pool.query('UPDATE inventarios set ? where Id_Inventario = ?', [{
Nombre: nombreI,
Descripcion: descripcionI
},codigoI],
(err, rows) => {
req.flash('success', 'Se actualizó el inventario '+nombreI+' correctamente. ');
res.redirect('/inventarios');
});
};
```

5.9.3 Seleccionar

Esta opción es una de las más simples de las opciones CRUD donde se especifica qué campos se quieren mostrar, luego se realiza la consulta SQL y se almacena la información obtenida en vector que luego será mostrado en los campos de las vistas a renderizar.

Figura 13

Seleccionar en sitio web

```
//lista encargados
controller.listaencargados = (req, res) => {
pool.query('SELECT * FROM encargados', (err, admin) => {

  if (err) {
    res.json(err);
  }
res.render('administrador/encargados', {
  data: admin
});
});
};
```

5.9.4 Eliminar

Finalmente, esta la opción de eliminar, donde simplemente se selecciona el campo o dato a eliminar en la base de datos especificado desde la vista de html, luego se recibe como parámetro en la función de nodejs y finalmente se realiza la consulta SQL para eliminar este dato en la base de datos.

Figura 14

Eliminar en sitio web

```
controller.eliminarSoli = (req, res) => {
  const id = req.params.id;

  pool.query('DELETE FROM solicitudes WHERE Id_Solicitud = ?', [id], (err, rows) =>{

    if (err) {
      res.json(err);
    }

    req.flash('success', 'Se ha eliminado la solicitud correctamente. ');
    res.redirect('/solicitudes');

  });
};
```

6 Funciones principales aplicación móvil.

6.1 Validación de usuarios.

En la aplicación móvil para la validación de usuario según su rol asignado se realiza por medio de la integración de la Rest api para consultar la información en la base de datos y verificar que la contraseña ingresada que sea la correcta allí se realiza un proceso de verificación de la contraseña, ya que esta se encuentra encriptada con la función de Hash, en Android studios se habilita la pantalla principal de usuario que ha ingresado al sistema.

Figura 15

Consulta de usuario y validación de contraseña.

```

<?PHP
include "Conexion.php";
mysqli_set_charset($conexion, "utf8");
    if(isset($_GET["Email"]) && isset($_GET["Password"])){
        $Email=$_GET['Email'];
        $contra=$_GET['Password'];
        $json=array();

        // consulta si el email ingresado es existente
        $sql = mysqli_query($conexion,"SELECT * FROM sesiones where Email = '$Email' ");
        if($vec=mysqli_fetch_array($sql)){
            //verificar si la contraseña es la correcta
            if(password_verify($contra,$vec['Password'])){
                // Cargar informacion en array de la consulta
                $result["Nombres"]=$vec['Nombres'];
                $result["Apellidos"]=$vec['Apellidos'];
                $result["Rol"]=$vec['Rol'];
                $result["Email"]=$vec['Email'];
                $result["Dependencia"]=$vec['Dependencia'];
                $result["Id"]="Si";
                $json['sesiones'][]=$result;
            }
        }else {
            $result["Rol"]="No registrado";
            $result["Id"]="No";
            $json ['sesiones'][]=$result;
        }
    }

```

Consulta de email y validación de contraseña en php.

Por medio de la función password_verify se realiza la validación de que la contraseña ingresada sea igual a la de la base de datos.

Figura 16

Direccionamiento de pantalla según el rol

```

if (rol.equals("Administrador")) {
    Email.setText("");
    Password.setText("");
    startActivity(new Intent(getApplicationContext(), Menu_lateral.class));
    finish();
} else if (rol.equals("Usuario Dependencia")) {
    startActivity(new Intent(getApplicationContext(), Menu_lateral2.class));
    Email.setText("");
    Password.setText("");
    finish();
} else if (rol.equals("No registrado")) {
    Toast.makeText(context Login.this, text: "Email no registrado", Toast.LENGTH_SHORT).show();
    Email.setText("");
    Password.setText("");
} else if (rol.equals("No")) {
    Toast.makeText(context Login.this, text: "Contraseña erronea", Toast.LENGTH_SHORT).show();
    Password.setText("");
}

```

Direccionamiento según su rol de usuario.

6.2 Lector de código barras

Dentro la aplicación móvil se implementó el uso de un lector de código barras haciendo uso de la cámara de teléfono móvil, esto con el fin de realizar el proceso de chequeo de una manera más ágil.

Figura 17

Código funcionamiento lector de código de barras

```
// lector del código de barras
public void escanear() {
    IntentIntegrator intent = new IntentIntegrator( activity: this);
    intent.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
    intent.setPrompt("ESCANEAR ELEMENTO");
    intent.setCameraId(0);
    intent.setOrientationLocked(false);
    intent.setBeepEnabled(false);
    intent.setCaptureActivity(CaptureActivityPortrait.class);
    intent.setBarcodeImageEnabled(false);
    intent.initiateScan();
}
```

La clase escanear es la encargada de realizar cargar los primeros parámetros de lector de código barras como: Activar la cámara, Pedir permiso de la cámara, Orientación de la cámara y detectar el código de barras.

```
// Respuesta del lector de código de barra
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    // escaner
    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    Vibrator vibrator = (Vibrator) getApplicationContext().getSystemService(Context.VIBRATOR_SERVICE);
    if(result != null) {
        if(result.getContents() == null) {
            Toast.makeText( context: Bcodigo.this, text: "Cancelaste el escaneo", Toast.LENGTH_SHORT).show();
        } else {
            Placa.setText(result.getContents().toString());
            consultarpla();
            vibrator.vibrate( milliseconds: 150);
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
```

Dentro del método “onActivityResult” este es el encargado de recolectar el resultado que se obtiene de la clase escanear, cuando se detecta un código el celular realiza una pequeña vibración en seguida el resultado del escaneo se envía al campo placa de la interfaz y por último se ejecuta una clase que varía según en el módulo en que se encuentre, en este caso realiza ejecuta la clase de consultar la placa de elemento escaneado.

Figura 18

Permisos de cámara y vibrador del dispositivo.

```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.CAMERA" />
```

Para el uso de la cámara y la función de vibrar del teléfono móvil se debe agregar estos dos permisos en el archivo "AndroidManifest.xml"

6.3 Capturar foto con el teléfono móvil.

En la aplicación móvil se sigue utilizando la cámara de teléfono móvil, en este caso para la opción de ingresar y actualizar un elemento para dar uso a esta función se debe contar con los permisos del teléfono móvil para hacer uso de cámara y tomar la foto.

Figura 19

Permisos para abrir la cámara del dispositivo.

```
private void fotopermisos(){
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        if (ActivityCompat.checkSelfPermission( context: Actualizar.this, Manifest.permission.CAMERA) == PackageManager.PERMISSION_GRANTED) {
            tomarfoto();
        }else {
            ActivityCompat.requestPermissions( activity: Actualizar.this, new String[]{Manifest.permission.CAMERA}, REQUEST_PERMISOS_CAMARA);
        }
    } else {
        tomarfoto();
    }
}
```

Se deben solicitar los permisos al usuario de teléfono móvil cuando la versión de Android se mayor a 6.0 de lo contrario el aplicativo abrirá de inmediato la cámara.

Figura 20

Código para tomar la foto desde la aplicación

```
private void tomarfoto (){
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if ( intent.resolveActivity(getPackageManager())!=null){
        startActivityForResult(intent,Cod_foto);
    }
}
```

Por medio de la clase “tomarfoto” esta se encarga de abrir la cámara del dispositivo, al ser una aplicación nativa cuenta con el beneficio de contar con todas las funcionalidades de la cámara de teléfono móvil.

Figura 21

Procesamiento de la foto.

```

if (resultCode == Activity.RESULT_OK && data != null) {
    bitmap=(Bitmap)data.getExtras().get("data");
    foto.setImageBitmap(bitmap);
    // Toast.makeText(this,"Tomar la foto de manera horizontal ",Toast.LENGTH_SHORT).show();
    bitmap = redimensionarImagen(bitmap, anchoNuevo: 1280, altoNuevo: 960);
}else {
    Toast.makeText( context: this, text: "No se ha tomado ninguna foto ",Toast.LENGTH_SHORT).show();
}

```

En el método “onActivityResult” se realiza el proceso lógico si el usuario capturó la foto o no, si se capturó se procede a almacenar la foto en una variable de bitmap que esta será enviada a la clase de redimensionar imágenes para el procesamiento.

6.4 Procesamiento de imágenes.

En el procesamiento de imágenes dentro la aplicación móvil se debe realizar diferentes procesos, desde el momento que se toma hasta en el momento donde se guarda la imagen en la base de datos.

Figura 22

Código para redimensionar foto.

```

private Bitmap redimensionarImagen(Bitmap bitmap, float anchoNuevo, float altoNuevo) {
    int ancho=bitmap.getWidth();
    int alto=bitmap.getHeight();
    if(ancho>anchoNuevo || alto>altoNuevo){
        float escalaAncho=anchoNuevo/ancho;
        float escalaAlto= altoNuevo/alto;
        Matrix matrix=new Matrix();
        matrix.postScale(escalaAncho,escalaAlto);
        return Bitmap.createBitmap(bitmap, x: 0, y: 0,ancho,alto,matrix, filter: false);
    }else{
        return bitmap;
    }
}

```

En esta clase se realiza el proceso de estandarizar las imágenes a un tamaño de 1280x960 es brindando que las imágenes cuando se almacenan en la base de datos el tamaño no sean muy grandes teniendo como consecuencia que el sistema se demore en las búsquedas con imagen.

Figura 23

Código para convertir la imagen en base 64 Android.

```
private String convertirImgString(Bitmap bitmap) {
    ByteArrayOutputStream array=new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.PNG, quality: 100,array);
    byte[] imagenByte=array.toByteArray();
    String imagenString= Base64.encodeToString(imagenByte, Base64.DEFAULT);
    return imagenString;
}
```

Para el envío de la imagen a la base de datos por medio de la Rest API se debe realizar una conversión de imagen a base64 así permitiendo enviar la imagen como texto.

Figura 24

Código para convertir de base64 a imagen en php.

```
file_put_contents($path,base64_decode($imagen));
$bytesArchivo=file_get_contents($path);
```

En el archivo de php alojado en el servidor se realizar de convertir de base64 a imagen así permitiendo el almacenamiento en la base de datos de manera exitosa.

Figura 25

Código para convertir la imagen en base 64.

```
$result["Foto"]=base64_encode($registro['Foto']);
```

En el momento que se necesite consultar un elemento y se necesite consultar la imagen de este se deberá convertir la imagen a base64 para enviar como texto por medio de la Rest api allí se será recibirá por la aplicación móvil.

Figura 26

Código para convertir de base64 a imagen Android.

```
public void setData(String dato) {
    this.dato = dato;
    try {
        byte[] byteCode = Base64.decode(dato, Base64.DEFAULT);
        this.imagen = BitmapFactory.decodeByteArray(byteCode, 0, byteCode.length);

        int alto = 450; //alto en pixeles
        int ancho = 510; //ancho en pixeles

        Bitmap foto = BitmapFactory.decodeByteArray(byteCode, 0, byteCode.length);
        this.imagen = Bitmap.createScaledBitmap(foto, ancho, alto, true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

En la recepción de la imagen en la aplicación móvil se deberá realizar la conversión de base64 a imagen, esta para ser visualizada en la interfaz.

6.5 Opciones CRUD

Dentro de la aplicación móvil se encuentran las opciones CRUD que son crear, leer, actualizar y borrar estas son utilizadas para el manejo de los datos en base de datos, en este caso aplican estas opciones para las funciones del módulo de artículos de inventario, para el uso de estas opciones primero en la interfaz de teléfono móvil se visualiza el formulario según la opción a utilizar, allí el usuario ingresa todos los datos y la aplicación se encarga de validar estos, si los datos ingresados son correctos se hace uso de la Rest api para dirigir a los archivos de php que se encuentran en el servidor, estos archivos se encargan de realizar la respectiva opción que seleccionó y por último se obtiene una respuesta con la opción realizada.

Figura 27

Librería para acceso a internet.

```
implementation 'com.android.volley:volley:1.1.1'
```

Para poder interactuar con internet se hace uso de la librería volley.

Figura 28

Envío de información por medio de la Rest api

```
String url = ruta + "/ingresarE.php";
StringRequest request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
```

En el uso de Rest Api se utiliza los method GET Y POST para las distintas opciones.

POST: para el ingreso de información a la base de datos

GET: para consultar información y visualizarla en la aplicación

6.5.1 Añadir elemento

Figura 29

Añadir elemento aplicativo móvil.

```
Map<String, String> parametros = new HashMap<>();
parametros.put("placa", iplaca);
parametros.put("descripcion", ides);
parametros.put("fechac", ifechac);
parametros.put("grupo", igrupo);
parametros.put("costo", icosto);
parametros.put("ubicacion", iubicacion);
parametros.put("imagen", imagen);
parametros.put("estado_u", iestadou);
parametros.put("id_inventario", dato);
parametros.put("cedula", iencargado);

return parametros;
}
};
RequestQueue requestQueue = Volley.newRequestQueue(context: Anadir.this);
requestQueue.add(request);
}
}
```

Figura 30

Almacenar la información por method post.

```

$placa = $_POST["placa"];
$des = $_POST["descripcion"];
$estado="Activo";
$fechac = $_POST["fechac"];
$grupo = $_POST["grupo"];
$costo = $_POST["costo"];
$ubicacion = $_POST["ubicacion"];
$imagen = $_POST["imagen"];
$id_inventario=$_POST["id_inventario"];
$cedula=$_POST["cedula"];
$chequeo="0";
$estado_u = $_POST["estado_u"];

```

Al momento de que el usuario ingrese la información en cada uno de los campos la aplicación valida que todos los campos se hallan llenado siendo así se organizan los datos en variables para ser enviados a Rest Api y ser recibidos por el archivo php en el servidor que este se encargará de agregar la información a la base de datos.

Figura 31

Ingresar datos a la base de datos.

```

$sql="INSERT INTO elementos VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?)";
$stmt=$conexion->prepare($sql);
$stmt->bind_param('isssiisssis',$placa,$des,$estado,$fechac,$grupo,$costo,
$ubicacion,$bytesArchivo,$id_inventario,$cedula,$chequeo,$estado_u);

```

Para ingresar la información a la base de datos se cuenta con consultas preparadas estas con el fin de evitar ataques de inyección SQL.

6.5.2 Actualizar elementos

Figura 32

Organizar información de búsqueda.

```
try {
    jsonObject = json.getJSONObject( index: 0);
    consule.setPlaca(jsonObject.optInt( name: "Placa"));
    consule.setDescripcion(jsonObject.optString( name: "Descripcion"));
    consule.setUbicacion_Elemento(jsonObject.optString( name: "Ubicacion_Elemento"));
    consule.setEstado_Uso(jsonObject.optString( name: "Estado_Uso"));
    consule.setDato(jsonObject.optString( name: "Foto"));
} catch (JSONException e) {
    e.printStackTrace();
}
```

Figura 33

Mostrar información en pantalla de la búsqueda.

```
Tdes.setText("Descripción");
Tubicaion.setText("Ubicación" );
Testado.setVisibility(View.VISIBLE);
descrpcion.setText(consule.getDescripcion());
ubicacion.setText(consule.getUbicacion_Elemento());
descrpcion.setVisibility(View.VISIBLE);
ubicacion.setVisibility(View.VISIBLE);
estado.setVisibility(View.VISIBLE);
foto.setVisibility(View.VISIBLE);
imagen.setVisibility(View.VISIBLE);

des=consule.getDescripcion();
ubi=consule.getUbicacion_Elemento();|
```

Para la opción de actualizar datos sobre algún elemento primero se debe de buscar el elemento a actualizar para cargar los datos que actualmente tiene, para esto se realiza la búsqueda por medio de la Rest api este dando como respuesta la información solicitada de esta manera se procede a clasificar la información para visualizar en la interfaz para que el usuario proceda a actualizar los datos.

Figura 34

Modificar datos en la base de datos.

```
$sql="UPDATE elementos SET Descripcion=?,Ubicacion_Elements=?,Foto=?,Estado_Uso=? WHERE placa=?";
$stmt=$conexion->prepare($sql);
$stmt->bind_param('ssssi',$descripcion,$ubicacion,$bytesArchivo,$estado_u,$placa);
```

Una vez modificados los datos se envían por la Rest api para ser modificados en la base de datos.

6.5.3 Consultar elementos

Figura 35

Búsquedas por el method Get.

```
url = ruta+"/consulta_placa.php?placa=" + dato+"&dependencia="+dependencia;

jsonObjectRequest = new JsonObjectRequest(Request.Method.GET, url, jsonRequest: null, listener: this, errorListener: this);
request.add(jsonObjectRequest);
```

En esta opción se realiza varios tipos de búsquedas sea por: placa, descripción, inventario, lugar físico, para esta se realiza por el método post enviando los parámetros de la búsqueda a solicitar.

Figura 36

Consulta de la búsqueda a la base de datos.

```
mysqli_set_charset($conexion, "utf8");
$query="SELECT Id_Inventario, Nombre,Dependencia FROM inventarios ORDER BY `inventarios`.`Dependencia` ASC";
$resultado=mysqli_query($conexion,$query);

while($registro=mysqli_fetch_array($resultado)){
    $result["Id_Inventario"]=$registro['Id_Inventario'];
    $result["Nombre_Inventario"]=$registro['Nombre'];
    $result["Dependencia"]=$registro['Dependencia'];
    $json['Inventario'][]=$result;
}

mysqli_close($conexion);
echo json_encode($json);
```

Figura 37

Clasificar información de la búsqueda.

```

elementos.setNumero(iten = iten + 1);
elementos.setPlaca(jsonObject.optString( name: "Placa"));
elementos.setDescripcion(jsonObject.optString( name: "Descripcion"));
elementos.setUbicacion_Elemento(jsonObject.optString( name: "Ubicacion_Elemento"));

lista_datos.add(elementos);

```

Figura 38

Mostrar recycler view en la pantalla.

```

// mostrar el recycler View en el activity
ListAdapter listAdapter = new ListAdapter(lista_datos, context: this, (item) → {
    moveToDescription(item);
}

```

En el momento de realizar la solicitud de los datos a buscar se recibe una respuesta de tipo JSON, que contiene la información buscada, en este caso la información se almacena en un Array list para que de esta manera se pueda mostrar una lista recyclerview en la pantalla.

6.5.4 Eliminar solicitud

Figura 39

Eliminar solicitud en base de datos.

```

$sql=" DELETE FROM solicitudes WHERE solicitudes.Id_Solicitud =?";
$stmt=$conexion->prepare($sql);
$stmt->bind_param('i',$Id);

```

La opción de eliminar en el aplicativo móvil se utiliza en el caso de eliminar las solicitudes que realiza el usuario dependencia, el usuario solo debe de dar en el botón eliminar y se enviará la petición de eliminar esa solicitud con el id que le corresponda.

6.6 Cambio de encargado de inventario

En la gestión de inventarios se cuenta con la opción de cambiar de encargado de inventario esto con el fin de asignar un responsable a los elementos de los distintos inventarios.

Figura 40

Cambio de encargado consulta mysql.

```
$sql=("UPDATE elementos SET Cedula=? WHERE Id_Inventario=?");
$stmt=$conexion->prepare($sql);
$stmt->bind_param('ss',$cedula,$id_inventario);
```

Para el cambio de encargado se envía los datos de inventario a cambiar de encargado y la cédula de encargado que debidamente tuvo que haber sido registrado, por medio de la consulta de mysql esta se encarga de cambiar la el campo cédula a cada uno de los elementos de inventario asignado.

6.7 Notificaciones.

Para las notificaciones en la aplicación se utilizan solo en el usuario administrador cuando tiene solicitudes pendientes de lo contrario no.

Figura 41

Librerías para generar notificaciones.

```
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
```

Estas son las librerías esenciales para el uso de las notificaciones en Android.

Figura 42

Código para visualizar notificaciones.

```
builder = new NotificationCompat.Builder(context, id);
Intent actividaditen = new Intent( packageContext: this, Solicitudes_activas.class);
PendingIntent conIntent = PendingIntent.getActivity(
    context: this, requestCode: 0, actividaditen, flags: 0);
builder.setContentTitle("Udecinv").setCategory(Notification.CATEGORY_SERVICE)
    .setSmallIcon(R.drawable.icon_logo_2)
    .setContentTitle("Solicitudes")
    .setContentText("Tiene " + iten + " solicitudes pendientes por revisar")
    .setPriority(NotificationCompat.PRIORITY_HIGH)
    .setCategory(NotificationCompat.CATEGORY_MESSAGE)
    .setDefaults(NotificationCompat.DEFAULT_SOUND)
    //setLargeIcon(BitmapFactory.decodeResource(res, R.drawable.campana))
    .setColor(Color.TRANSPARENT)
    .setVibrate(new long[]{100, 250, 100, 500})
    //iten a la notificacion
    .setContentIntent(conIntent)
    .setAutoCancel(true)
    .setOnlyAlertOnce(true);

Notification notification = builder.build();
notifManager.notify(NOTIFY_ID, notification);
```

En las notificaciones se cuenta con una gran variedad de atributos para su visualización y el diseño que se debe tener, la aplicación se encuentra en una clase que solo se ejecuta cuando se cumple la condición, esta se ejecuta cuando el Administrador tiene solicitudes pendientes por atender.

Al momento de dar clic sobre la notificación está lo redirige a la venta de solicitudes pendientes esto se lleva a cabo por la opción de PendingIntent.



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

Especificación de requisitos de software

Proyecto: Aplicación móvil y sistema de información como plataformas de desarrollo para la gestión de inventarios de la universidad de Cundinamarca seccional Ubaté (Udecinv) caso de estudio programa ingeniería de sistemas



Abril del 2021

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
19/04/2021		Castellanos Nestor Ivan Rodríguez Manuel Stiven Hurtado Ana Lucia	

Documento validado por las partes en fecha:

Por la comunidad	Por la universidad
Universidad de Cundinamarca	Universidad de Cundinamarca seccional Ubaté

Especificación de requisitos de software

Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	6
1.4 Referencias	6
1.5 Resumen	6
2 DESCRIPCIÓN GENERAL	7
2.1 Perspectiva del producto	7
2.2 Funcionalidad del producto	7
2.3 Características de los usuarios	8
2.4 Restricciones	9
2.5 Suposiciones y dependencias	9
3 REQUISITOS ESPECÍFICOS	9
3.1 Requisitos comunes de las interfaces	20
3.1.1 Interfaces de usuario	20
3.1.2 Interfaces de hardware	20
3.1.3 Interfaces de software	21
3.1.4 Interfaces de comunicación	21
3.2 Requisitos funcionales	21
3.2.1 Requisito funcional 1	21
3.2.2 Requisito funcional 2	21
3.2.3 Requisito funcional 3	21
3.2.4 Requisito funcional 4	21
3.2.5 Requisito funcional 5	21
3.2.6 Requisito funcional 6	22
3.2.7 Requisito funcional 7	22
3.2.8 Requisito funcional 8	22

Especificación de requisitos de software

3.2.9	Requisito funcional 9	22
3.2.10	Requisito funcional 10	22
3.2.11	Requisito funcional 11	22
3.2.12	Requisito funcional 12	22
3.2.13	Requisito funcional 13	22
3.2.14	Requisito funcional 14	23
3.2.15	Requisito funcional 15	23
3.2.16	Requisito funcional 16	23
3.2.17	Requisito funcional 17	23
3.2.18	Requisito funcional 18	23
3.2.19	Requisito funcional 19	23
3.2.20	Requisito funcional 20	23
3.2.21	Requisito funcional 21	24
3.2.22	Requisito funcional 22	24
3.2.23	Requisito funcional 23	24
3.2.24	Requisito funcional 24	24
3.3	Requisitos no funcionales	24
3.3.1	Conectividad	24
3.3.2	Base de datos	24
3.3.3	Interfaz	24
3.3.4	Adaptabilidad	25
3.3.5	Usabilidad	25
3.3.6	Velocidad de respuesta	25
3.3.7	Seguridad	25
3.3.8	Pruebas	25
3.3.9	Disponibilidad	25
4	DISEÑO DEL SISTEMA	26
4.1	Diseño sitio web	26

Especificación de requisitos de software

1 Introducción

Las tecnologías de la información y comunicación han revolucionado el sector empresarial y se ha convertido en un reto implementar soluciones tecnológicas que permitan optimizar procedimientos y automatizar diferentes tareas en las áreas administrativas y operacionales. Una gran solución para esto ha sido la creación de los sistemas multiplataforma, los cuales facilitan el acceso a la información, brindan la capacidad de estar disponibles en cualquier dispositivo tecnológico y reducen tiempo y costos económicos para las empresas.

Para el caso de la Universidad de Cundinamarca existe un proceso relacionado a la gestión de inventarios, el cual es fundamental dentro de la institución debido a que por este se lleva el registro y seguimiento de los inventarios activos físicos que hacen parte de su patrimonio. Por tal motivo es de vital importancia para la Universidad de Cundinamarca seccional Ubaté, contar con un software que le permita la gestión de inventarios y optimice este proceso por medio de aplicación móvil y sitio web, donde se sincronice la información en tiempo real, se agilice la entrega y verificación por medio de la etiqueta de código de barras de cada uno de los elementos y se genere reportes del estado de los elementos del inventario con sus diferentes características y notificaciones sobre solicitudes de cambios en el inventario.

Teniendo en cuenta lo anterior se decide desarrollar un sistema para inventarios (Udecinv) que se encuentre disponible en dos plataformas informáticas, aplicación para dispositivos móviles y sitio web, las cuales estarán sincronizadas para mantener la información actualizada en cualquier dispositivo que se use, así mismo tendrán la capacidad de hacer más fácil la entrega y verificación del inventario con tan solo escanear el código de barras del elemento, validando su existencia en este. Por consiguiente, Udecinv permitirá optimizar la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté, logrando implementar fortalecer procesos administrativos, para el desarrollo del proyecto se realiza el proceso de implementación para el programa Ingeniería de sistemas como caso de estudio y con el fin de hacer todas las pruebas necesarias dentro del sistema.

1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información web que permitirá la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté caso de estudio programa de ingeniería de sistemas.

1.2 Alcance

El sistema multiplataforma está diseñado para que todas las dependencias de la universidad de Cundinamarca seccional Ubaté pueden hacer uso de este, por medio de un usuario y contraseña que será asignada por el encargado de almacén que tiene el rol de administrador, en primera instancia Udecinv trabaja con la población objetivo del programa de ingeniería de sistemas de la universidad de Cundinamarca seccional Ubaté.

Se cuenta con dos roles, donde el primer rol es administrador y el segundo rol es para el usuario dependencia (funcionario que tengan a su cargo elementos del inventario), hay que

Especificación de requisitos de software

mencionar que estos roles facilitan el control de la información de los inventarios que se manejan en la seccional, así mismo cada rol cuenta con diferentes módulos en el sitio web y aplicativo móvil respectivamente.

1.3 Personal involucrado

Nombre	Nestor Ivan Castellanos
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante de ingeniería de sistemas
Responsabilidad	Análisis de información, diseño y programación sitio web
Información de contacto	nicastellanos@ucundinamarca.edu.co

Nombre	Manuel Stiven Rodríguez
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante de ingeniería de sistemas
Responsabilidad	Análisis de información, diseño y programación de aplicación móvil
Información de contacto	manuelsrodriguez@ucundinamarca.edu.co

Nombre	Ana Lucia Hurtado
Rol	Asesor de tesis de grado
Categoría Profesional	Magister
Responsabilidad	Revisión documental y pruebas de aceptación.
Información de contacto	alhurtado@ucundinamarca.edu.co

1.4 Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.5 Resumen

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

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

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

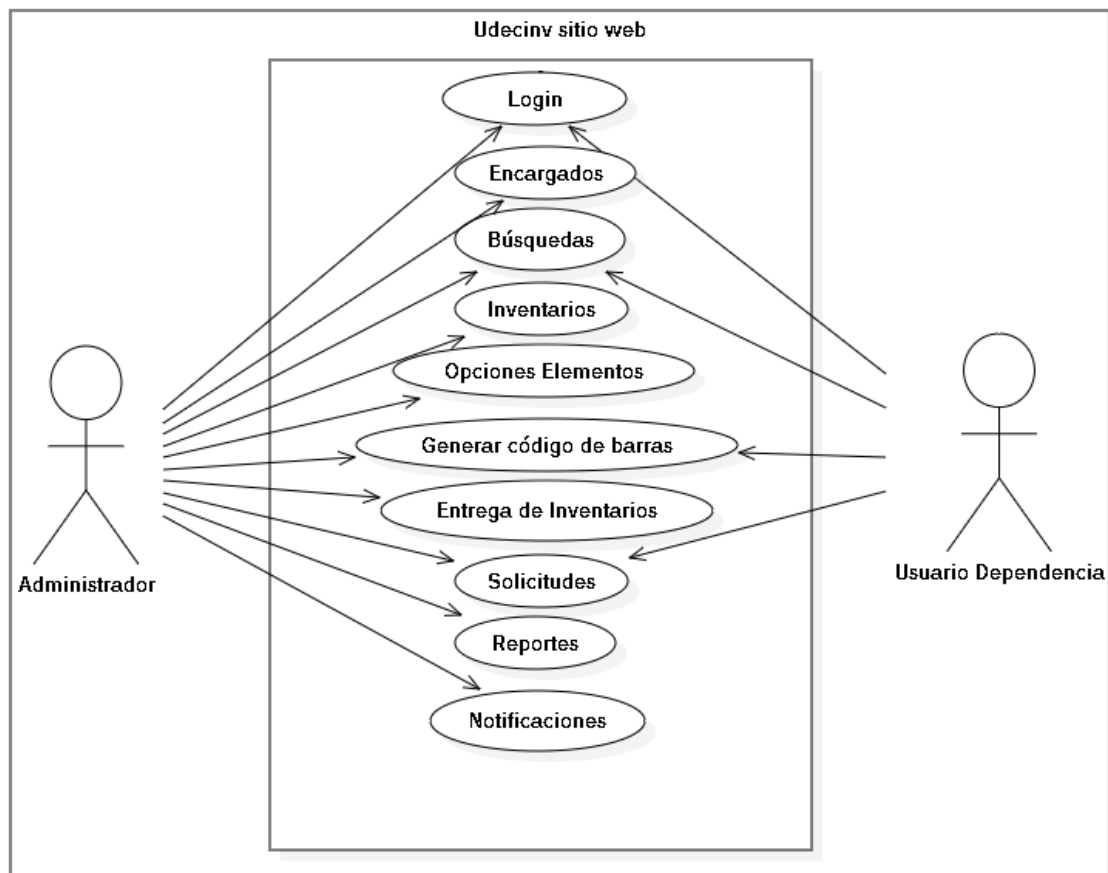
Especificación de requisitos de software

2 Descripción general

2.1 Perspectiva del producto

El sistema multiplataforma Udecinv en el sitio web es un sistema de información que permite la gestión de inventarios para la universidad de Cundinamarca seccional Ubaté que brinda la capacidad de dar un mayor rendimiento en el proceso de verificación de los inventarios para optimizar tiempos.

2.2 Funcionalidad del producto



Especificación de requisitos de software

2.3 Características de los usuarios

Administrador

Actor	Administrador	Identificador	AC01
Descripción	El usuario administrador es el encargado del correcto funcionamiento del sistema multiplataforma, con este rol se puede realizar el proceso de entregas de los elementos de los inventarios por dependencia, agregar y asignar los inventarios al personal administrativo, cuenta con la opción de (Buscar, Agregar y Actualizar) los artículos del inventario. De igual manera podrá generar reportes sobre el checklist de los inventarios de la seccional Ubaté. Finalmente se encarga de llevar el control de las sesiones de los usuarios que cuenten con un inventario a cargo.		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Dependencia, Email y Contraseña.		
Relación	Este rol tiene una relación indirecta con el usuario dependencia por medio del Login, búsquedas, generar código de barras y solicitudes.		
Referencias	Login, encargados, búsquedas, inventarios, opciones elementos, generar código de barras, entrega de inventarios, solicitudes, reportes y notificaciones.		
Atributos			
Nombre	Descripción	Tipo	
Id_Sesion	Identificador primario	Numérico	
Nombres	Nombre del administrador	Alfabético	
Apellidos	Apellido del administrador	Alfabético	
Rol	Tipo de usuario	Alfabético	
Dependencia	Área administrativa que tiene a su cargo varios inventarios.	Alfabético	
Email	Correo institucional	Alfanumérico	
Contraseña	Clave de verificación de usuario	Alfanumérico	

Usuario dependencia

Actor	Usuario Dependencia	Identificador	AC02
Descripción	El usuario dependencia está dirigido para las personas que se les asignan los inventarios por un periodo de tiempo, allí tendrán opciones de búsquedas de los elementos del inventario, realizar solicitudes al administrador para actualizar los elementos del inventario, cuenta con la opción de crear códigos de barras para los elementos que no cuenten con este.		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Dependencia, Email y Contraseña.		
Relación	Este rol tiene una relación indirecta con el usuario administrador por		

Especificación de requisitos de software

	medio del Login, las búsquedas, generar código de barras y las solicitudes.	
Referencias	Login, búsquedas, generar código de barras, notificaciones.	
Atributos		
Nombre	Descripción	Tipo
Id_Sesion	Identificador primario	Numérico
Nombres	Nombre del usuario dependencia	Alfabético
Apellidos	Apellido del usuario dependencia	Alfabético
Rol	Tipo de usuario	Alfabético
Dependencia	Área administrativa que tiene a su cargo varios inventarios.	Alfabético
Email	Correo institucional	Alfanumérico
Contraseña	Clave de verificación de usuario	Alfanumérico

2.4 Restricciones

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

2.5 Suposiciones y dependencias

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

3 Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Roles.
Características:	Se tendrán 2 roles que se clasificaron en dos usuarios, el primero administrador y el segundo el usuario dependencia.

Especificación de requisitos de software

Descripción del requerimiento:	El sistema permitirá el acceso a los 2 tipos de usuarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF08
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Rol Administrador.
Características:	El usuario administrador será el usuario principal y podrá crear usuarios dependencia, en el sitio web.
Descripción del requerimiento:	El sistema brindará diferentes opciones a cada rol donde el rol principal será el rol administrador y este tendrá la mayoría de opciones.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF08
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Funcionalidad del rol administrador.
Características:	El usuario administrador podrá crear y modificar inventarios, encargados, elementos y usuarios dependencia.
Descripción del requerimiento:	Este usuario podrá tener la capacidad de crear nuevos usuario y atributos que sean necesarios dentro de la gestión de inventarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Checklist.
Características:	El administrador podrá crear nuevos checklist para la entrega de inventarios.
Descripción del requerimiento:	Este usuario tendrá las funcionalidades para poder trabajar con checklist.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF05
Prioridad del requerimiento: Alta	

Especificación de requisitos de software

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Entregas de inventario.
Características:	El rol administrador deberá tener el rol de entregas de inventario para realizar el proceso de entrega inicial y final de los diferentes inventarios que cuenta cada dependencia.
Descripción del requerimiento:	Las entregas de inventario podrán realizarse de manera sencilla en el rol administrador con el fin de optimizar este proceso.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF05 • RNF08
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Menú Principal.
Características:	En la ventana principal del administrador se podrá visualizar los inventarios, encargados y usuarios dependencia.
Descripción del requerimiento:	El administrador podrá ver todas las opciones del sistema en la página principal.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF05 • RNF08
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Creación de usuarios.
Características:	Para la creación de usuarios de dependencia se requerirá información como nombres, apellidos, email, dependencia y contraseñas.
Descripción del requerimiento:	Este requerimiento brinda información acerca de la funcionalidad para crear nuevos usuarios dependencia.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF05 • RNF08
Prioridad del requerimiento: Alta	

Especificación de requisitos de software

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Asignación de inventarios y elementos.
Características:	En el módulo administrador se podrá asignar inventarios a los diferentes encargados y elementos a los diferentes inventarios que cuenten con más de un elemento.
Descripción del requerimiento:	El requerimiento brinda información acerca de la asignación de los inventarios a los diferentes encargados y de los elementos a los diferentes inventarios registrados en el sistema.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF08
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Estado de uso de los elementos.
Características:	En el módulo inventarios del rol administrador se podrá visualizar el estado de los elementos de todos los inventarios, clasificados de la siguiente manera (Bueno, Malo, Regular, Mantenimiento).
Descripción del requerimiento:	La información recopilada con este requerimiento busca que se pueda evidenciar el estado de los elementos de los diferentes inventarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF08
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Reporte de checklist.
Características:	El usuario administrador podrá generar y ver los reportes en pdf de los checklist finalizados para su posterior uso en informes.
Descripción del requerimiento:	Se permitirá generar un informe en pdf de los checklist realizados.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF08
Prioridad del requerimiento:	Media

Especificación de requisitos de software

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Foto de los elementos.
Características:	El usuario administrador podrá actualizar las fotos de los elementos desde el sitio web.
Descripción del requerimiento:	El usuario podrá ver y actualizar la foto de cada uno de los elementos con objetivo de que se facilite la entrega de los inventarios
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF08
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Solicitudes.
Características:	El administrador podrá resolver o eliminar solicitudes realizadas por el usuario dependencia, para el cambio de estado de uso o ubicación del elemento.
Descripción del requerimiento:	El usuario administrador es el encargado de actualizar la información que el usuario dependencia necesita y solicita mediante las solicitudes.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF08
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Solicitudes pendientes.
Características:	El administrador podrá ver las notificaciones de las solicitudes pendientes por resolver.
Descripción del requerimiento:	El administrador podrá ver las solicitudes pendientes por resolver las cuales solicito el usuario dependencia.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF08
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Dar de baja a los elementos

Especificación de requisitos de software

Características:	El usuario administrador podrá dar de baja a un elemento modificando su estado a inactivo en la base de datos, permitiendo que este no aparezca en los inventarios.
Descripción del requerimiento:	Dentro del sistema multiplataforma se debe contemplar la opción de quitar un elemento de la lista de inventarios, pasando su estado de activo a inactivo en la base de datos y de esta manera en el proceso de checklist no se tenga en cuenta la placa de este elemento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF08 • RNF10
Prioridad del requerimiento: Media.	

Identificación del requerimiento:	RF15
Nombre del Requerimiento:	Visualizar elementos del inventario.
Características:	El usuario dependencia podrá ver sus inventarios asignados en su página de inicio.
Descripción del requerimiento:	El sistema multiplataforma en sitio web permitirá visualizar los inventarios por dependencia que se encuentre asignado dentro de la pantalla de inicio del rol usuario dependencia.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF07 • RNF08 • RNF10
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF16
Nombre del Requerimiento:	Obtener información de los elementos.
Características:	El usuario dependencia podrá ver toda la información de sus elementos en sus diferentes inventarios asignados, así mismo la foto de cada uno de los elementos.
Descripción del requerimiento:	Dentro del sistema multiplataforma en el usuario dependencia podrá visualizar la información de los elementos por sus distintos atributos (Placa, Descripción, Fecha de compra, Grupo activo, Costo, Estado, Uso, Código de inventario, Ubicación física, Inventario, Encargado, Dependencia y foto).
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03

Especificación de requisitos de software

	<ul style="list-style-type: none"> • RNF04 • RNF05 • RNF07 • RNF08 • RNF10
Prioridad del requerimiento:	
Media	

Identificación del requerimiento:	RF17
Nombre del Requerimiento:	Solicitudes usuario dependencia.
Características:	El usuario dependencia podrá generar solicitudes de los elementos que necesite hacerle una modificación de cambio de lugar físico o estado de uso de los elementos.
Descripción del requerimiento:	Dentro del sistema multiplataforma el usuario dependencia podrá realizar solicitudes al administrador para cambiar el estado de uso del elemento y su ubicación física de este.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF08 • RNF10
Prioridad del requerimiento:	
Media	

Identificación del requerimiento:	RF18
Nombre del Requerimiento:	Solicitudes realizadas.
Características:	El usuario dependencia podrá ver las solicitudes que aun el administrador no ha solucionado, así mismo podrá eliminarlas.
Descripción del requerimiento:	Dentro del sistema multiplataforma se podrá observar las solicitudes realizadas al usuario administrador, también tendrá que tener la opción de eliminar una solicitud realizada.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF08 • RNF10
Prioridad del requerimiento:	
Media	

Identificación del requerimiento:	RF19
Nombre del Requerimiento:	Nueva etiqueta de código de barras

Especificación de requisitos de software

Características:	Los dos usuarios podrán generar y descargar el código de barras para cada uno de los elementos cuando lo requieran en el sitio web.
Descripción del requerimiento:	Los usuarios del sistema podrán generar y descargar una nueva etiqueta con el código de barras del elemento, para aquellos que ya no cuenten con esta.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF08 • RNF10
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF20
Nombre del Requerimiento:	Plataformas de uso.
Características:	El sistema estará en dos plataformas las cuales son plataforma web y en aplicación móvil para Android.
Descripción del requerimiento:	Para el funcionamiento del sistema se deberá realizar en las plataformas de sitio web y aplicación móvil en sistema operativo Android donde deberán contar con una sincronización en tiempo real.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF03 • RNF04 • RNF05 • RNF07 • RNF08 • RNF10
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF21
Nombre del Requerimiento:	Diseño del sistema.
Características:	El color del diseño de la interfaz del sistema en las dos plataformas debe estar basado en los colores institucionales de la universidad de Cundinamarca
Descripción del requerimiento:	Dentro del diseño de la interfaz de usuario debe contar con los colores institucionales con los que cuenta la Universidad de Cundinamarca en su manual institucional.
Requerimiento	<ul style="list-style-type: none"> • RNF05

Especificación de requisitos de software

NO funcional:	<ul style="list-style-type: none"> RNF09
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF22
Nombre del Requerimiento:	Opciones en los elementos
Características:	El usuario administrador podrá ver toda la información de los elementos en sus diferentes inventarios, así mismo la foto de cada uno de los elementos y de igual manera actualizar información pertinente.
Descripción del requerimiento:	El usuario administrador contara con los permisos para ver todos los elementos del inventario que se encuentre agregados y de igual manera poder actualizar algunos de sus atributos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> RNF01 RNF02 RNF08 RNF10
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF23
Nombre del Requerimiento:	Búsquedas de los elementos
Características:	El usuario administrador contará con búsquedas por distintos filtros como: (Placa, inventario, descripción y lugar físico) en sus dos plataformas.
Descripción del requerimiento:	Debe contar con la opción de realizar búsquedas por distintos atributos con los que cuenta el elemento, esto agilizará el proceso de encontrar un elemento dentro de la seccional Ubaté.
Requerimiento NO funcional:	<ul style="list-style-type: none"> RNF01 RNF02 RNF08 RNF10
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF24
Nombre del Requerimiento:	Verificación de chequeo
Características:	El usuario administrador en el módulo de entregas de inventario podrá visualizar los elementos chequeados y chequeados de cada uno de los checklist que se encuentre activos.
Descripción del requerimiento:	Para agilizar el proceso de verificación de los elementos se visualizará los elementos que ya fueron chequeados y no chequeados.
Requerimiento	<ul style="list-style-type: none"> RNF01

Especificación de requisitos de software

NO funcional:	<ul style="list-style-type: none"> • RNF02 • RNF05 • RNF08 • RNF10
Prioridad del requerimiento: Alta	

Requerimientos No Funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Conexión a internet.
Características:	Para el acceso al sistema debe de disponerse de conexión a internet en las dos plataformas.
Descripción del requerimiento:	El sistema debe tener conexión a internet.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Base de datos.
Características:	El motor de base de datos será mysql y podrá ser accedida de manera simultánea por las dos plataformas sin problemas de tiempos de respuesta.
Descripción del requerimiento:	La base de datos debe ser poder accedida por las dos plataformas.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Diseño de la interfaz.
Características:	La información suministrada en las dos plataformas debe ser legible, clara y de fácil comprensión por parte de los usuarios.
Descripción del requerimiento:	El sistema multiplataforma debe ser de fácil comprensión por parte del usuario.
Prioridad del requerimiento: Alta	

Identificación del	RNF04
---------------------------	-------

Especificación de requisitos de software

requerimiento:	
Nombre del Requerimiento:	Adaptabilidad
Características:	El sitio web podrá visualizarse desde cualquier dispositivo de manera tal que su interfaz se adapte a los requerimientos de la pantalla.
Descripción del requerimiento:	El sistema multiplataforma debe adaptarse a cualquier pantalla desde el sitio web.
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Usabilidad
Características:	El sistema multiplataforma debe ser fácil de utilizar e intuitivo para que los usuarios lo usen.
Descripción del requerimiento:	El sistema debe brindar la capacidad de fácil uso por parte de los usuarios.
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RNF06
Nombre del Requerimiento:	Velocidad de respuesta
Características:	El sistema multiplataforma debe proporcionar tiempos de respuesta rápidos en cada uno de los procedimientos requeridos.
Descripción del requerimiento:	El sistema multiplataforma debe contar con una óptima velocidad las opciones CRUD hacia la base de datos.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF07
Nombre del Requerimiento:	Seguridad de los datos
Características:	El sistema debe proporcionar todo tipo de seguridad a la información de los usuarios.
Descripción del requerimiento:	Los datos deberán ser protegidos de la mejor forma para evitar posibles ataques de robo de información.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF08
Nombre del Requerimiento:	Usabilidad
Características:	El sistema debe poderse usar por diferentes usuarios a la vez.

Especificación de requisitos de software

Descripción del requerimiento:	El sistema deberá tener un óptimo rendimiento con varios usuarios realizando peticiones al sistema multiplataforma.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF09
Nombre del Requerimiento:	Pruebas.
Características:	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.
Descripción del requerimiento:	El sistema multiplataforma se le deberán realizar una serie de pruebas don se validarán los campos y funciones de este, también deberán ser aceptado por el cliente que cumpla con el propósito.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF-10
Nombre del Requerimiento:	Navegadores
Características:	Se requiere el uso de navegadores de internet confiables como: Google Chrome, Mozilla, Opera y Microsoft Edge.
Descripción del requerimiento:	Es recomendable de hacer uso de navegadores cotidianos para el uso del sistema en sitio web, esto para brindar una mejor seguridad de este y obtener una mejor experiencia dentro de su uso
Prioridad del requerimiento:	Alta

3.1 Requisitos comunes de las interfaces

3.1.1 Interfaces de usuario

La interfaz del sitio web debe estar adaptada a cualquier pantalla donde se use con el fin de que los usuarios puedan tener una buena experiencia de usabilidad, así mismo debe estar desarrollada de una manera intuitiva donde se logre la comprensión rápida de las funciones por parte del usuario, finalmente el sistema debe implementar los colores institucionales de la Universidad de Cundinamarca.

3.1.2 Interfaces de hardware

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

- Dispositivo móvil o computador con conexión a internet.

Especificación de requisitos de software

- Lector de código de barras.

3.1.3 Interfaces de software

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

3.1.4 Interfaces de comunicación

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

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1

- **Roles:** Se tendrán 2 roles que se clasificaron en dos usuarios, el primero administrador y el segundo el usuario dependencia.

3.2.2 Requisito funcional 2

- **Rol Administrador:** El usuario administrador será el usuario principal y podrá crear usuarios dependencia, en el sitio web.

3.2.3 Requisito funcional 3

- **Funcionalidad rol administrador:** El usuario administrador podrá crear y modificar inventarios, encargados, elementos y usuarios dependencia.

3.2.4 Requisito funcional 4

- **Checklist:** El administrador podrá crear nuevos checklist para la entrega de inventarios.

3.2.5 Requisito funcional 5

- **Entregas de inventario:** El rol administrador deberá tener el rol de entregas de inventario para realizar el proceso de entrega inicial y final de los diferentes inventarios que cuenta cada dependencia.

Especificación de requisitos de software

3.2.6 Requisito funcional 6

- **Menú principal:** En la ventana principal del administrador se podrá visualizar los inventarios, encargados y usuarios dependencia.

3.2.7 Requisito funcional 7

- **Creación de usuarios:** Para la creación de usuarios de dependencia se requerirá información como nombres, apellidos, email, dependencia y contraseñas.

3.2.8 Requisito funcional 8

- **Asignación de inventarios y elementos:** En el módulo administrador se podrá asignar inventarios a los diferentes encargados y elementos a los diferentes inventarios que cuenten con más de un elemento.

3.2.9 Requisito funcional 9

- **Estado de uso de los elementos:** En el módulo inventarios del rol administrador se podrá visualizar el estado de los elementos de todos los inventarios, clasificados de la siguiente manera (Bueno, Malo, Regular, Mantenimiento).

3.2.10 Requisito funcional 10

- **Reporte de checklist:** El usuario administrador podrá generar y ver los reportes en pdf de los checklist finalizados para su posterior uso en informes.

3.2.11 Requisito funcional 11

- **Foto de los elementos:** El usuario administrador podrá actualizar las fotos de los elementos desde el sitio web.

3.2.12 Requisito funcional 12

- **Solicitudes:** El administrador podrá resolver o eliminar solicitudes realizadas por el usuario dependencia, para el cambio de estado de uso o ubicación del elemento.

3.2.13 Requisito funcional 13

Especificación de requisitos de software

- **Solicitudes pendientes:** El administrador podrá ver las notificaciones de las solicitudes pendientes por resolver.

3.2.14 Requisito funcional 14

- **Dar de baja a los elementos:** El usuario administrador podrá dar de baja a un elemento modificando su estado a inactivo en la base de datos, permitiendo que este no aparezca en los inventarios.

3.2.15 Requisito funcional 15

- **Visualizar elementos del inventario:** El usuario dependencia podrá ver sus inventarios asignados en su página de inicio.

3.2.16 Requisito funcional 16

- **Obtener información de los elementos:** El usuario dependencia podrá ver toda la información de sus elementos en sus diferentes inventarios asignados, así mismo la foto de cada uno de los elementos.

3.2.17 Requisito funcional 17

- **Solicitudes usuario dependencia:** El usuario dependencia podrá generar solicitudes de los elementos que necesite hacerle una modificación de cambio de lugar físico o estado de uso de los elementos.

3.2.18 Requisito funcional 18

- **Solicitudes realizadas:** El usuario dependencia podrá ver las solicitudes que aun el administrador no ha solucionado, así mismo podrá eliminarlas.

3.2.19 Requisito funcional 19

- **Nueva etiqueta de código de barras:** Los dos usuarios podrán generar y descargar el código de barras para cada uno de los elementos cuando lo requieran en el sitio web.

3.2.20 Requisito funcional 20

Especificación de requisitos de software

- **Plataformas de uso:** El sistema estará en dos plataformas las cuales son plataforma web y en aplicación móvil para Android.

3.2.21 Requisito funcional 21

- **Diseño del sistema:** El color del diseño de la interfaz del sistema en las dos plataformas debe estar basado en los colores institucionales de la universidad de Cundinamarca.

3.2.22 Requisito funcional 22

- **Opciones en los elementos:** El usuario administrador podrá ver toda la información de los elementos en sus diferentes inventarios, así mismo la foto de cada uno de los elementos y de igual manera actualizar información pertinente.

3.2.23 Requisito funcional 23

- **Búsquedas de los elementos:** El usuario administrador contará con búsquedas por distintos filtros como: (Placa, inventario, descripción y lugar físico) en sus dos plataformas.

3.2.24 Requisito funcional 24

- **Verificación de chequeo:** El usuario administrador en el módulo de entregas de inventario podrá visualizar los elementos chequeados y chequeados de cada uno de los checklist que se encuentre activos.

3.3 Requisitos no funcionales

3.3.1 Conectividad

- Para el acceso al sistema debe de disponerse de conexión a internet en las dos plataformas.

3.3.2 Base de datos

- El motor de base de datos será mysql y podrá ser accedida de manera simultánea por las dos plataformas sin problemas de tiempos de respuesta.

3.3.3 Interfaz

- La información suministrada en las dos plataformas debe ser legible, clara y de fácil comprensión por parte de los usuarios.

Especificación de requisitos de software

3.3.4 Adaptabilidad

- El sitio web podrá visualizarse desde cualquier dispositivo de manera tal que su interfaz se adapte a los requerimientos de la pantalla.

3.3.5 Usabilidad

- El sistema multiplataforma debe ser fácil de utilizar e intuitivo para que los usuarios lo usen.

3.3.6 Velocidad de respuesta

- El sistema multiplataforma debe proporcionar tiempos de respuesta rápidos en cada uno de los procedimientos requeridos.

3.3.7 Seguridad

- El sistema debe proporcionar todo tipo de seguridad a la información de los usuarios.

3.3.8 Pruebas

- El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.

3.3.9 Disponibilidad

- Se requiere el uso de navegadores de internet confiables como: Google Chrome, Mozilla, Opera y Microsoft Edge.

Especificación de requisitos de software

4 Diseño del sistema

4.1 Definición de colores de interfaz



HEX: #000000
RGB: 0 0 0
CMYK: 84 83 73 80



HEX: #4E4B48
RGB: 78 75 72
CMYK: 76 69 68 33



HEX: #DAAA00
RGB: 218 170 0
CMYK: 2 22 100 8



HEX: #F8F8F8
RGB: 248 248 248
CMYK: 4 3 3 0



HEX: #FFFFFF
RGB: 255 255 255
CMYK: 0 0 0 0

4.2 Diseño sitio web

- Login para ingreso al sistema

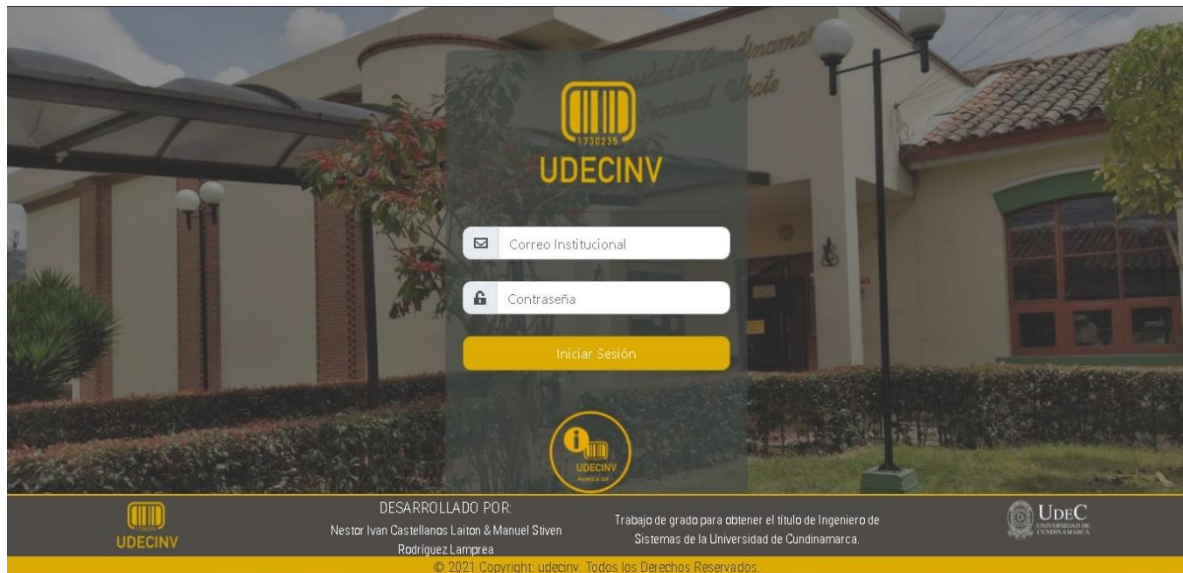


Figura 1 Login

- Pagina principal rol administrador

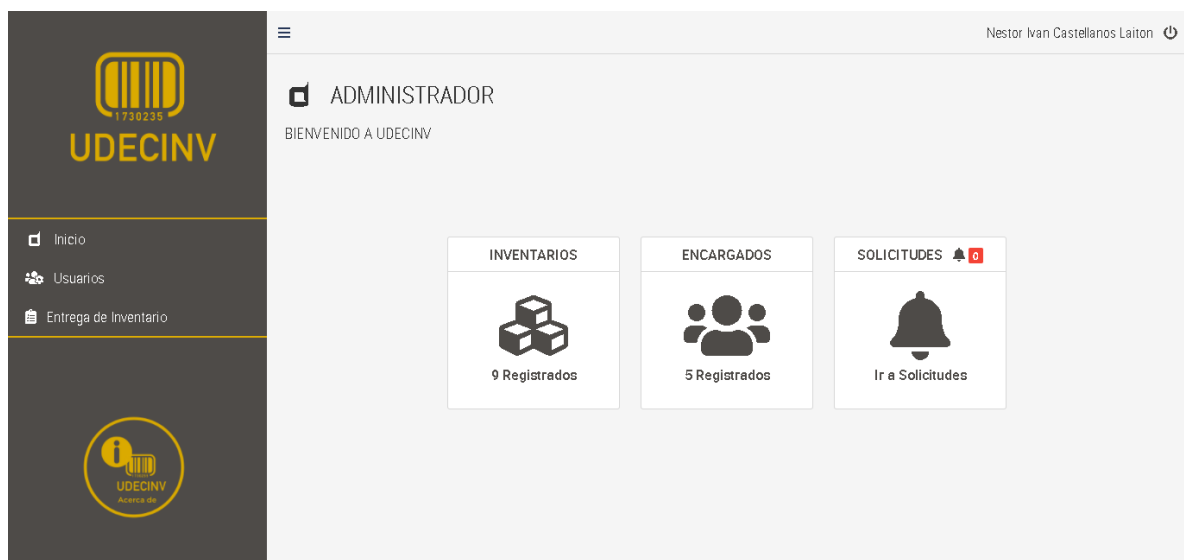


Figura 2 Página principal rol administrador

Especificación de requisitos de software

- Pagina principal rol usuario dependencia



Figura 3 Página principal rol usuario dependencia

- Información de los elementos



Figura 4 Información de los elementos

Especificación de requisitos de software

- Checklist de los inventarios

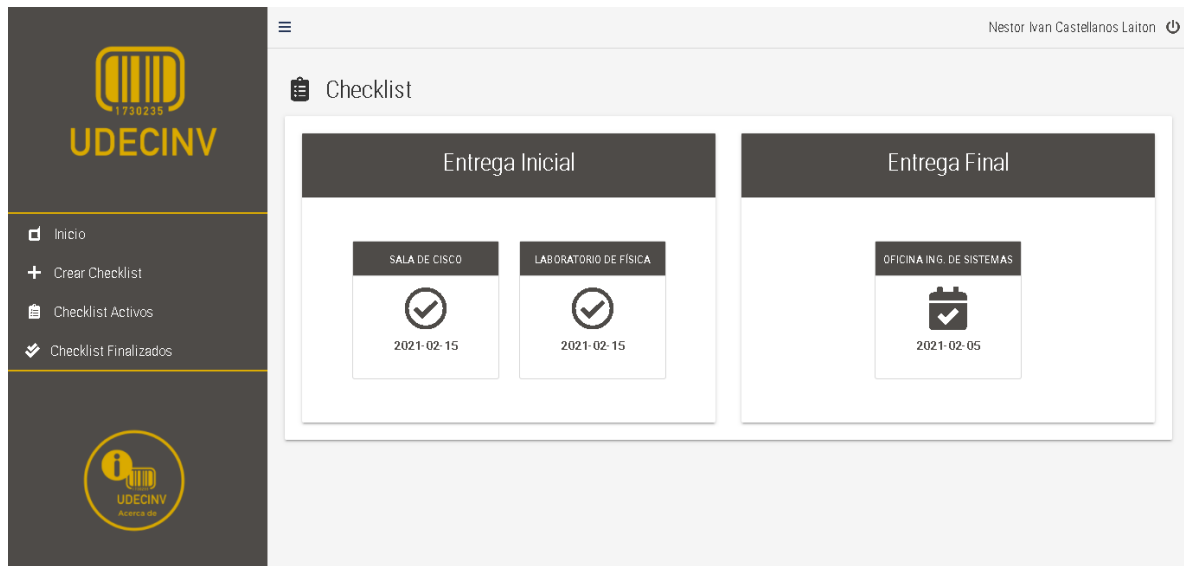


Figura 5 Checklist de los inventarios

- Elementos chequeados

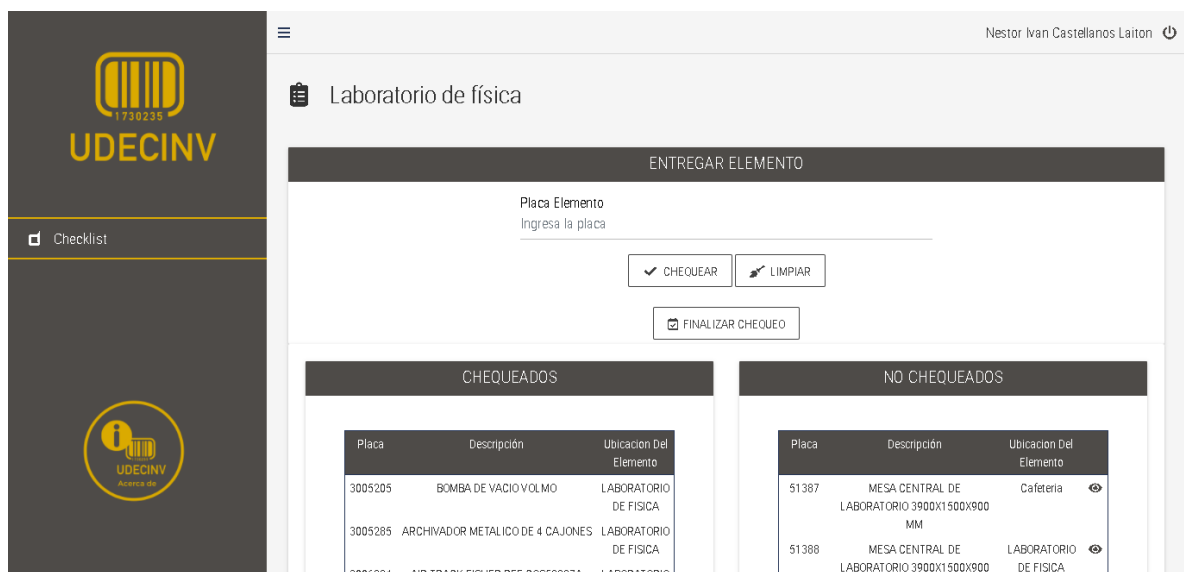


Figura 6 Elementos chequeados



UDEC
UNIVERSIDAD DE
CUNDINAMARCA

Especificación de requisitos de software Aplicación móvil

Proyecto: Aplicación móvil y sistema de información como plataformas de desarrollo para la gestión de inventarios de la universidad de Cundinamarca seccional Ubaté (Udecinv) caso de estudio programa ingeniería de sistemas



Abril del 2021

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. Calidad.
19/04/2021		Castellanos Laiton Nestor Ivan Rodriguez Lamprea Manuel Stiven Hurtado Mesa Ana Lucia	

Documento validado por las partes en fecha:

Por la comunidad	Por la universidad
Universidad de Cundinamarca	Universidad de Cundinamarca seccional Ubaté

Especificación de requisitos de software aplicación móvil

Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	6
1.4 Referencias	6
1.5 Resumen	6
2 DESCRIPCIÓN GENERAL	7
2.1 Perspectiva del producto	7
2.2 Funcionalidad del producto	7
2.3 Características de los usuarios	8
2.4 Restricciones	9
2.5 Suposiciones y dependencias	9
3 REQUISITOS ESPECÍFICOS	9
3.1 Requisitos comunes de las interfaces	20
3.1.1 Interfaces de usuario	20
3.1.2 Interfaces de hardware	20
3.1.3 Interfaces de software	20
3.1.4 Interfaces de comunicación	20
3.2 Requisitos funcionales	20
3.2.1 Requisito funcional 1	20
3.2.2 Requisito funcional 2	21
3.2.3 Requisito funcional 3	21
3.2.4 Requisito funcional 4	21
3.2.5 Requisito funcional 5	21
3.2.6 Requisito funcional 6	21
3.2.7 Requisito funcional 7	21
3.2.8 Requisito funcional 8	21

Especificación de requisitos de software aplicación móvil

3.2.9	Requisito funcional 9	22
3.2.10	Requisito funcional 10	22
3.2.11	Requisito funcional 11	22
3.2.12	Requisito funcional 12	22
3.2.13	Requisito funcional 13	22
3.2.14	Requisito funcional 14	22
3.2.15	Requisito funcional 15	22
3.2.16	Requisito funcional 16	23
3.2.17	Requisito funcional 18	23
3.2.18	Requisito funcional 19	23
3.2.19	Requisito funcional 20	23
3.2.20	Requisito funcional 21	23
3.2.21	Requisito funcional 22	23
3.3	Requisitos no funcionales	23
3.3.1	Conexión a internet.	23
3.3.2	Versión mínima de Android.	24
3.3.3	Base de datos.	24
3.3.4	Diseño de la interfaz.	24
3.3.5	Cámara.	24
3.3.6	Almacenamiento.	24
3.3.7	Usabilidad.	24
3.3.8	Velocidad de respuesta.	24
3.3.9	Accesibilidad.	24
3.3.10	Seguridad.	24
3.3.11	Usabilidad.	25
3.3.12	Pruebas.	25
3.3.13	Conectividad	25
4	DISEÑO DE LA APLICACIÓN MÓVIL.	26
4.1	Definición de Gama de colores.	26
4.2	Diseño de aplicación móvil.	26

Especificación de requisitos de software aplicación móvil

1 Introducción

Las tecnologías de la información y comunicación han revolucionado el sector empresarial y se ha convertido en un reto implementar soluciones tecnológicas que permitan optimizar procedimientos y automatizar diferentes tareas en las áreas administrativas y operacionales. Una gran solución para esto ha sido la creación de los sistemas multiplataforma, los cuales facilitan el acceso a la información, brindan la capacidad de estar disponibles en cualquier dispositivo tecnológico y reducen tiempo y costos económicos para las empresas.

Para el caso de la Universidad de Cundinamarca existe un proceso relacionado a la gestión de inventarios, el cual es fundamental dentro de la institución debido a que por este se lleva el registro y seguimiento de los inventarios activos físicos que hacen parte de su patrimonio. Por tal motivo es de vital importancia para la Universidad de Cundinamarca seccional Ubaté, contar con un software que le permita la gestión de inventarios y optimice este proceso por medio de aplicación móvil y sitio web, donde se sincronice la información en tiempo real, se agilice la entrega y verificación por medio de la etiqueta de código de barras de cada uno de los elementos y se genere reportes del estado de los elementos del inventario con sus diferentes características y notificaciones sobre solicitudes de cambios en el inventario.

Teniendo en cuenta lo anterior se decide desarrollar un sistema para inventarios (Udecinv) que se encuentre disponible en dos plataformas informáticas, aplicación para dispositivos móviles y sitio web, las cuales estarán sincronizadas para mantener la información actualizada en cualquier dispositivo que se use, así mismo tendrán la capacidad de hacer más fácil la entrega y verificación del inventario con tan solo escanear el código de barras del elemento, validando su existencia en este. Por consiguiente, Udecinv permitirá optimizar la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté, logrando implementar fortalecer procesos administrativos, para el desarrollo del proyecto se realiza el proceso de implementación para el programa Ingeniería de sistemas como caso de estudio y con el fin de hacer todas las pruebas necesarias dentro del sistema.

1.1 Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un aplicativo móvil para teléfonos con sistema operativo Android que permitirá la gestión de inventarios de la Universidad de Cundinamarca seccional Ubaté caso de estudio programa de ingeniería de sistemas.

1.2 Alcance

El sistema multiplataforma está diseñado para que todas las dependencias de la universidad de Cundinamarca seccional Ubaté pueden hacer uso de este, por medio de un usuario y contraseña que será asignada por el encargado de almacén que tiene el rol de administrador, en primera instancia Udecinv trabaja con la población objetivo del programa de ingeniería de sistemas de la universidad de Cundinamarca seccional Ubaté.

Se cuenta con dos roles, donde el primer rol es administrador y el segundo rol es para el usuario dependencia (funcionario que tengan a su cargo elementos del inventario), hay que

Especificación de requisitos de software aplicación móvil

mencionar que estos roles facilitan el control de la información de los inventarios que se manejan en la seccional, así mismo cada rol cuenta con diferentes módulos en el sitio web y aplicativo móvil respectivamente.

1.3 Personal involucrado

Nombre	Nestor Ivan Castellanos Laiton
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante de ingeniería de sistemas
Responsabilidad	Análisis de información, diseño y programación sitio web
Información de contacto	nicastellanos@ucundinamarca.edu.co

Nombre	Manuel Stiven Rodriguez Lamprea
Rol	Analista, diseñador y programador
Categoría Profesional	Estudiante de ingeniería de sistemas
Responsabilidad	Análisis de información, diseño y programación de aplicación móvil
Información de contacto	manuelsrodriguez@ucundinamarca.edu.co

Nombre	Ana Lucia Hurtado Mesa
Rol	Asesor de tesis de grado
Categoría Profesional	Magister
Responsabilidad	Directora de proyecto
Información de contacto	alhurtado@ucundinamarca.edu.co

1.4 Referencias

Titulo del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.5 Resumen

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

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

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

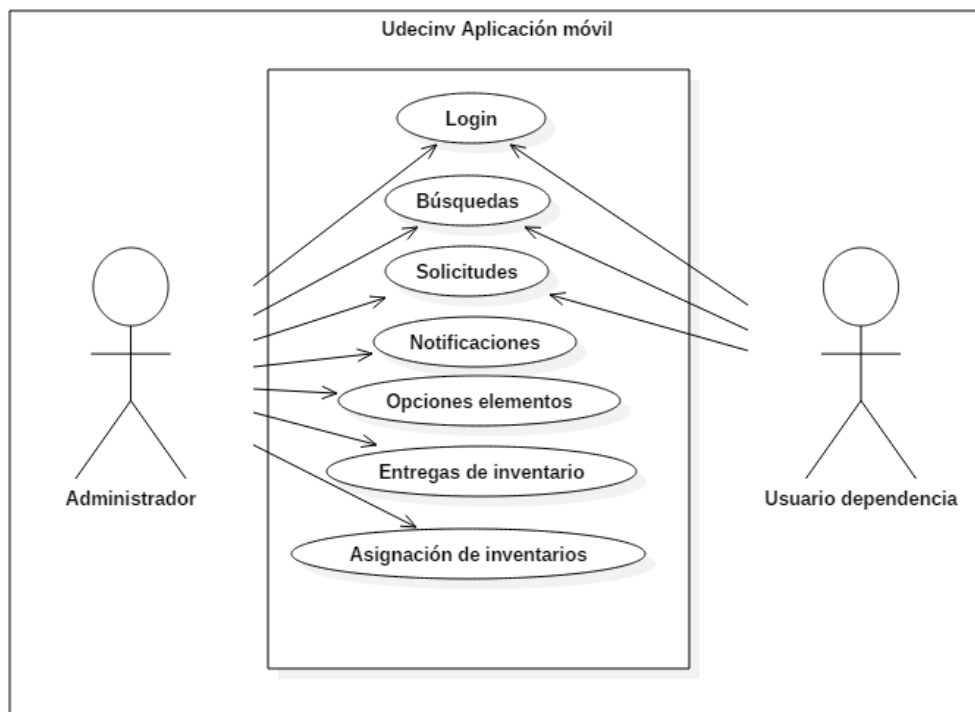
Especificación de requisitos de software aplicación móvil

2 Descripción general

2.1 Perspectiva del producto

El sistema multiplataforma udecinv en aplicativo móvil para dispositivos Android brinda optimizar tiempos el proceso de verificación de los elementos de los inventarios por medio del lector de código de barras con la cámara del teléfono móvil, de igual forma brinda llevar un mejor seguimiento del estado de los elementos.

2.2 Funcionalidad del producto



Especificación de requisitos de software aplicación móvil

2.3 Características de los usuarios

Administrador

Actor	Administrador	Identificador	AC01
Descripción	El usuario administrador es el encargado del correcto funcionamiento del sistema multiplataforma, con este rol se puede realizar el proceso de entregas de los elementos de los inventarios por dependencia, agregar y asignar los inventarios al personal administrativo, cuenta con la opción de (Buscar, Agregar y Actualizar) los artículos del inventario. De igual manera podrá generar reportes sobre el checklist de los inventarios de la seccional Ubaté. Finalmente se encarga de llevar el control de las sesiones de los usuarios que cuenten con un inventario a cargo.		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Dependencia, Email y Contraseña.		
Relación	Este rol tiene una relación indirecta con el usuario dependencia por medio del Login, búsquedas, generar código de barras y solicitudes.		
Referencias	Login, encargados, búsquedas, inventarios, opciones elementos, generar código de barras, entrega de inventarios, solicitudes, reportes y notificaciones.		
Atributos			
Nombre	Descripción	Tipo	
Id_Sesion	Identificador primario	Numérico	
Nombres	Nombre del administrador	Alfabético	
Apellidos	Apellido del administrador	Alfabético	
Rol	Tipo de usuario	Alfabético	
Dependencia	Área administrativa que tiene a su cargo varios inventarios.	Alfabético	
Email	Correo institucional	Alfanumérico	
Contraseña	Clave de verificación de usuario	Alfanumérico	

Usuario dependencia

Actor	Usuario Dependencia	Identificador	AC02
Descripción	El usuario dependencia es para las personas que se le asigna los inventarios de las distintas dependencias administrativas, esta cuenta con funciones como: opciones de búsquedas (placa, descripción, inventario y lugar físico) y realizar solicitudes al administrador para actualizar los elementos del inventario (Lugar físico y cambio de estado de uso).		
Características	El actor cuenta con: Id, Nombres, Apellidos, Rol, Email, Contraseña y Email.		

Especificación de requisitos de software aplicación móvil

Relación	Este rol tiene una relación indirecta con el administrador por medio del Login, Búsquedas y Solicitudes.	
Referencias	Login, búsquedas y solicitudes	
Atributos		
Nombre	Descripción	Tipo
Id_Sesion	Identificador primario	Numérico
Nombres	Nombre del usuario dependencia	Alfabético
Apellidos	Apellido del usuario dependencia	Alfabético
Rol	Tipo de usuario	Alfabético
Email	Correo institucional	Alfanumérico
Contraseña	Clave de verificación de usuario	Alfanumérico
Dependencia	Dependencia administrativa a la que pertenece	Alfanumérico

2.4 Restricciones

- Aplicación móvil para teléfonos con sistema operativo Android.
- Tienda de aplicaciones en play store.
- Lenguajes y tecnologías en uso: Java y PHP.
- El servidor debe ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según un modelo cliente/servidor.
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

2.5 Suposiciones y dependencias

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

3 Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Roles.
Características:	Se tendrán 2 roles que se clasificaron en dos usuarios, el primero administrador y el segundo el usuario dependencia.

Especificación de requisitos de software aplicación móvil

Descripción del requerimiento:	El sistema permitirá el acceso a los 2 tipos de usuarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF03 • RNF13
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Funcionalidad del rol administrador.
Características:	El usuario administrador podrá crear y modificar inventarios, encargados, elementos y usuarios dependencia.
Descripción del requerimiento:	Este usuario podrá tener la capacidad de crear nuevos usuario y atributos que sean necesarios dentro de la gestión de inventarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF06
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Checklist.
Características:	El administrador podrá crear nuevos checklist para la entrega de inventarios.
Descripción del requerimiento:	Este usuario tendrá las funcionalidades para poder trabajar con checklist.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF06
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Entregas de inventario.
Características:	El rol administrador deberá tener el rol de entregas de inventario para realizar el proceso de entrega inicial y final de los diferentes inventarios que cuenta cada dependencia.
Descripción del requerimiento:	Las entregas de inventario podrán realizarse de manera sencilla en el rol administrador con el fin de optimizar este proceso.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF 06 • RNF09
Prioridad del requerimiento: Alta	

Especificación de requisitos de software aplicación móvil

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Menú Principal.
Características:	En la ventana principal del administrador se podrá visualizar los inventarios, encargados y usuarios dependencia.
Descripción del requerimiento:	El administrador podrá ver todas las opciones del sistema en la página principal.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF06 • RNF09
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Asignación de inventarios y elementos.
Características:	En el módulo administrador se podrá asignar inventarios a los diferentes encargados y elementos a los diferentes inventarios que cuenten con más de un elemento.
Descripción del requerimiento:	El requerimiento brinda información acerca de la asignación de los inventarios a los diferentes encargados y de los elementos a los diferentes inventarios registrados en el sistema.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF09
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Estado de uso de los elementos.
Características:	En el módulo inventarios del rol administrador se podrá visualizar el estado de los elementos de todos los inventarios, clasificados de la siguiente manera (Bueno, Malo, Regular, Mantenimiento).
Descripción del requerimiento:	La información recopilada con este requerimiento busca que se pueda evidenciar el estado de los elementos de los diferentes inventarios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF09
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF08
Nombre del	Foto de los elementos.

Especificación de requisitos de software aplicación móvil

Requerimiento:	
Características:	El usuario administrador podrá actualizar las fotos de los elementos desde el sitio web .
Descripción del requerimiento:	El usuario podrá ver y actualizar la foto de cada uno de los elementos con objetivo de que se facilite la entrega de los inventarios
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF04 • RNF09
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Solicitudes.
Características:	El administrador podrá resolver o eliminar solicitudes realizadas por el usuario dependencia, para el cambio de estado de uso o ubicación del elemento.
Descripción del requerimiento:	El usuario administrador es el encargado de actualizar la información que el usuario dependencia necesita y solicita mediante las solicitudes.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF13
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Solicitudes pendientes.
Características:	El administrador podrá ver las notificaciones de las solicitudes pendientes por resolver.
Descripción del requerimiento:	El administrador podrá ver las solicitudes pendientes por resolver las cuales solicito el usuario dependencia.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF03 • RNF13
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Dar de baja a los elementos
Características:	El usuario administrador podrá dar de baja a un elemento modificando su estado a inactivo en la base de datos, permitiendo que este no aparezca en los inventarios.
Descripción del requerimiento:	Dentro del sistema multiplataforma se debe contemplar la opción de quitar un elemento de la lista de inventarios, pasando su estado de activo a inactivo en la base de datos y de esta manera en el proceso de

Especificación de requisitos de software aplicación móvil

	checklist no se tenga en cuenta la placa de este elemento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF03 • RNF09 • RNF10 • RNF13
Prioridad del requerimiento: Media.	

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Obtener información de los elementos.
Características:	El usuario dependencia podrá ver toda la información de sus elementos en sus diferentes inventarios asignados, así mismo la foto de cada uno de los elementos.
Descripción del requerimiento:	Dentro del sistema multiplataforma en el usuario dependencia podrá visualizar la información de los elementos por sus distintos atributos (Placa, Descripción, Fecha de compra, Grupo activo, Costo, Estado, Uso, Código de inventario, Ubicación física, Inventario, Encargado, Dependencia y foto).
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF04 • RNF09 • RNF11 • RNF13
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Solicitudes usuario dependencia.
Características:	El usuario dependencia podrá generar solicitudes de los elementos que necesite hacerle una modificación de cambio de lugar físico o estado de uso de los elementos.
Descripción del requerimiento:	Dentro del sistema multiplataforma el usuario dependencia podrá realizar solicitudes al administrador para cambiar el estado de uso del elemento y su ubicación física de este.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF03 • RNF04 • RNF05 • RNF09 • RNF11 • RNF13
Prioridad del requerimiento: Media	

Especificación de requisitos de software aplicación móvil

Identificación del requerimiento:	RF14
Nombre del Requerimiento:	Solicitudes realizadas.
Características:	El usuario dependencia podrá ver las solicitudes que aun el administrador no ha solucionado, así mismo podrá eliminarlas.
Descripción del requerimiento:	Dentro del sistema multiplataforma se podrá observar la solicitudes realizadas al usuario administrador, también tendrá que tener la opción de eliminar una solicitud realizada.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF04 • RNF09 • RNF11 • RNF13
Prioridad del requerimiento: Media	

Identificación del requerimiento:	RF15
Nombre del Requerimiento:	Lector de código de barras.
Características:	Se dispondrá del uso de la cámara del dispositivo móvil para el escaneo del código de barras de cada uno de los elementos.
Descripción del requerimiento:	Por medio de la aplicación móvil en teléfonos con sistema operativo Android se tendrá en cuenta el uso de la cámara como lector de código de barras para leer las etiquetas de los elementos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF04 • RNF07 • RNF09 • RNF13
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF16
Nombre del Requerimiento:	Plataformas de uso.
Características:	El sistema estará en dos plataformas las cuales son plataforma web y en aplicación móvil para Android.
Descripción del requerimiento:	Para el funcionamiento del sistema se deberá realizar en las plataformas de sitio web y aplicación móvil en sistema operativo Android donde deberán contar con una sincronización en tiempo real.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02

Especificación de requisitos de software aplicación móvil

	<ul style="list-style-type: none"> • RNF03 • RNF05 • RNF07 • RNF08 • RNF09 • RNF10 • RNF11 • RNF12 • RNF13
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF17
Nombre del Requerimiento:	Checklist con lector de código de barras
Características:	El sistema reconocerá el código de barras para el proceso de checklist.
Descripción del requerimiento:	El sistema contara con opción de realizar el proceso de realizar el proceso de verificación con ayuda del lector de código de barras para realizar la verificación de cada uno de los elementos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF06 • RNF08 • RNF09 • RNF11 • RNF12 • RNF13
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF18
Nombre del Requerimiento:	Diseño del sistema.
Características:	El color del diseño de la interfaz del sistema en las dos plataformas debe estar basado en los colores institucionales de la universidad de Cundinamarca
Descripción del requerimiento:	Dentro del diseño de la interfaz de usuario debe contar con los colores institucionales con los que cuenta la Universidad de Cundinamarca en su manual institucional.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF05 • RNF13
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF19
--	------

Especificación de requisitos de software aplicación móvil

Nombre del Requerimiento:	Opciones en los elementos
Características:	El usuario administrador podrá ver toda la información de los elementos en sus diferentes inventarios, así mismo la foto de cada uno de los elementos y de igual manera actualizar información pertinente.
Descripción del requerimiento:	El usuario administrador contara con los permisos para ver todos los elementos del inventario que se encuentre agregados y de igual manera poder actualizar algunos de sus atributos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF04 • RNF08 • RNF09 • RNF11 • RNF13
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF20
Nombre del Requerimiento:	Búsquedas de los elementos
Características:	El usuario administrador contará con búsquedas por distintos filtros como: (Placa, inventario, descripción y lugar físico) en sus dos plataformas.
Descripción del requerimiento:	Debe contar con la opción de realizar búsquedas por distintos atributos con los que cuenta el elemento, esto agilizará el proceso de encontrar un elemento dentro de la seccional Ubaté.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF21
Nombre del Requerimiento:	Verificación de chequeo
Características:	El usuario administrador en el módulo de entregas de inventario podrá visualizar los elementos chequeados y no chequeados de cada uno de los checklist que se encuentren activos.
Descripción del requerimiento:	Para agilizar el proceso de verificación de los elementos se visualizará los elementos que ya fueron chequeados y no chequeados.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF09 • RNF12 • RNF13
Prioridad del requerimiento: Alta	

Especificación de requisitos de software aplicación móvil

Requerimientos No Funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Conexión a internet.
Características:	Para el acceso al sistema debe de disponerse de conexión a internet en las dos plataformas.
Descripción del requerimiento:	El sistema debe tener conexión a internet.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Versión mínima de Android.
Características:	En plataforma móvil los dispositivos deberán tener como mínimo la versión de Android 4.1 para su funcionamiento.
Descripción del requerimiento:	La versión mínima de android debe ser 4.1.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Base de datos.
Características:	El motor de base de datos será mysql y podrá ser accedida de manera simultánea por las dos plataformas sin problemas de tiempos de respuesta.
Descripción del requerimiento:	La base de datos debe ser poder accedida por las dos plataformas.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Diseño de la interfaz.
Características:	La información suministrada en las dos plataformas debe ser legible, clara y de fácil comprensión por parte de los usuarios.
Descripción del requerimiento:	El sistema multiplataforma debe ser de fácil comprensión por parte del usuario.
Prioridad del requerimiento: Alta	

Especificación de requisitos de software aplicación móvil

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Cámara
Características:	La aplicación requerirá una cámara del dispositivo de mínimo 5 megapíxeles.
Descripción del requerimiento:	La cámara de los dispositivos móviles debe ser como mínimo de 5 megapíxeles.
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RNF06
Nombre del Requerimiento:	Almacenamiento.
Características:	El dispositivo móvil debe contar con al menos 20 Mb de almacenamiento para poder instalar la aplicación.
Descripción del requerimiento:	Los dispositivos móviles deben tener la capacidad de almacenar como mínimo 20 Mb para poder instalar la aplicación móvil.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF07
Nombre del Requerimiento:	Usabilidad
Características:	El sistema multiplataforma debe ser fácil de utilizar e intuitivo para que los usuarios lo usen.
Descripción del requerimiento:	El sistema debe brindar la capacidad de fácil uso por parte de los usuarios.
Prioridad del requerimiento:	Media

Identificación del requerimiento:	RNF08
Nombre del Requerimiento:	Velocidad de respuesta
Características:	El sistema multiplataforma debe proporcionar tiempos de respuesta rápidos en cada uno de los procedimientos requeridos.
Descripción del requerimiento:	El sistema multiplataforma debe contar con una óptima velocidad la opciones CRUD hacia la base de datos.
Prioridad del requerimiento:	Alta

Identificación del requerimiento:	RNF9
Nombre del Requerimiento:	Accesibilidad

Especificación de requisitos de software aplicación móvil

Características:	La aplicación debe estar disponible en la play store, para su descarga y posterior instalación.
Descripción del requerimiento:	La aplicación móvil podrá ser descargada dentro de la tienda de aplicaciones play store para teléfonos móviles.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF10
Nombre del Requerimiento:	Seguridad
Características:	El sistema debe proporcionar todo tipo de seguridad a la información de los usuarios.
Descripción del requerimiento:	Los datos deberán ser protegidos de la mejor forma para evitar posibles ataques de robo de información.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF11
Nombre del Requerimiento:	Usabilidad
Características:	El sistema debe poderse usar por diferentes usuarios a la vez.
Descripción del requerimiento:	El sistema deberá tener un óptimo rendimiento con varios usuarios realizando peticiones al sistema multiplataforma.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF12
Nombre del Requerimiento:	Pruebas.
Características:	El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.
Descripción del requerimiento:	El sistema multiplataforma se le deberán realizar una serie de pruebas don se validaran los campos y funciones de este, también deberán ser aceptado por el cliente que cumpla con el propósito.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF13
Nombre del Requerimiento:	Conectividad
Características:	Para el uso de sistema multiplataforma es recomendable tener una conexión a internet con una velocidad mínima a 2 Mbps simétrica.
Descripción del	Dentro del sistema multiplataforma es recomendable tener una buena

Especificación de requisitos de software aplicación móvil

requerimiento:	conectividad a internet para que el uso de este se ha fluido en el momento del uso del mismo
Prioridad del requerimiento:	Alta

3.1 Requisitos comunes de las interfaces

3.1.1 Interfaces de usuario

La interfaz para la aplicación móvil para los usuarios consistirá en un conjunto de ventanas con botones, listas, campos de textos para ingresar la información requerida y aplicación de la cámara del dispositivo. Ésta deberá ser construida para la app y será ejecutado en dispositivos con versión de Android mayor igual a 4.2.

3.1.2 Interfaces de hardware

Será necesario disponer de un teléfono con sistema operativo Android en perfecto estado con las siguientes características:

- Android mayor a 4.1.
- Tarjeta de red wifi.
- Almacenamiento disponible de 20mb.
- Memoria Ram 1gb.
- Cámara Mayor a 4 Mega pixeles.

3.1.3 Interfaces de software

- Sistema Operativo: Android Studio.

3.1.4 Interfaces de comunicación

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

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1

- **Roles:** Se tendrán 2 roles que se clasificaron en dos usuarios, el primero administrador y el segundo el usuario dependencia.

Especificación de requisitos de software aplicación móvil

3.2.2 Requisito funcional 2

- **Funcionalidad del rol administrador:** El usuario administrador podrá crear y modificar inventarios, encargados, elementos y usuarios dependencia.

3.2.3 Requisito funcional 3

- **Checklist:** El administrador podrá crear nuevos checklist para la entrega de inventarios.

3.2.4 Requisito funcional 4

- **Entregas de inventario:** El rol administrador deberá tener el rol de entregas de inventario para realizar el proceso de entrega inicial y final de los diferentes inventarios que cuenta cada dependencia.

3.2.5 Requisito funcional 5

- **Menú Principal:** En la ventana principal del administrador se podrá visualizar los inventarios, encargados y usuarios dependencia.

3.2.6 Requisito funcional 6

- **Asignación de inventarios y elementos:** En el módulo administrador se podrá asignar inventarios a los diferentes encargados y elementos a los diferentes inventarios que cuenten con más de un elemento.

3.2.7 Requisito funcional 7

- **Estado de uso de los elementos:** En el módulo inventarios del rol administrador se podrá visualizar el estado de los elementos de todos los inventarios, clasificados de la siguiente manera (Bueno, Malo, Regular, Mantenimiento).

3.2.8 Requisito funcional 8

- **Foto de los elementos:** El usuario administrador podrá actualizar las fotos de los elementos desde el sitio web.

Especificación de requisitos de software aplicación móvil

3.2.9 Requisito funcional 9

- **Solicitudes:** El administrador podrá resolver o eliminar solicitudes realizadas por el usuario dependencia, para el cambio de estado de uso o ubicación del elemento.

3.2.10 Requisito funcional 10

- **Solicitudes pendientes:** El administrador podrá ver las notificaciones de las solicitudes pendientes por resolver.

3.2.11 Requisito funcional 11

- **Dar de baja a los elementos:** El usuario administrador podrá dar de baja a un elemento modificando su estado a inactivo en la base de datos, permitiendo que este no aparezca en los inventarios.

3.2.12 Requisito funcional 12

- **Obtener información de los elementos:** El usuario dependencia podrá ver toda la información de sus elementos en sus diferentes inventarios asignados, así mismo la foto de cada uno de los elementos.

3.2.13 Requisito funcional 13

- **Solicitudes usuario dependencia:** El usuario dependencia podrá generar solicitudes de los elementos que necesite hacerle una modificación de cambio de lugar físico o estado de uso de los elementos.

3.2.14 Requisito funcional 14

- **Solicitudes realizadas:** El usuario dependencia podrá ver las solicitudes que aun el administrador no ha solucionado, así mismo podrá eliminarlas.

3.2.15 Requisito funcional 15

- **Lector de código de barras:** Se dispondrá del uso de la cámara del dispositivo móvil para el escaneo del código de barras de cada uno de los elementos.

Especificación de requisitos de software aplicación móvil

3.2.16 Requisito funcional 16

- **Plataformas de uso:** Se dispondrá del uso de la cámara del dispositivo móvil para el escaneo del código de barras de cada uno de los elementos.

3.2.17 Requisito funcional 18

- **Checklist con lector de código de barras:** El sistema reconocerá el código de barras para el proceso de checklist.

3.2.18 Requisito funcional 19

- **Diseño del sistema:** El color del diseño de la interfaz del sistema en las dos plataformas debe estar basado en los colores institucionales de la universidad de Cundinamarca.

3.2.19 Requisito funcional 20

- **Opciones en los elementos:** El usuario administrador podrá ver toda la información de los elementos en sus diferentes inventarios, así mismo la foto de cada uno de los elementos y de igual manera actualizar información pertinente.

3.2.20 Requisito funcional 21

- **Búsquedas de los elementos:** El usuario administrador contará con búsquedas por distintos filtros como: (Placa, inventario, descripción y lugar físico) en sus dos plataformas.

3.2.21 Requisito funcional 22

- **Verificación de chequeo:** El usuario administrador en el módulo de entregas de inventario podrá visualizar los elementos chequeados y chequeados de cada uno de los checklist que se encuentre activos.

3.3 Requisitos no funcionales

3.3.1 Conexión a internet.

- Para el acceso al sistema debe de disponerse de conexión a internet en las dos plataformas.

Especificación de requisitos de software aplicación móvil

3.3.2 Versión mínima de Android.

- En plataforma móvil los dispositivos deberán tener como mínimo la versión de Android 4.1 para su funcionamiento.

3.3.3 Base de datos.

- El motor de base de datos será mysql y podrá ser accedida de manera simultánea por las dos plataformas sin problemas de tiempos de respuesta.

3.3.4 Diseño de la interfaz.

- La información suministrada en las dos plataformas debe ser legible, clara y de fácil comprensión por parte de los usuarios.

3.3.5 Cámara.

- La aplicación requerirá una cámara del dispositivo de mínimo 5 megapíxeles.

3.3.6 Almacenamiento.

- El dispositivo móvil debe contar con al menos 20 Mb de almacenamiento para poder instalar la aplicación.

3.3.7 Usabilidad.

- El sistema multiplataforma debe ser fácil de utilizar e intuitivo para que los usuarios lo usen.

3.3.8 Velocidad de respuesta.

- El sistema multiplataforma debe proporcionar tiempos de respuesta rápidos en cada uno de los procedimientos requeridos.

3.3.9 Accesibilidad.

- La aplicación debe estar disponible en la play store, para su descarga y posterior instalación.

3.3.10 Seguridad.

- El sistema debe proporcionar todo tipo de seguridad a la información de los usuarios.

Especificación de requisitos de software aplicación móvil

3.3.11 Usabilidad.

- El sistema debe poderse usar por diferentes usuarios a la vez.

3.3.12 Pruebas.

- El sistema deberá de tener una interfaz de usuario, teniendo en cuenta las características de la web de la institución.

3.3.13 Conectividad

- Para el uso de sistema multiplataforma es recomendable tener una conexión a internet con una velocidad mínima a 2 Mbps simétrica.

Especificación de requisitos de software aplicación móvil

4 Diseño de la aplicación móvil.

4.1 Definición de Gama de colores.



4.2 Diseño de aplicación móvil.

- Login aplicación móvil.

Especificación de requisitos de software aplicación móvil



Figura 1 Login

- Menú principal de usuario administrador y dependencia.



Figura 2 Menú principal aplicación móvil.

- Menú lateral desplegable.

Especificación de requisitos de software aplicación móvil

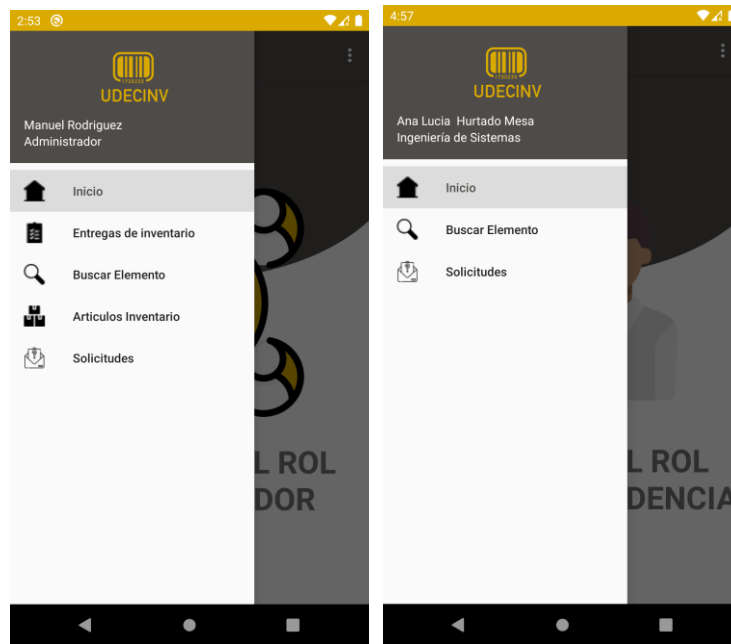


Figura 3 Menú lateral desplegable de usuarios.

- Opciones de inventarios



Figura 4 Opciones de inventario.

- Búsquedas de elementos.

Especificación de requisitos de software aplicación móvil



Figura 5 Búsquedas por atributos

- Checklist de los inventarios

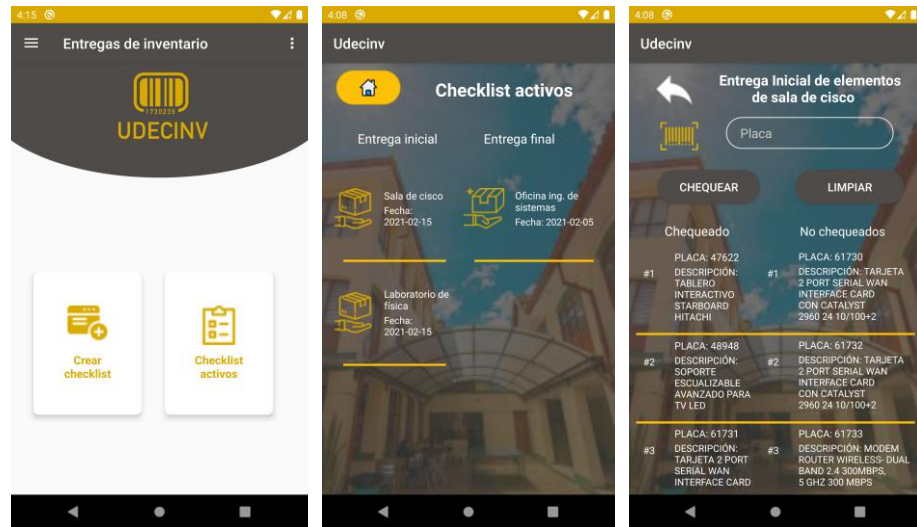


Figura 6 Checklist de inventarios.

- Ver información del elemento.

Especificación de requisitos de software aplicación móvil

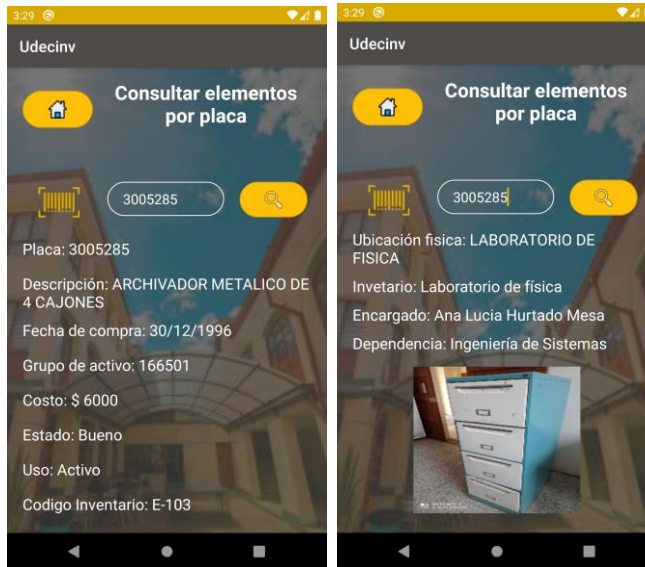


Figura 7 Información del elemento.

- Solicitudes

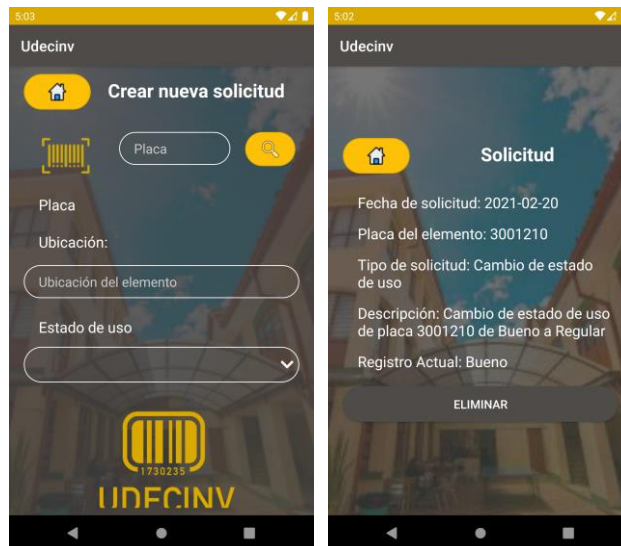


Figura 8 Crear solicitud en usuario dependencia.

Especificación de requisitos de software aplicación móvil

- Notificaciones de solicitudes

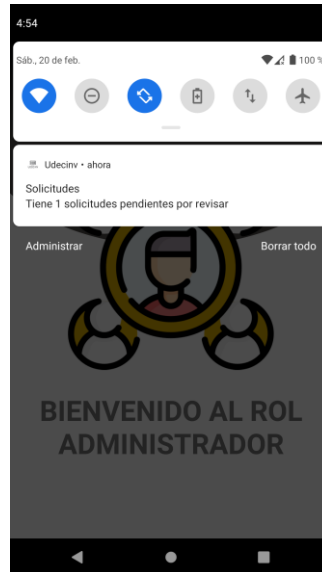


Figura 9 notificaciones en aplicación móvil.



MINISTERIO DEL INTERIOR
DIRECCION NACIONAL DE DERECHO DE AUTOR
UNIDAD ADMINISTRATIVA ESPECIAL
OFICINA DE REGISTRO
CERTIFICADO DE REGISTRO DE SOPORTE LOGICO - SOFTWARE

Libro - Tomo - Partida

13-85-20

Fecha Registro

02-jun.-2021

Page 1 of 2

1. DATOS DE LAS PERSONAS

AUTOR

Nombres y Apellidos	NESTOR IVAN CASTELLANOS LAITON	No de identificación CC	1068954157
Nacional de	COLOMBIA		
Dirección	CALLE 9 N° 4-75	Ciudad:	SIMIJACA

AUTOR

Nombres y Apellidos	MANUEL STIVEN RODRÍGUEZ LAMPREA	No de identificación CC	1073385680
Nacional de	COLOMBIA		
Dirección	CALLE 9 N° 4-75	Ciudad:	SIMIJACA

AUTOR

Nombres y Apellidos	ANA LUCIA HURTADO MESA	No de identificación CC	1076648118
Nacional de	COLOMBIA		
Dirección	CALLE 35 N° 5-11	Ciudad:	CHIQUINQUIRA

PRODUCTOR

Nombres y Apellidos	NESTOR IVAN CASTELLANOS LAITON	No de identificación CC	1068954157
Nacional de	COLOMBIA		
Dirección	CALLE 9 N° 4-75	Ciudad:	SIMIJACA

PRODUCTOR

Nombres y Apellidos	MANUEL STIVEN RODRÍGUEZ LAMPREA	No de identificación CC	1073385680
Nacional de	COLOMBIA		
Dirección	CALLE 9 N° 4-75	Ciudad:	SIMIJACA

PRODUCTOR

Nombres y Apellidos	ANA LUCIA HURTADO MESA	No de identificación CC	1076648118
Nacional de	COLOMBIA		
Dirección	CALLE 35 N° 5-11	Ciudad:	CHIQUINQUIRA

2. DATOS DE LA OBRA

Título Original UDECINV

Año de Creación 2020 País de Origen COLOMBIA Año Edición

CLASE DE OBRA INEDITA

CARACTER DE LA OBRA OBRA ORIGINARIA

CARACTER DE LA OBRA OBRA EN COLABORACION

ELEMENTOS APORTADOS DE SOPORTE LOGICO PROGRAMA DE COMPUTADOR

3. DESCRIPCIÓN DE LA OBRA

UDECINV ES UNA APLICACIÓN PARA ANDROID QUE PERMITE REALIZAR LA GESTIÓN DE INVENTARIOS DE LA UNIVERSIDAD DE CUNDINAMARCA SECCIONAL UBATE, ESTA UTILIZA LA CÁMARA DEL DISPOSITIVO MÓVIL COMO LECTOR DE CÓDIGO DE BARRAS Y REALIZA LAS ENTREGAS DE LOS INVENTARIOS DE LA SECCIONAL.

4. OBSERVACIONES GENERALES DE LA OBRA

5. DATOS DEL SOLICITANTE

Nombres y Apellidos	NESTOR IVAN CASTELLANOS LAITON	No de Identificación	1068954157
Nacional de	COLOMBIA	Medio Radicación	REGISTRO EN LINEA
Dirección	CALLE 9 N° 4-75	Ciudad	SIMIJACA
Correo electrónico	N.NESTEL@HOTMAIL.COM	Teléfono	31114724539
En representación de	EN NOMBRE PROPIO	Radicación de entrada	1-2021-38699