



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

www.bdigital.ula.ve
DISEÑO DE UN SISTEMA DE INVENTARIO PARA EL TALLER
DE ELECTRÓNICA ULA

Br. Kevin José Bottini Ramos

Mérida, marzo, 2021

Reconocimiento-No comercial- Compartir igual



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

www.bdigital.ula.ve
DISEÑO DE UN SISTEMA DE INVENTARIO PARA EL TALLER
DE ELECTRÓNICA ULA

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero
Electricista

Br. Kevin José Bottini Ramos
Tutor(es): Francisco J. Vilorio M.
Emiro J. Ibarra S.

Mérida, marzo, 2021

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA

DISEÑO DE UN SISTEMA DE INVENTARIO PARA EL TALLER DE ELECTRÓNICA ULA

Br. Kevin José Bottini Ramos

Trabajo de Grado, presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

www.bdigital.ula.ve

MSc. Luis Ramon Araujo Rangel

Ing. Marco Antonio Molina Paz

MSc. Francisco Javier Vilorio M.

DEDICATORIA

A Dios.

A mis padres Reinaldo y Tania, los amo.

A mis hermanos Patricia y Brando, los amo.

www.bdigital.ula.ve

AGRADECIMIENTOS

A Dios, gracias por darme salud, sabiduría, paciencia y por colocar en mi camino las personas correctas para emprender este camino de aprendizaje profesional y personal.

A mi papa, gracias por permitirme escoger mi camino y acompañarme durante este proceso, tu amor, apoyo incondicional y tus sabios consejos me acompañan en cada paso que doy. Esto es para ti, aunque no es nada comparado con lo que has hecho por mí.

A mi mama, espero estés orgullosa, sé que espiritualmente me has acompañado en cada decisión que tuve que tomar para completar este sueño y en las circunstancias de más dudas aclaraste mis sentimientos.

A mi hermana Patricia, eres mi gran orgullo, tu fortaleza, amor y talento me han enseñado a ver el mundo desde otra perspectiva, nada es imposible para ti.

A mi hermano Brando, eres parte de mi esencia, al lado de mama espero te encuentres con una gran sonrisa por este logro que hemos alcanzado.

A mi amor Evely, de la mano hemos caminado juntos todo este ciclo, me has brindado alegría, amor, confianza, lealtad y me enseñaste como potenciar mis habilidades para alcanzar mis sueños. Tu apoyo y paciencia ha sido vital para culminar esta etapa y empezar un nuevo capítulo en mi vida, muchas gracias por esa sonrisa que me motiva a lograr más.

A Freya, una persona que me ha acompañado casi toda mi vida y considero como mi hermana mayor, juntos hemos coleccionado una gran cantidad de aventuras y experiencias, espero la vida nos vuelva a reencontrar y poder abrazarnos y reír como siempre.

A Manuel, un gran compañero con el que atravesé este camino de tantos obstáculos y que finalmente se convirtió en mi hermano. Gracias por tu valiosa amistad y por brindarme tanta sabiduría a nivel académico y personal.

Al Sr. Gilberto González, usted fue un segundo padre para mí, cuando inicie este camino me dio un gran consejo, el cual me mantuvo de pie en los momentos más difíciles, definitivamente esta carrera si era de resistencia, gracias.

A mi tutor, Francisco Vilorio, un gran académico, sus conocimientos han contribuido enormemente en mi formación profesional, no solo en este trabajo de grado si no en las materias que tuve la oportunidad de cursar con usted, gracias por su compromiso irrestricto para poder culminar esta etapa de mi vida.

A la ilustre Universidad de Los Andes, por formarme en sus aulas durante toda mi trayectoria académica. Me siento feliz y orgulloso de formar parte de esta casa de estudios.

A todo el personal docente, administrativo y obrero de la Escuela de Ingeniería Eléctrica, por contribuir de una u otra manera en mi formación como profesional y más importante aún como persona.

Y a todos aquellos familiares, amigos y compañeros que de alguna manera contribuyeron en el alcance y realización de este trabajo.

Kevin José Bottini Ramos. Diseño de un sistema de inventario para el taller de electrónica U.L.A. Universidad de Los Andes. Tutor(es): Francisco J. Vilorio M. y Emiro J. Ibarra S. marzo de 2021.

Resumen

En este trabajo de grado se presenta el diseño de una aplicación web en un ambiente local para el Sistema de Inventario del Taller de Electrónica U.L.A (S.I.T.E.U.L.A), el lenguaje de programación principal utilizado para el desarrollo de este sistema fue PHP, en conjunto con el *framework* Laravel. El programa se creó bajo la arquitectura modelo vista controlador y cuenta con una base de datos personalizada gestionada en MySQL, que permite almacenar los datos más relevantes y garantizar el correcto funcionamiento de este aplicativo. La metodología empleada para la construcción del *software* de inventario se denomina “*Scrumban*”, este método es una combinación de dos metodologías ágiles llamadas, *Scrum* y *Kanban*, el motivo principal por el cual se escogió es debido, a la flexibilidad que supone para los desarrolladores, garantizando un avance dinámico en las tareas asignadas y así, acelerar el proceso de desarrollo.

Descriptores: Sistema de inventario, base de datos, aplicación web, modelo vista controlador, *framework*, PHP, Laravel, *Scrumban*, MySQL.

ÍNDICE GENERAL

APROBACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTOS	v
RESUMEN.....	vii
INDICE GENERAL..	viii
INDICE DE FIGURAS	x
INDICE DE TABLAS.....	xii
INTRODUCCIÓN	1
Capítulo	pp.
1. SISTEMA DE INVENTARIO DEL TALLER DE ELECTRÓNICA U.L.A (S.I.T.E.U.L.A)	2
1.1 Formulación del problema.....	2
1.2 Justificación.....	3
1.3 Objetivos.....	3
1.3.1 Generales	3
1.3.2 Específicos	4
1.4 Metodología	4
1.5 Alcances	8
1.6 Limitaciones	9
2. MARCO TEORICO	10
2.1 Antecedentes	10
2.2 Bases de datos	12
2.2.1 Bases de datos relacionales	12
2.3 Sistemas gestores de bases de datos	13
2.3.1 MySQL	13
2.4 Modelo Vista Controlador	13
2.4.1 Modelo	13
2.4.2 Vista	14
2.4.3 Controlador	14
2.4.4 Ciclo de vida de una solicitud MVC	14
2.5 Front-end	15
2.6 Back-end	15
2.7 Aplicación web	16
2.7.1 XAMPP	16
2.7.2 Apache	16
2.7.3 PHP	16

2.7.4	HTML y CSS	16
2.7.5	JavaScript	17
2.8	Framework	17
2.9	Laravel	18
2.10	Arquitectura cliente servidor	19
3.	DISEÑO Y MODELADO DE LA APLICACIÓN	20
3.1	Modelado de requerimientos	20
3.1.1	Requerimientos funcionales	20
3.1.2	Requerimientos no funcionales	21
3.2	Modelado de actores	21
3.3	Casos de uso	23
3.3.1	Descripción de los casos de uso	23
3.3.2	Diagramas de los casos de uso	35
3.4	Diseño del software	38
3.4.1	Diagrama de navegabilidad	38
3.4.2	Diseño arquitectónico del sistema	39
3.4.3	Descripción de los componentes	39
3.5	Modelado de la base de datos	44
4.	IMPLEMENTACIÓN DEL SISTEMA	45
4.1	Herramientas de desarrollo	45
4.2	Control de versiones	46
4.3	Creación del proyecto y construcción de la interfaz	46
4.4	Interacción cliente servidor	56
4.5	Restablecimiento de contraseña	63
5.	VALIDACIONES	65
5.1	Alertas	65
5.2	Pruebas de los casos de uso	71
5.2.1	Pruebas al caso de uso prestar componente	73
5.2.2	Pruebas al caso de uso retornar componente	76
5.2.3	Pruebas al caso de uso entradas al inventario	79
5.2.4	Pruebas al caso de uso salidas del inventario	81
5.3	Roles de usuario	84
6.	CONCLUSIONES Y MEJORAS FUTURAS	85
6.1	Conclusiones.....	85
6.2	Mejoras futuras.....	86
	REFERENCIAS	87

ÍNDICE DE FIGURAS

Figura	pp.
1.1	Proceso Scrumban 6
1.2	Listado de requerimientos preliminares en el tablero Kanban 7
1.3	Avance intermedio en el desarrollo de la aplicación 8
2.1	Ciclo de vida de una petición con la arquitectura MVC 15
2.2	Arquitectura cliente servidor 19
3.1	Diagrama caso de uso gestión de usuarios 36
3.2	Diagrama caso de uso gestión de alumnos 36
3.3	Diagrama caso de uso gestión de componentes 37
3.4	Diagrama caso de uso gestión de inventario 37
3.5	Diagrama de navegación del sistema 38
3.6	Diagrama del despliegue del sistema 39
3.7	Diagrama de los componentes que constituyen el sistema 43
3.8	Modelo entidad-relación de la base de datos del sistema 44
4.1	Login de acceso al aplicativo 47
4.2	Vista principal del aplicativo 47
4.3	Módulo de gestión de usuarios 48
4.4	Módulo de gestión de alumnos 48
4.5	Módulo de gestión de inventario 49
4.6	Generador de código de barras 49
4.7	Módulo de movimientos de inventario 50
4.8	Módulo de entradas al inventario 50
4.9	Módulo de salidas del inventario 51
4.10	Módulo de gestión de componentes 51
4.11	Módulo de consulta de componentes 52
4.12	Módulo de préstamo de componentes 52
4.13	Módulo de retorno de componentes 53
4.14	Modal de registro de usuario 53
4.15	Modal de edición de usuario 54
4.16	Ventana de confirmación de eliminación de usuario 54
4.17	Modal de registro de alumno 54
4.18	Modal de edición de alumno 55
4.19	Ventana de confirmación de eliminación de alumno 55
4.20	Modal de registro de componente 55
4.21	Modal de edición de componente 56
4.22	Modal de registro de componente desde entradas al inventario 56
4.23	Diagrama realizar préstamo (Administrador) 57
4.24	Diagrama realizar devolución (Administrador) 58
4.25	Diagrama entradas al inventario 59

4.26	Diagrama salidas del inventario	60
4.27	Diagrama realizar préstamo (Técnico)	61
4.28	Diagrama realizar devolución (Técnico)	62
4.29	Formulario de <i>login</i>	63
4.30	Formulario de cambio de contraseña	63
4.31	Modal de cambio de contraseña	64
5.1	Alertas del formulario de registro de usuario	65
5.2	Alertas del formulario de edición de usuario	66
5.3	Alertas del formulario de registro de alumno	66
5.4	Alertas del formulario de edición de alumno	67
5.5	Alertas del formulario de registro de componente	67
5.6	Alertas del formulario de edición de componente	68
5.7	Alertas del formulario de registro de componente en el módulo de entradas al inventario.....	68
5.8	Alerta de confirmación de registro de usuario	69
5.9	Alerta de confirmación de actualización de usuario	69
5.10	Alerta de confirmación de eliminación de usuario	69
5.11	Alerta de confirmación de registro de alumno	70
5.12	Alerta de confirmación de actualización de alumno	70
5.13	Alerta de confirmación de eliminación de alumno	70
5.14	Alerta de confirmación de registro de componente	71
5.15	Alerta de confirmación de actualización de componente	71
5.16	Alerta de alumno no registrado (préstamo)	74
5.17	Alerta de componente no registrado (préstamo)	74
5.18	Alerta de valores negativos o cero en el campo unidades (préstamo)	74
5.19	Alerta de exceso de unidades requeridas para el préstamo	75
5.20	Alerta de componente sin unidades disponibles para el préstamo	75
5.21	Alerta de intento de registro de préstamo sin indicar la cedula del alumno	75
5.22	Alerta de registro de préstamo exitoso	76
5.23	Alerta de alumno no registrado (retorno)	77
5.24	Alerta de alumno sin componentes prestados	77
5.25	Alerta de valores negativos o cero en el campo unidades (retorno)	78
5.26	Alerta de devolución parcial	78
5.27	Alerta de devolución total	79
5.28	Alerta para recordar seleccionar el motivo de ingreso	80
5.29	Alerta de componente no registrado (entradas al inventario)	80
5.30	Alerta de valores negativos o cero en el campo unidades (entradas al inventario)...	81
5.31	Alerta de confirmación de ingreso	81
5.32	Alerta para recordar seleccionar el motivo de egreso	82
5.33	Alerta de componente no registrado (salidas del inventario)	83
5.34	Alerta de valores negativos o cero en el campo unidades (salidas del inventario)...	83

5.35	Alerta de confirmación de egreso	84
5.36	Página de acceso no autorizado	84

www.bdigital.ula.ve

ÍNDICE DE TABLAS

Tabla		pp.
3.1	Descripción de los actores del sistema	21
3.2	Caso de uso ingresar al sistema	23
3.3	Caso de uso prestar componente	23
3.4	Caso de uso retornar componente	24
3.5	Caso de uso consultar componente	25
3.6	Caso de uso ingresar componente	26
3.7	Caso de uso editar componente	27
3.8	Caso de uso entradas al inventario	28
3.9	Caso de uso salidas del inventario	29
3.10	Caso de uso generar código de barras	30
3.11	Caso de uso registrar usuario	30
3.12	Caso de uso editar usuario	31
3.13	Caso de uso eliminar usuario	32
3.14	Caso de uso registrar alumno	33
3.15	Caso de uso editar alumno	33
3.16	Caso de uso eliminar alumno	34
3.17	Caso de uso consultar alumno	35
5.1	Pruebas a los casos de uso básicos	71
5.2	Pruebas prestar componente	73
5.3	Pruebas retornar componente	76
5.4	Pruebas entradas al inventario	79
5.5	Pruebas salidas al inventario	82

INTRODUCCIÓN

Actualmente vivimos en una época enfocada en la transformación digital de los procesos productivos desempeñados por los seres humanos, para las empresas u organizaciones en cualquier sector, supone un gran reto la administración de sus productos, por lo tanto, es imprescindible establecer un sistema de inventario.

Ante la situación planteada, surgió la necesidad de automatizar los procedimientos de gestión de inventario del Taller de Electrónica U.L.A, la intención con este proyecto fue realizar un sistema de inventario web que optimizara principalmente, el préstamo y retorno de los componentes electrónicos que utilizan los estudiantes para los proyectos asignados de la carrera, asegurando más eficiencia en el tiempo de ejecución de las distintas actividades relacionadas a este servicio.

El presente manuscrito se compone de seis capítulos. En primer lugar, el capítulo 1, describe la problemática existente con el sistema de inventario actual y, por ende, los motivos que impulsaron la creación de esta herramienta. Además, se indica el tipo de metodología empleada para el desarrollo del *software*. Todos los conceptos teóricos asociados a las herramientas y tecnologías empleadas para el diseño de este sistema, se especifican en el capítulo 2. Adicionalmente, se muestran las principales referencias seleccionadas para la ejecución de este proyecto.

Los capítulos 3 y 4, indican los pasos seguidos para el desarrollo del aplicativo, es decir, el diseño de la base de datos, la construcción de la interfaz, la descripción de los componentes creados para la operación del sistema y las interacciones establecidas entre los usuarios y la aplicación.

Finalmente, los capítulos 5 y 6 contienen todas las pruebas realizadas al sistema, las conclusiones y posibles adiciones, respectivamente.

CAPÍTULO 1

SISTEMA DE INVENTARIO DEL TALLER DE ELECTRÓNICA U.L.A

El presente capítulo describe las bases necesarias para ejecutar este trabajo de grado, estableciendo las distintas razones que motivaron a su realización, las mismas se describen a continuación.

1.1 FORMULACIÓN DEL PROBLEMA

En el ámbito del manejo de bienes y servicios, se requiere del empleo de un sistema organizado y eficiente con el objetivo de garantizar un funcionamiento más ágil en las distintas tareas a desempeñar. Uno de los aspectos más importantes a destacar de cualquier empresa u organización, es la administración del inventario, esta actividad es fundamental y conlleva mucha responsabilidad, por lo tanto, necesita especial atención de todos los individuos involucrados para dicho fin.

Sobre la base de las consideraciones anteriores, se aborda la situación del Taller de Electrónica de la Escuela de Ingeniería Eléctrica de la Universidad de Los Andes (U.L.A), este servicio posee una administración del inventario poco eficiente debido a que, la contabilización y clasificación de los componentes electrónicos almacenados se maneja en una hoja de cálculo de Excel, a su vez, todos los préstamos y retornos que realizan las personas que utilizan este servicio se registran en una hoja de papel, generando que el volumen de préstamos y retornos resulte muy complejo de gestionar.

1.2 JUSTIFICACIÓN

La conformación de una organización que se encargue de la adquisición y distribución de artículos de consumo masivo, requiere metodologías y herramientas altamente eficientes, por lo tanto, resulta imperativo implementar un sistema que permita obtener datos confiables de los bienes adquiridos, para ello se establece el denominado control de inventario, el cual “es una técnica que permite mantener la existencia de los productos a los niveles deseados” [1].

La Escuela de Ingeniería Eléctrica de la Universidad de Los Andes, a través de una oficina denominada Taller de Electrónica, provee un servicio de préstamos de componentes electrónicos, dichos componentes son usados en las prácticas de las diversas asignaturas que integran el pensum de estudio, así como para la investigación o la realización de tesis, etc.

La mayoría de los préstamos se realizan cuando va a empezar una clase práctica, creando una serie de problemas: pérdida de datos, confusión de identidades, colas, etc. Dado el número de estudiantes y el volumen de préstamos de componentes que se pueden presentar simultáneamente, se requiere automatizar el proceso de préstamo y retorno.

La problemática existente con el sistema de inventario actual, se aborda desde el punto de vista de los empleados que laboran en el taller de electrónica de la Escuela de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de Los Andes.

1.3 OBJETIVOS

1.3.1 Generales

- Desarrollar un sistema de información web para el control de inventario, préstamo y retorno de componentes electrónicos.

1.3.2 Específicos

- Analizar y diseñar una herramienta que sea fácil, rápida y de bajo costo que permita llevar un eficaz control en los bienes del taller de electrónica.
- Analizar el sistema actual de inventario, préstamo y retorno de componentes.

- Identificar los errores que se presentan en el sistema actual de gestión del taller, para determinar las necesidades a corregir por la herramienta propuesta.
- Organizar la información recolectada para el diseño de una herramienta que mejore la administración del taller.
- Diseñar e implementar la base de datos que almacene el inventario y toda la información requerida por el sistema de control de inventario.
- Implementar la tecnología de lectura de código de barras para el control del inventario, préstamo y retorno por medio de un escáner.
- Desarrollar un módulo de inventario para tener un control más preciso sobre el inventario.
- Desarrollar un módulo para el listado de los componentes eléctricos en existencia.
- Desarrollar un módulo de préstamo y devolución teniendo así un control de cada préstamo.
- Desarrollar un sistema de ingreso para dar permisos a cada usuario, de acuerdo a la necesidad que solicita en la aplicación.
- Realizar las respectivas pruebas.

1.4 METODOLOGÍA

Mediante un análisis de los requerimientos iniciales y problemas detectados, se realizó una evaluación de los requisitos del software y se construyó un prototipo, el cual se evaluó de forma exhaustiva para luego efectuar la implementación y las pruebas del software final. En el área

del desarrollo de software existen múltiples métodos de planificación y ejecución, los cuales poseen distintas características entre sí, garantizando una adecuación perfecta a las necesidades tanto de los desarrolladores como de los clientes.

Para la planificación y el desarrollo del Sistema de Inventario del Taller de Electrónica de la Universidad de Los Andes (S.I.T.E.U.L.A), se ha establecido el uso de una metodología ágil denominada *Scrumban*. “Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno” [2].

La metodología *Scrumban*, es un método de desarrollo de software híbrido entre *Scrum* y *Kanban*, el cual nos permite planificar los proyectos según se va ejecutando. Las fases que define un proceso de *Scrumban* son las siguientes (ver figura 1-1):

- 1) Reunirse con el cliente y generar una lista de los requerimientos iniciales (*Product Backlog*)
- 2) Establecer prioridades en la lista de tareas a ejecutar de acuerdo a las necesidades del cliente y escribirlas en la columna por hacer del tablero *Kanban* (*Sprint Planning*).
- 3) Establecer un tiempo variante para la realización de las tareas listadas en el tablero y realizar las asignaciones correspondientes a los miembros del equipo.
- 4) Las tareas se empiezan a desarrollar y se van actualizando en las distintas columnas del tablero *Kanban*, de acuerdo al proceso que van atravesando.
- 5) Al cumplir los *sprints*, se realiza la revisión y retroalimentación por el cliente respecto al trabajo realizado.
- 6) Por último, se realizan todas las pruebas correspondientes

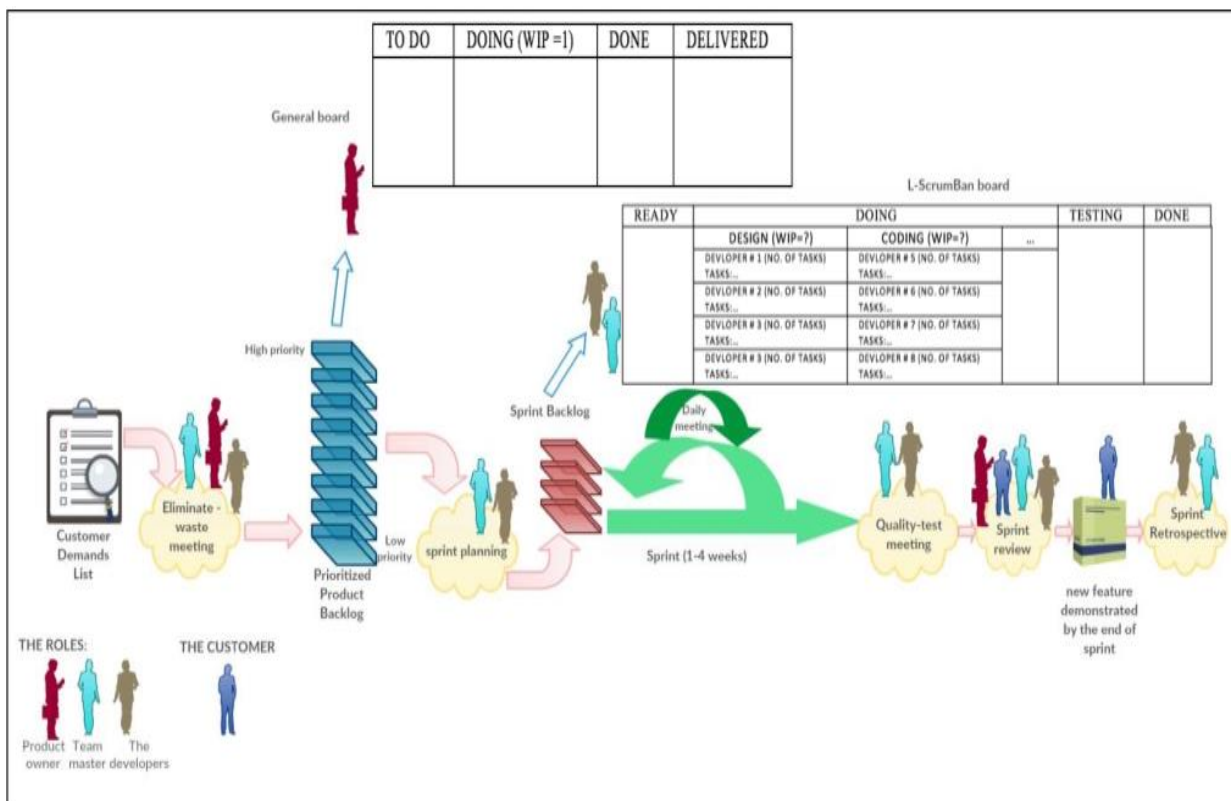


Figura 1.1: Proceso Scrumban – Extraída de The Proposed L-Scrumban Methodology to Improve the Efficiency of Agile Software Development, por Aysha Abdullah Albarqi, Rizwan Qureshi, 2018.

El diseño del tablero *Kanban* permite listar todas las actividades que se requieren cumplir para culminar con el desarrollo del sistema de inventario, para realizar dicho tablero se utilizó la herramienta web lucidchart (ver figura 1.2).

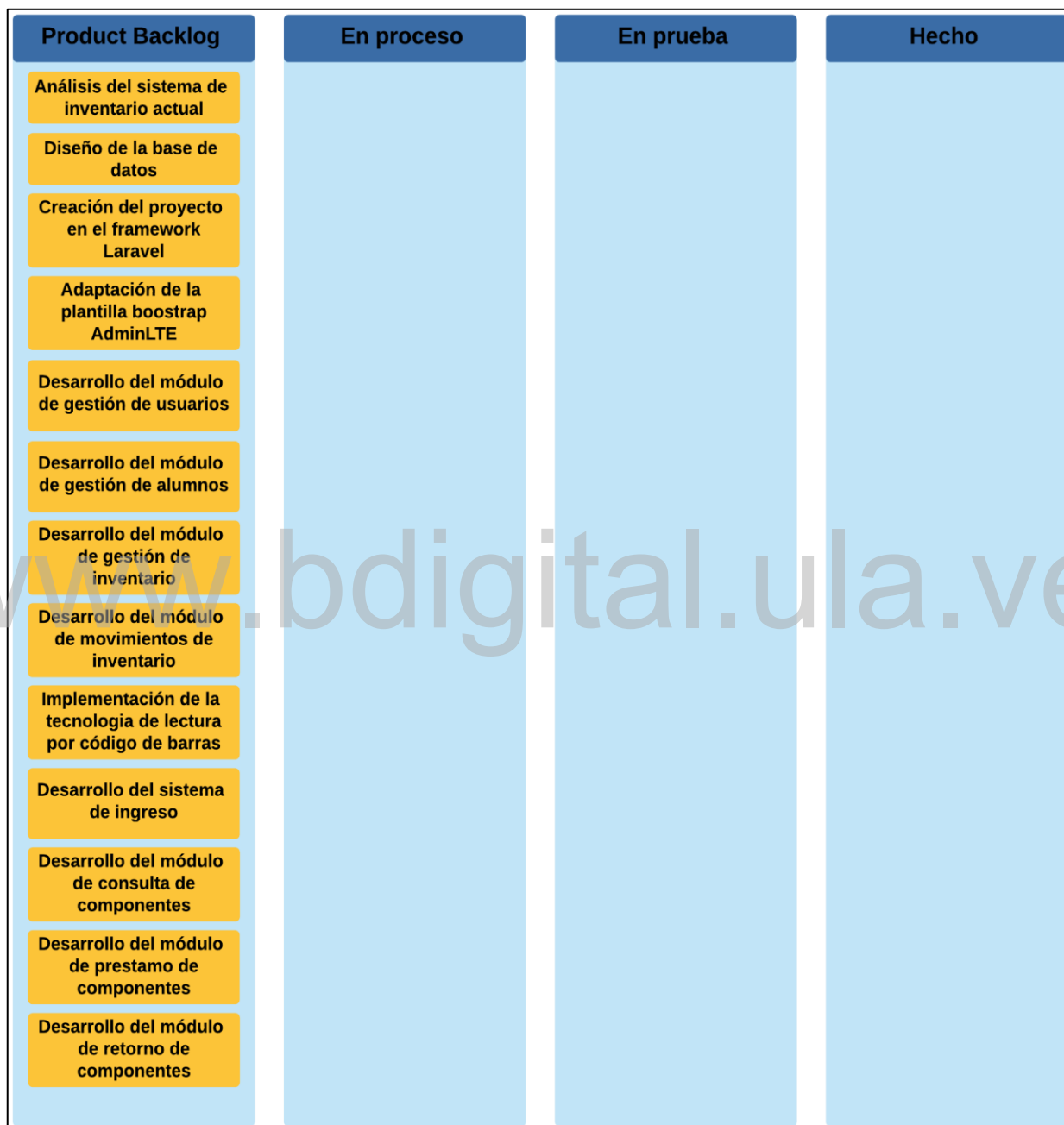


Figura 1.2: Listado de requerimientos preliminares en el tablero Kanban - Elaboración propia.

En la siguiente figura se puede apreciar un avance intermedio en el cumplimiento de los requerimientos esenciales para el desarrollo de la aplicación web.

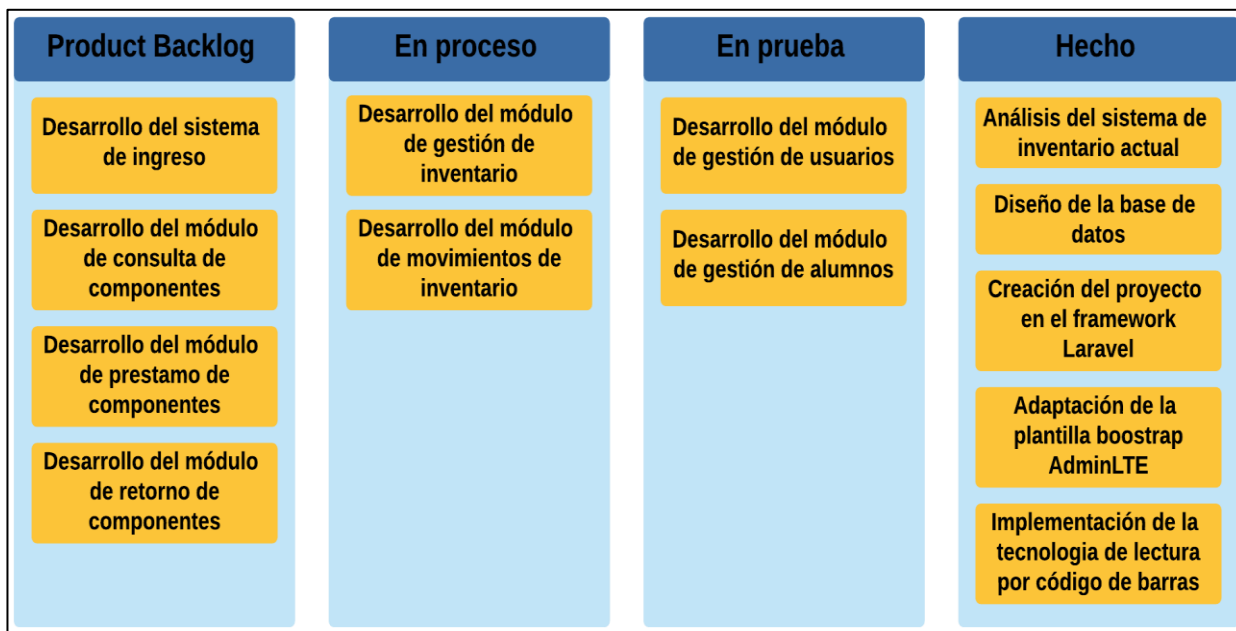


Figura 1.3: Avance intermedio en el desarrollo de la aplicación - Elaboración propia.

1.5 ALCANCES

Los alcances contemplados con el trabajo de grado son los siguientes:

- 1) Diseñar una base de datos que permita almacenar toda la información requerida para el software.
- 2) Diseñar una interfaz gráfica amigable e intuitiva para los usuarios del sistema web.
- 3) Implementar un escáner de código de barras para registrar los préstamos y retornos de componentes electrónicos.
- 4) Generar los códigos de barras para los distintos componentes electrónicos.
- 5) Manejo más preciso del inventario del taller.

1.6 LIMITACIONES

Las limitaciones contempladas con el trabajo de grado son las siguientes:

- 1) El sistema de información web no contará con un servidor físico para instalarse, por lo tanto, se alojará en un servidor local, es decir, en la computadora del taller.
- 2) La aplicación contará con un sistema de recuperación local, debido a los problemas existentes de acceso a internet y de conexión telefónica en la Facultad de Ingeniería.

www.bdigital.ula.ve

CAPÍTULO 2

MARCO TEÓRICO

En este capítulo se definen los fundamentos teóricos necesarios para la comprensión y desarrollo del proyecto. Por lo tanto, se trata de hacer una cuidadosa revisión de los estudios teóricos y prácticos que ya se han realizado y que tienen relación con el problema planteado.

2.1 ANTECEDENTES

Se denomina inventario a la cantidad de materiales o bienes almacenados, empleados para facilitar la producción o para satisfacer las demandas de los consumidores [3].

Un sistema de inventario para una empresa u organización de cualquier rubro, se define como un método de contabilización de la materia prima, suministros o productos terminados con disponibilidad inmediata. Debido a los procesos complejos que conlleva, muchas empresas requieren un *software* o un sistema de gestión de inventario, con la capacidad de incluir todos los aspectos de las actividades del inventario (recepción, adquisición, almacenamiento, entre otros). Independientemente del tipo de sistema de control de inventarios que se escoja, hay que asegurarse de que incluya un sistema para identificar los artículos del inventario y su información (etiquetas de códigos de barras). También es importante que incluya herramientas de hardware para escanear etiquetas códigos de barras, una base de datos central para todo el inventario, además de la capacidad de analizar datos, generar informes y pronosticar la demanda.

Para el desarrollo de este trabajo de grado, se consideraron distintas fuentes referenciales sobre sistemas de gestión de inventarios, préstamo y retorno de materiales o recursos electrónicos.

Entre las investigaciones destacadas en el área, se pueden mencionar:

- Estela Morales Feliz (2016) elaboró el proyecto titulado: “Diseño de una aplicación de gestión de préstamo del material electrónico para los laboratorios de la Universidad Politécnica de Cataluña (U.P.C)”. Este proyecto plantea el desarrollo de una aplicación mediante el uso de la metodología de desarrollo de software denominada *Kanban*, con el objetivo de sustituir el mecanismo de gestión de préstamos a través de hojas de cálculo en Excel, por un sistema automatizado que acelere el volumen de préstamos de material electrónico de la Escuela de Ingeniería de Telecomunicaciones y Aeroespacial de Castelldefels (EETAC) [5].

Dentro de esta investigación existe gran analogía respecto a la problemática sobre el mecanismo de gestión de inventario requerido en este trabajo de grado, por lo tanto, sirvió como guía principal para la confección del sistema de administración del material electrónico y para el diseño de la interfaz gráfica de la aplicación web, específicamente para los formularios de los módulos de préstamo, retorno y gestión de los componentes electrónicos.

- Javier Sánchez Velamazán (2015) desarrolló el proyecto titulado: “Diseño de un inventario web de productos químicos para la facultad de química”. El presente trabajo de titulación, propone el diseño e implementación de una herramienta web, empleando las metodologías de *Scrum* y *Kanban*, las cuales forman parte de las metodologías ágiles del desarrollo de software. La aplicación permitirá administrar el inventario de los productos químicos de la Facultad de Química de la Universidad de Barcelona, facilitando el trabajo de los técnicos de los laboratorios e incluso de los profesores al momento de consultar los almacenes [6].

Este proyecto ayudó a definir los distintos niveles de acceso en la aplicación de los actores involucrados. Al mismo tiempo, proporcionó el mecanismo de selección de la metodología más adecuada a implementar para el desarrollo del software de inventario (*Scrumban*). Finalmente, mediante una serie de comparaciones realizadas entre los *frameworks* más utilizados en el desarrollo web, se seleccionó el mejor *framework* de

PHP para el diseño de la aplicación (Laravel), además de exponer todos los fundamentos teóricos asociados a dicha herramienta.

- Yesid Orduz Navarrete (2016) elaboró el proyecto titulado: “Análisis, diseño e implementación de un sistema de información para la gestión de alquiler y mantenimiento de vehículos”. En este trabajo, se desarrolla un software administrativo para la empresa Transportes Zambrano, en búsqueda de optimizar la forma de operación y dejar atrás el uso exclusivo de registros físicos, dicha aplicación ofrece un control de todos los procesos operativos ejecutados por todos los individuos asociados a la organización [7].

Los aportes más significativos de este proyecto con relación al diseño de la aplicación de inventario para el Taller de Electrónica U.L.A son los siguientes:

- Exposición detallada de los distintos casos de uso, de acuerdo a todos los actores involucrados sobre el sistema y a las acciones a ejecutar.
- Explicación sobre el diseño y modelado de la base de datos requerida para el funcionamiento de la aplicación.

2.2 BASES DE DATOS

Es un contenedor de información especializada, el cual se encuentra sujeto a interacciones entre los datos almacenados. Las bases de datos ofrecen la posibilidad de intervenir en la edición, inserción, eliminación, actualización y creación de tablas, campos, registros y datos mediante un lenguaje de programación. Una base de datos reduce la posibilidad de redundancia o duplicación de los datos, mejorando el aprovechamiento del espacio [8].

2.2.1 Bases de datos relacionales

El modelo relacional se basa en la definición de relaciones entre las distintas tablas de la base de datos que compone un sistema. Estas relaciones son lógicas y ayudan a estructurar mejor la información. Por ejemplo, el proceso de préstamo de una biblioteca requiere del uso de bases de datos relacionales, por lo tanto, tiene que existir una tabla especializada en el registro de

préstamos y devoluciones, que guarda el número de identificación del usuario y del documento que se le presta. Ese número de identificación hace posible recuperar a su vez los datos del usuario, en una tabla distinta [9].

2.3 SISTEMAS GESTORES DE BASES DE DATOS

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente [10].

2.3.1 MySQL

Es un sistema de administración de bases de datos relacionales, es decir, archiva datos en tablas separadas en lugar de guardar todos los datos en un gran archivo, lo que le permite tener mayor velocidad y flexibilidad [11].

2.4 MODELO VISTA CONTROLADOR

La arquitectura Modelo Vista Controlador (MVC) es un patrón de diseño en el área del desarrollo de software, el cual permite dividir la estructura funcional de una aplicación en tres (3) módulos. Esta arquitectura basada en capas logra separar el código en función de sus responsabilidades o conceptos [12].

2.4.1 Modelo

Es la capa donde se trabaja con los datos, por lo tanto, tendrá mecanismos para acceder a la información. Los datos se almacenan habitualmente en una base de datos, por lo que en los modelos se tienen todas las funciones que accederán a las tablas y ejecutarán las acciones requeridas por el usuario [12].

2.4.2 Vista

Es la capa que contiene el código que va a generar la visualización de las interfaces de usuario en la aplicación. En las vistas generalmente se trabaja con los datos, sin embargo, no se realiza un acceso directo a éstos [12].

2.4.3 Controlador

Es la capa que funge de intermediario o enlace entre la capa modelo y la de vista, contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida [12].

2.4.4 Ciclo de vida de una solicitud MVC

- 1) El usuario a través de la vista realiza una solicitud al sistema.
- 2) El controlador recibe dicha solicitud.
- 3) El controlador procesa dicha solicitud y llama al modelo.
- 4) El modelo se comunica e interactúa con la base de datos y retorna la información al controlador.
- 5) El controlador recibe la información y la envía a la vista.
- 6) La vista procesa dicha información y la muestra de forma entendible para el usuario.

En la figura 2.1 se ilustra el ciclo detallado de una petición MVC.

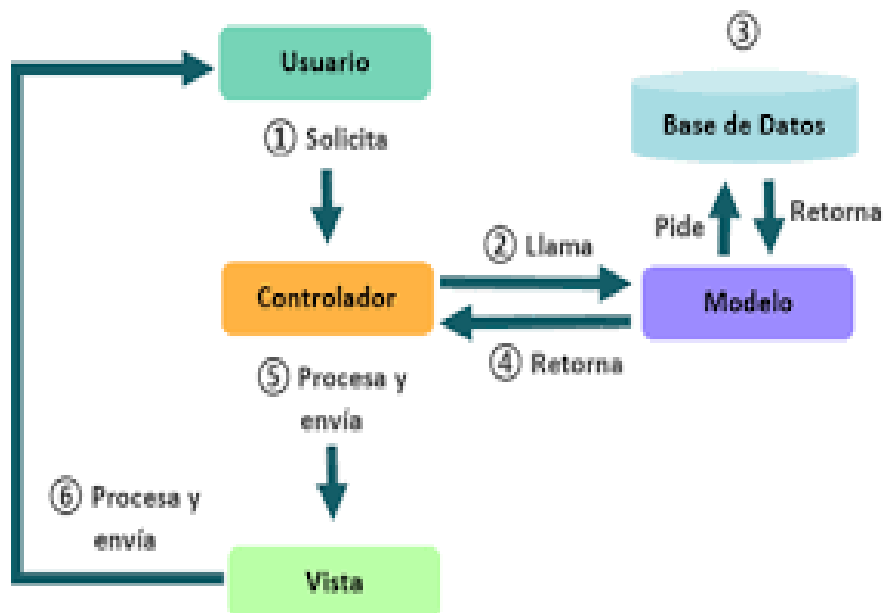


Figura 2.1: Ciclo de vida de una petición con la arquitectura MVC – Extraída de <http://rodrigogr.com/blog/modelo-vista-controlador/>

2.5 FRONT-END

Se define como la parte del desarrollo web con la cual interactúan directamente los usuarios, normalmente estas tecnologías son desarrolladas en los lenguajes de HTML, CSS y JavaScript. El objetivo de este elemento, es desarrollar una interfaz gráfica de usuario que garantice una experiencia de uso bien valorada por el usuario final [13].

2.6 BACK-END

Es el área del desarrollo web que se encarga de toda la lógica de programación, es decir, es el elemento diseñado para garantizar que todo funcione como debería, por ejemplo: la interacción con la base de datos, la verificación y el manejo de sesiones de usuarios, la seguridad de la información que se manipule, entre otras [14].

2.7 APLICACIÓN WEB

Se denomina aplicación web a un programa informático capaz de brindar herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet (red informática interna) mediante un navegador [15].

2.7.1 XAMPP

Es un paquete de software libre diseñado para los desarrolladores web, con el objetivo de testear el trabajo que realizan. Las herramientas que constituyen este programa son: un servidor web Apache, una base de datos MySQL y los intérpretes para los lenguajes PHP y Perl [16].

2.7.2 Apache

Es un software de servidor web gratuito encargado de aceptar las peticiones solicitadas a través de los navegadores para luego mostrar el contenido al usuario [17].

2.7.3 PHP

PHP es un lenguaje de programación de código abierto diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno [18].

2.7.4 HTML y CSS

Son los lenguajes que permiten crear sitios web. Cuando se accede a un sitio web y se realiza una petición a través del navegador, el navegador cumple la función de traductor de las líneas de código escritas en estos lenguajes para saber que debe mostrar con exactitud [19].

- HTML (*Hypertext Markup Language*): Es la forma como se organiza o se estructura la información en una página web. Es la que ordena todo el contenido desde la ubicación del título hasta el pie de página [19].

- CSS (*Cascading Style Sheets*): Es el lenguaje con el que puedes crear estilos al contenido de la página, de esta manera la página no se verá como letras una encima de la otra, sino con cuadros, estilos, diagramas y demás cosas agradables ante el ojo humano, que la hace ser atractiva [19].

2.7.5 JavaScript

Es uno de los tres principales lenguajes para el desarrollo web en conjunto con HTML y CSS. JavaScript fue creado para agregar capacidades interactivas a los sitios web, brindando una interfaz de usuario activa y garantizando retroalimentación a los visitantes de páginas web. Un ejemplo notable del uso de este lenguaje, es la validación de formularios para asegurar que la información introducida es correcta, sin la necesidad de enviar ninguna información al servidor [20].

- jQuery: Es una librería de código abierto del lenguaje JavaScript, una de sus principales virtudes es el desarrollo web empleando menos líneas de código hechas con JavaScript puro. Sin embargo, la función principal de esta biblioteca es mejorar el nivel de interacción con el usuario [21].
- Ajax: Es una técnica de comunicación entre el cliente y el servidor sin necesidad de refrescar la página, es decir, un usuario puede realizar una petición al servidor, ésta se ejecuta en segundo plano permitiendo que se desarrollen otras acciones en la página mientras la solicitud es respondida, en otras palabras, la página permanece estática visualmente en todo momento [22].

2.8 FRAMEWORK

En el mundo de la informática y programación, se define como una pieza de software o entorno de trabajo que proporciona a los desarrolladores web una base de código y formas consistentes y estandarizadas para crear aplicaciones web, brindando a los programadores una mejor forma de codificación [23].

2.9 LARAVEL

Es un entorno o marco de trabajo (*framework*) de código abierto. Laravel sigue el patrón MVC, posee una gran cantidad de componentes adicionales para agilizar el diseño de aplicaciones web [24], algunos de los componentes más destacados se describen a continuación:

- *Composer*: Es un gestor de dependencias de PHP, encargado principalmente de crear el proyecto o la estructura de directorios del *framework*, además tiene la capacidad de instalar paquetes y librerías que se requieran para el desarrollo de las aplicaciones [25].
- *Artesano*: Es una consola que permite ejecutar los comandos internos predeterminados en el *framework* [25].
- *Rutas*: Permiten administrar de forma más sencilla, el flujo de solicitudes y respuestas, desde y hacia el cliente [25].
- *Migraciones*: Permiten realizar la creación de las bases de datos, gestionando las bases de datos desde código PHP para no tener que ir cambiando de SQL a PHP continuamente [25].
- *Semilleros*: Permiten cargar información a las bases de datos para no tener que insertar los datos de prueba de forma manual [25].
- *Eloquent*: Esta herramienta permite realizar mapeos en las bases de datos, con la finalidad tener un nivel de interacción mucho más alto sin la necesidad de escribir código SQL de forma manual [25].
- *Middlewares*: Son mecanismos que se usan para filtrar solicitudes HTTP en una aplicación (Hypertext Transfer Protocol) [25].
- *Requests*: Permiten validar la información que se envía a través de formularios, con el objetivo de no guardar información inútil dentro de las bases de datos [25].
- *Plantillas Blade*: Es un sistema de plantillas que posee Laravel, el cual permite generar código HTML dinámico con una sintaxis más limpia [25].

2.10 ARQUITECTURA CLIENTE SERVIDOR

Es un modelo de comunicación, usado en muchos servicios y protocolos establecidos de Internet para mantener la integridad y seguridad de la información, dos actores fundamentales conforman esta arquitectura: cliente y servidor. El cliente es una entidad que demanda un servicio, puede ser un ordenador o una aplicación informática, mientras que el servidor es una maquina con un hardware bastante potente y software específico para almacenar datos, gestionar base de datos o aplicaciones, esta entidad se encarga de dar respuesta a las solicitudes hechas por el cliente [26].

Dentro de la arquitectura cliente-servidor existen tres tipos:

- Arquitectura de dos capas.
- Arquitectura de tres capas.
- Arquitectura de n capas.

El desarrollo de este trabajo de grado se centrará en la arquitectura de dos capas, esta se utiliza para describir sistemas cliente servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud con sus propios recursos [26].

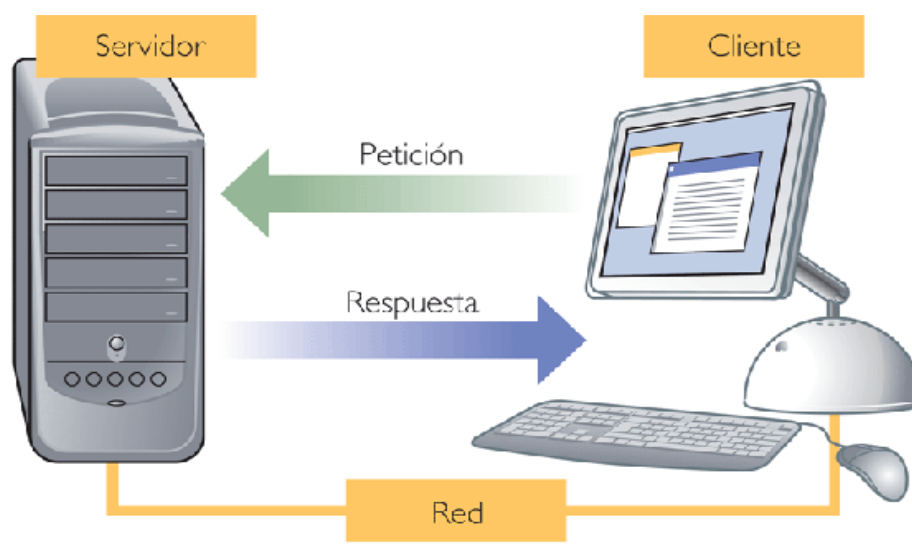


Figura 2.2: Arquitectura cliente servidor – Extraída de <https://sites.google.com/site/redesordenadoresgrupoc/home/arquitectura-cliente-servidor>

CAPÍTULO 3

DISEÑO Y MODELADO DE LA APLICACIÓN

En el desarrollo de software, existen distintas técnicas para la confección del listado de requisitos. En este capítulo, se describe los pasos más adecuados a seguir para el desarrollo de un sistema de información web.

3.1 MODELADO DE REQUERIMIENTOS

Es una herramienta para determinar las necesidades de los usuarios de la aplicación a diseñar. El objetivo principal es garantizar un sistema con automatización en todos los procesos a ejecutarse.

3.1.1 Requerimientos funcionales

- Gestión de usuarios (agregar, modificar y eliminar).
- Identificación del material electrónico mediante la lectura de código de barras.
- Gestión de préstamos y devoluciones del material electrónico mediante la lectura código de barras.
- Gestión del inventario (agregar, modificar y desechar componentes del inventario).
- Gestión de los alumnos (agregar, modificar y eliminar).
- Consulta de los recursos electrónicos en existencia.

3.1.2 Requerimientos no funcionales

- La aplicación debe ser compatible y desarrollada para un entorno web (navegador).
- La interfaz del aplicativo debe de uso intuitivo.
- Acceso a las diferentes funciones del sistema de acuerdo al rol del usuario.

3.2 MODELADO DE ACTORES

El sistema contará con dos (2) perfiles, los cuales son:

- Administrador.
- Técnico.

Para exponer a detalle las distintas actividades que realizan los dos (2) tipos de usuarios y cómo interactúan con el sistema, en la Tabla 3.1 se muestran las tareas correspondientes.

Tabla 3.1: Descripción de los actores del sistema.

Actores Procesos	Administrador	Técnico
Ingresar al sistema	Permitir a los usuarios registrados ingresar al sistema de web de inventario.	
Prestar componente	Despliega un formulario de préstamo que permite seleccionar todos los componentes electrónicos requeridos por el estudiante.	
Retornar componente	Despliega un listado para realizar la devolución de los componentes en calidad de préstamo que posea el estudiante.	
Consultar componente	Despliega un listado con todos los componentes almacenados en el inventario del taller.	
Ingresar componente	Cargar los componentes que estarán disponibles en el inventario del taller.	

Tabla 3.1: Continuación de la descripción de los actores del sistema.

Actores	Administrador	Técnico
Procesos		
Editar componente	Editar los datos de los componentes registrados en el sistema.	
Entradas al inventario	Despliega un formulario para el ingreso de nuevas unidades adquiridas mediante compra o donación.	
Salidas del inventario	Despliega un formulario para el egreso en el inventario de los componentes en mal estado o con defectos de fábrica.	
Generar código de barras	Generación de los códigos de barras correspondientes para cada componente registrado en el inventario.	
Registrar usuario	Despliega un formulario para el registro de los usuarios que tendrán acceso al sistema.	
Editar usuario	Despliega un formulario para la modificación de los datos de los usuarios.	
Eliminar usuario	Eliminar los usuarios registrados en el sistema.	
Registrar alumno	Despliega un formulario para el registro de los alumnos que harán uso del servicio.	
Editar alumno	Despliega un formulario para la modificación de los datos de los alumnos.	
Eliminar alumno	Eliminar los usuarios registrados en el sistema.	
Consultar alumno	Despliega un listado con todos los alumnos registrados en el sistema.	

3.3 CASOS DE USO

El modelado de aplicaciones en la industria del software mediante la implementación de los modelos de casos de uso, representa una herramienta simple para describir todas las interacciones o acciones que puede tener un usuario (actor) sobre un sistema o subsistema [27].

3.3.1 Descripción de los casos de uso

Tabla 3.2: Caso de uso ingresar al sistema.

Caso de uso:	Ingresar al sistema
Descripción	Esta funcionalidad permite el acceso al sistema de inventario web a los usuarios previamente registrados.
Actores	Administrador y técnico.
Pre-condiciones	No haber sido autenticado como usuario registrado.
Flujo normal	1. El usuario rellena los campos requeridos para el ingreso al sistema (correo y contraseña) y pulsa el botón de ingresar. 2. El sistema procede a validar los datos suministrados por el usuario con los datos guardados en la base de datos.
Flujo alternativo	2a. Si la información suministrada es incorrecta, el usuario no podrá ingresar.
Post-condiciones	Acceso al sistema según el perfil que posea el usuario validado.
Notas	Ninguna.

Tabla 3.3: Caso de uso prestar componente.

Caso de uso:	Prestar componente
Descripción	Esta funcionalidad permite a los usuarios realizar préstamos del material electrónico disponible en el taller.
Actores	Administrador y técnico.
Pre-condiciones	El componente se encuentra registrado y existen unidades disponibles.
Flujo normal	1. El usuario selecciona el módulo de gestión de componentes. 2. El usuario selecciona el servicio de préstamo. 3. El usuario ingresa el número de cédula del estudiante.

Tabla 3.3: Continuación del caso de uso prestar componente.

Caso de uso:	Prestar componente
Flujo normal	<p>4. El usuario ingresa el código del componente mediante el lector de código de barras o de forma manual. Cada componente cargado se refleja en la lista de resumen de la operación.</p> <p>5. Se coloca el número de unidades requeridas por cada componente registrado en la tabla de resumen.</p> <p>6. El usuario valida la información mostrada en el resumen, pulsa el botón de realizar préstamo.</p>
Flujo alternativo	<p>3a. Si el alumno no se encuentra registrado, se desplegará una alerta indicando este hecho, por lo tanto, el usuario deberá pulsar el botón de registrar alumno y completar el formulario con los datos correspondientes.</p> <p>4a. Si al momento de cargar los componentes en el formulario y alguno de ellos no se encuentra registrado o no posee unidades disponibles, se mostrarán alertas para indicar estos eventos.</p> <p>5a. Si el número de unidades ingresado es negativo, cero o es mayor al número de unidades disponibles en el inventario, se tienen alertas para indicarle al usuario la existencia de estos conflictos.</p> <p>6a. Si no se ha ingresado los datos del alumno que requiere el préstamo, se despliega una alerta notificando el error.</p>
Post-condiciones	Se realiza el préstamo requerido y lo indica a través de una alerta.
Notas	Ninguna.

Tabla 3.4: Caso de uso retornar componente.

Caso de uso:	Retornar componente
Descripción	Esta funcionalidad permite a los usuarios regresar los recursos electrónicos adquiridos por los estudiantes.
Actores	Administrador y técnico.
Pre-condiciones	Que el componente este prestado.
Flujo normal	<p>1. El usuario selecciona el módulo de gestión de componentes.</p> <p>2. El usuario selecciona el servicio de retorno.</p>

Tabla 3.4: Continuación del caso de uso retornar componente.

Caso de uso:	Retornar componente
Flujo normal	<p>3. El usuario ingresa el número de cédula del estudiante y da click al botón consultar.</p> <p>4. Se despliega una tabla con todos los componentes asociados al estudiante. En la columna U.P (unidades pendientes) se indican el número de unidades en posesión del estudiante por cada componente. Además, se despliega el botón regresar todo para devolver todos los componentes asociados al alumno en turno.</p> <p>5. Para realizar un retorno selectivo de los componentes prestados, se provee el botón agregar asociado a cada registro, este botón añadirá el componente que se desea a la tabla de resumen de retorno.</p> <p>6. El usuario mediante el campo editable de la tabla de resumen, ubicado en la columna U.R (unidades a retornar), indicará el número de unidades a retornar.</p> <p>7. Por último, presiona el botón realizar retorno para ejecutar la devolución de los componentes listados en el resumen.</p>
	<p>3a. Si el usuario ingresado no posee ningún componente en calidad de préstamo, se mostrará una alerta indicando este evento.</p> <p>6a. Si el número de unidades en la columna U.R es mayor al de la columna U.P se mostrará una alerta pidiendo corregir este error. Además, se desplegará una alerta si se coloca un valor negativo o cero en el campo U.R.</p>
Post-condiciones	Se realiza la devolución parcial o total de los componentes en estado de préstamo y lo indica a través de una alerta.
Notas	Ninguna.

Tabla 3.5: Caso de uso consultar componente.

Caso de uso:	Consultar componente
Descripción	Esta funcionalidad permite hacer consultas sobre el material electrónico disponible en el inventario.
Actores	Administrador y técnico.
Pre-condiciones	Que existan componentes registrados en el inventario.
Flujo normal	1. El usuario selecciona el módulo de gestión de componentes.

Tabla 3.5: Continuación del caso de uso consultar componente.

Caso de uso:	Consultar componente
Flujo normal	<p>2. Se despliega un listado con todos los componentes registrados en el inventario.</p> <p>3. El usuario ubica el recurso a consultar mediante un código previamente asignado, el cual generalmente es el provisto por el fabricante, para ello se provee una barra de búsqueda y el ingreso del código se puede hacer manual o con el lector de código de barras.</p> <p>4. Si la búsqueda genera resultados, se visualizará en la pantalla el recurso consultado, con todos los datos relevantes sobre el mismo.</p>
Flujo alternativo	4a. Si no hay ningún resultado que concuerde con la búsqueda hecha, el componente no se encuentra registrado en el inventario, por lo tanto, se debe proceder a registrarlo, si el usuario en sesión posee el perfil de administrador se podrá dirigir al módulo de gestión de inventario y realizar el registro del componente.
Post-condiciones	El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.
Notas	Ninguna.

Tabla 3.6: Caso de uso ingresar componente.

Caso de uso:	Ingresar componente
Descripción	Esta funcionalidad permite el registro de los componentes que se encontrarán disponibles para las personas que harán uso del servicio.
Actores	Administrador.
Pre-condiciones	El componente no ha sido registrado en el sistema.
Flujo normal	<p>1. El usuario selecciona el módulo de gestión de inventario.</p> <p>2. Se despliega la vista principal, mostrando los componentes que se encuentran almacenados en el inventario.</p> <p>3. El usuario pulsa el botón nuevo componente y se despliega un formulario en forma de ventana modal.</p> <p>4. El usuario completa todos los campos obligatorios y pulsa el botón de guardar datos.</p>

Tabla 3.6: Continuación del caso de uso ingresar componente.

Caso de uso:	Ingresar componente
Flujo normal	5. Si la validación de los datos introducidos es correcta, se añade el componente al inventario y muestra un mensaje para avisarle al usuario que la operación se ha ejecutado correctamente.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo. 4a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario y así invalidar el registro.
Post-condiciones	El usuario ha añadido el componente al inventario.
Notas	Ninguna.

Tabla 3.7: Caso de uso editar componente.

Caso de uso:	Editar componente
Descripción	Esta funcionalidad permite editar los datos de los componentes registrados en el inventario.
Actores	Administrador.
Pre-condiciones	El componente está almacenado en el sistema.
Flujo normal	1. El usuario selecciona el módulo de gestión de inventario. 2. Se despliega un listado con todos los componentes almacenados en el inventario. 3. El usuario pulsa el botón de editar componente asociado al recurso que desea modificar, luego, se despliega un formulario en forma de ventana modal con los datos almacenados previamente. 4. El usuario edita los datos que necesita y pulsa el botón de actualizar datos. 5. Si la validación es correcta, se modifican los datos almacenados en la base de datos y muestra un mensaje para avisarle al usuario que la operación se ha concretado correctamente.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo.

Tabla 3.7: Continuación del caso de uso editar componente.

Caso de uso:	Editar componente
Flujo alternativo	4a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario, invalidando la ejecución de esta acción.
Post-condiciones	El usuario ha actualizado los datos del componente existente.
Notas	Ninguna.

Tabla 3.8: Caso de uso entradas al inventario.

Caso de uso:	Entradas al inventario
Descripción	Esta funcionalidad permite cargar nuevas unidades de los componentes registrados en el inventario del taller.
Actores	Administrador.
Pre-condiciones	El componente ha sido registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de inventario. 2. El usuario pulsa el botón de movimientos de inventario. 3. Se muestra la vista principal del módulo de movimientos de inventario con el listado de las entradas y salidas existentes sobre el inventario del taller. 4. El usuario pulsa el botón entradas. 5. Se despliega el formulario para ingresar las nuevas unidades adquiridas. 6. El usuario selecciona el motivo de ingreso y coloca el código del componente, finalmente le da click al botón cargar a la lista para añadir el componente al resumen de la operación. 7. Se selecciona el número de unidades que se añadirán. 8. Click a botón de ejecutar ingreso.
Flujo alternativo	<p>5a. El usuario cancela la operación y vuelve a la vista principal del módulo.</p> <p>6a. Si el componente no se encuentra registrado, se desplegará una alerta indicando este hecho, el usuario deberá pulsar el botón de registrar componente y completar el formulario. También se mostrará una alerta si se ingresa el código del componente sin haber ingresado el motivo en primer lugar.</p>

Tabla 3.8: Continuación del caso de uso entradas al inventario.

Caso de uso:	Entradas al inventario
Flujo alternativo	7a. Si el número de unidades ingresado es negativo o cero, se tienen alertas para indicarle al usuario la existencia de estos errores.
Post-condiciones	El usuario ha ingresado nuevas unidades de los componentes del inventario.
Notas	Ninguna.

Tabla 3.9: Caso de uso salidas del inventario.

Caso de uso:	Salidas del inventario
Descripción	Esta funcionalidad permite registrar el egreso de componentes del inventario con mal funcionamiento.
Actores	Administrador.
Pre-condiciones	El componente ha sido registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de inventario. 2. El usuario pulsa el botón de movimientos de inventario. 3. Se muestra la vista principal del módulo de movimientos de inventario con el listado de las entradas y salidas existentes sobre el inventario del taller. 4. El usuario pulsa el botón salidas. 5. Se despliega el formulario para desechar las unidades correspondientes. 6. El usuario selecciona el motivo del egreso y coloca el código del componente, finalmente le da click al botón cargar a la lista para añadir el componente al resumen de la operación. 7. Se selecciona el número de unidades que se descartaran. 8. Click a botón de ejecutar egreso.
Flujo alternativo	<p>5a. El usuario cancela la operación y vuelve a la vista principal del módulo.</p> <p>6a. Si el componente no se encuentra registrado, se desplegará una alerta indicando este hecho. También se mostrará una alerta si se ingresa el código del componente sin haber ingresado el motivo en primer lugar.</p>

Tabla 3.9: Continuación del caso de uso salidas del inventario.

Caso de uso:	Salidas del inventario
Flujo alternativo	7a. Si el número de unidades ingresado es negativo o cero, se tienen alertas para indicarle al usuario la existencia de estos errores.
Post-condiciones	El usuario ha descartado del inventario las unidades en mal estado.
Notas	Ninguna.

Tabla 3.10: Generar código de barras.

Caso de uso:	Generar código de barras
Descripción	Esta funcionalidad permite generar un código de barras, el cual se asocia al código suministrado al momento del ingreso del componente.
Actores	Administrador.
Pre-condiciones	El componente ha sido registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de inventario. 2. Se muestra un listado con todos los componentes registrados en el inventario. 3. El usuario pulsa el botón de generación de código de barras asociado al recurso de interés, luego, se despliega una ventana con el código de barras correspondiente al código identificador asociado al componente. 4. Desde esta ventana el usuario puede descargar la imagen del código de barras correspondiente dando click derecho en el ratón y luego en la opción guardar imagen como.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo.
Post-condiciones	El usuario ha descargado la imagen con el código de barras del componente para hacer uso del lector.
Notas	El estándar asociado al código de barras generado en el sistema es el code-128.

Tabla 3.11: Registrar usuario.

Caso de uso:	Registrar usuario
Descripción	Esta funcionalidad permite registrar a los usuarios que tendrán acceso al sistema.
Actores	Administrador.
Pre-condiciones	El usuario no se encuentra registrado en el sistema.

Tabla 3.11: Continuación del caso de uso registrar usuario.

Caso de uso:	Registrar usuario
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de usuarios. 2. Se muestra un listado con todos los usuarios que tienen acceso al sistema. 3. El usuario pulsa el botón nuevo usuario y se despliega un formulario en forma de ventana modal. 4. El usuario completa todos los campos obligatorios y pulsa el botón de guardar datos. 5. Si la validación de los datos introducidos es correcta, se añade el usuario nuevo que tendrán acceso al sistema con el respectivo perfil de acceso asignado y se muestra un mensaje para indicar que la operación se ha concretado correctamente.
Flujo alternativo	<ol style="list-style-type: none"> 4a. El usuario cancela la operación y vuelve a la vista principal del módulo. 5a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario.
Post-condiciones	El usuario ha añadido un nuevo usuario satisfactoriamente.
Notas	Ninguna.

Tabla 3.12: Caso de uso editar usuario.

Caso de uso:	Editar usuario
Descripción	Esta funcionalidad permite editar los datos de los usuarios registrados en el sistema.
Actores	Administrador
Pre-condiciones	El usuario se encuentra registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de usuarios. 2. Se despliega un listado con todos los usuarios con acceso al sistema. 3. El usuario pulsa el botón de editar usuario asociado al usuario que desea modificar, se despliega un formulario en forma de ventana modal con los datos almacenados previamente. 4. El usuario edita los datos que necesita y pulsa el botón de actualizar datos.

Tabla 3.12: Continuación del caso de uso editar usuario.

Caso de uso:	Editar usuario
Flujo normal	5. Si la validación es correcta, se modifican los datos almacenados en la base de datos y se muestra un mensaje para indicar que la operación se ha concretado correctamente.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo. 5a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario.
Post-condiciones	El usuario ha actualizado los datos deseados.
Notas	Ninguna.

Tabla 3.13: Caso de uso eliminar usuario.

Caso de uso:	Eliminar usuario
Descripción	Esta funcionalidad permite eliminar a los usuarios registrados en el sistema.
Actores	Administrador
Pre-condiciones	El usuario se encuentra registrado en el sistema.
Flujo normal	1. El usuario pulsa el botón de borrar usuario asociado al usuario que desea descartar, se despliega una ventana de confirmación. 2. El usuario pulsa el botón de eliminar. 3. Se muestra un mensaje para indicar que la operación se ha concretado correctamente. 4. El usuario edita los datos que necesita y pulsa el botón de actualizar datos. 5. Si la validación es correcta, se modifican los datos almacenados en la base de datos y se muestra un mensaje para indicar que la operación se ha concretado correctamente.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo. 5a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario.
Post-condiciones	El usuario ha sido eliminado permanentemente.
Notas	Ninguna.

Tabla 3.14: Caso de uso registrar alumno.

Caso de uso:	Registrar alumno
Descripción	Esta funcionalidad permite registrar a los alumnos que harán uso de este servicio.
Actores	Administrador y técnico.
Pre-condiciones	El alumno no se encuentra registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. Si el usuario posee el perfil de administrador, selecciona el módulo de gestión de alumnos. 2. Se despliega un listado con todos los alumnos registrados en el sistema. 3. El usuario pulsa el botón nuevo alumno y se despliega un formulario en forma de ventana modal. 4. El usuario completa todos los campos obligatorios y pulsa el botón de guardar datos. 5. Si la validación de los datos introducidos es correcta, se añade el alumno nuevo que podrán hacer uso de este servicio, además, se muestra un mensaje para avisarle al usuario que la operación se ha concretado correctamente.
Flujo alternativo	<p>1a. Los usuarios con el perfil administrador también pueden registrar a un alumno desde el módulo de gestión de componentes. En el formulario de la solicitud de préstamo se encuentra un botón para realizar esta acción, aunque esta opción se planteó mayormente para los usuarios con el perfil de técnico.</p> <p>3a. El usuario cancela la operación y vuelve a la vista principal del módulo.</p> <p>5a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario.</p>
Post-condiciones	El usuario ha registrado satisfactoriamente al alumno.
Notas	Ninguna.

Tabla 3.15: Caso de uso editar alumno.

Caso de uso:	Editar alumno
Descripción	Esta funcionalidad permite editar los datos de los alumnos registrados en el sistema.
Actores	Administrador.
Pre-condiciones	El alumno se encuentra registrado en el sistema.

Tabla 3.15: Continuación del caso de uso editar alumno.

Caso de uso:	Editar alumno
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de alumnos. 2. Se despliega un listado con todos los alumnos con ingresados en el sistema. 3. El usuario pulsa el botón de editar alumno asociado al alumno que desea modificar, se despliega un formulario en forma de ventana modal con los datos almacenados previamente. 4. El usuario edita los datos que necesita y pulsa el botón de actualizar datos. 5. Si la validación es correcta, se modifican los datos almacenados en la base de datos y se muestra un mensaje para indicar que la operación se ha concretado correctamente.
Flujo alternativo	<ol style="list-style-type: none"> 3a. El usuario cancela la operación y vuelve a la vista principal del módulo. 5a. Si la validación de datos es incorrecta, se tienen alertas personalizadas para mostrar los errores cometidos al rellenar los campos del formulario.
Post-condiciones	El usuario ha modificado satisfactoriamente los datos del alumno.
Notas	Ninguna.

Tabla 3.16: Eliminar alumno.

Caso de uso:	Eliminar alumno
Descripción	Esta funcionalidad permite eliminar a los alumnos que ya no requieren el uso de este servicio.
Actores	Administrador.
Pre-condiciones	El alumno se encuentra registrado en el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario selecciona el módulo de gestión de alumnos. 2. Se despliega un listado con todos los alumnos almacenados en el sistema. 3. El usuario pulsa el botón de borrar alumno asociado al alumno que desea descartar, luego se despliega una ventana de confirmación. 4. El usuario pulsa el botón de eliminar.

Tabla 3.16: Continuación del caso de uso eliminar alumno.

Caso de uso:	Eliminar alumno
Flujo normal	5. Se muestra un mensaje para avisarle al usuario que la operación se ha concretado correctamente.
Flujo alternativo	3a. El usuario cancela la operación y vuelve a la vista principal del módulo.
Post-condiciones	El usuario ha eliminado permanentemente al alumno seleccionado.
Notas	Ninguna.

Tabla 3.17: Consultar alumno.

Caso de uso:	Consultar alumno
Descripción	Esta funcionalidad permite hacer consultas sobre el material electrónico disponible en el inventario.
Actores	Administrador
Pre-condiciones	El alumno se encuentra registrado en el sistema.
Flujo normal	<p>1. El usuario selecciona el módulo de gestión de alumnos.</p> <p>2. Se despliega un listado con todos los alumnos registrados en el sistema.</p> <p>3. El usuario ubica al alumno mediante su número de cedula, para ello se provee una barra de búsqueda.</p> <p>4. Si la búsqueda genera resultados, se visualizará en la pantalla al alumno con todos los datos relevantes asociados.</p>
Flujo alternativo	4a. Si no hay ningún resultado que concuerde con la búsqueda hecha, el alumno no se encuentra en la base de datos del sistema, por lo tanto, se debe proceder a registrarlo.
Post-condiciones	El usuario ha visualizado los resultados de su búsqueda y puede realizar otra si lo desea.
Notas	Ninguna.

3.3.2 Diagramas de casos de uso

Son una representación gráfica encargada de mostrar la interacción existente entre el sistema y los usuarios con acceso a él, desde la figura 3.1 hasta la figura 3.4 se ilustran todos los casos de uso del sistema de inventario.

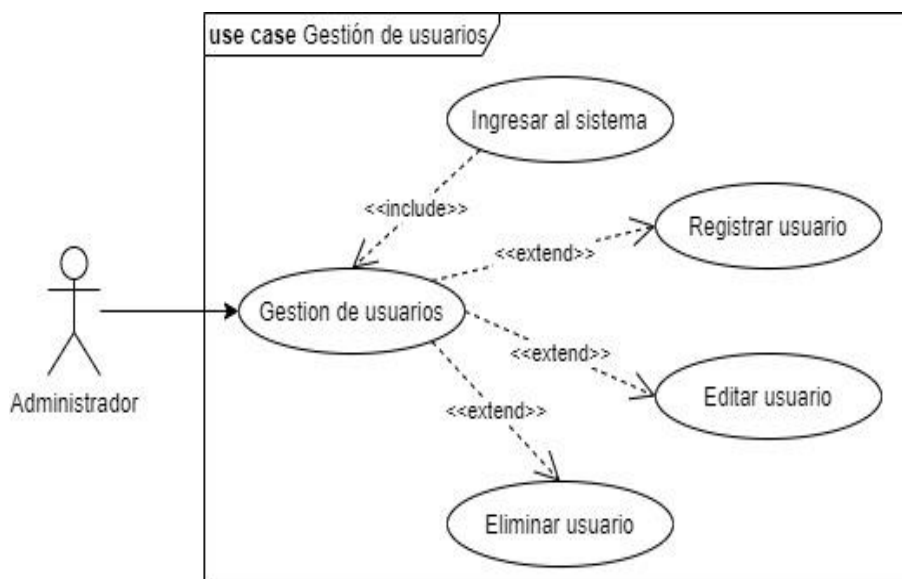


Figura 3.1: Diagrama caso de uso gestión de usuarios – Elaboración propia.

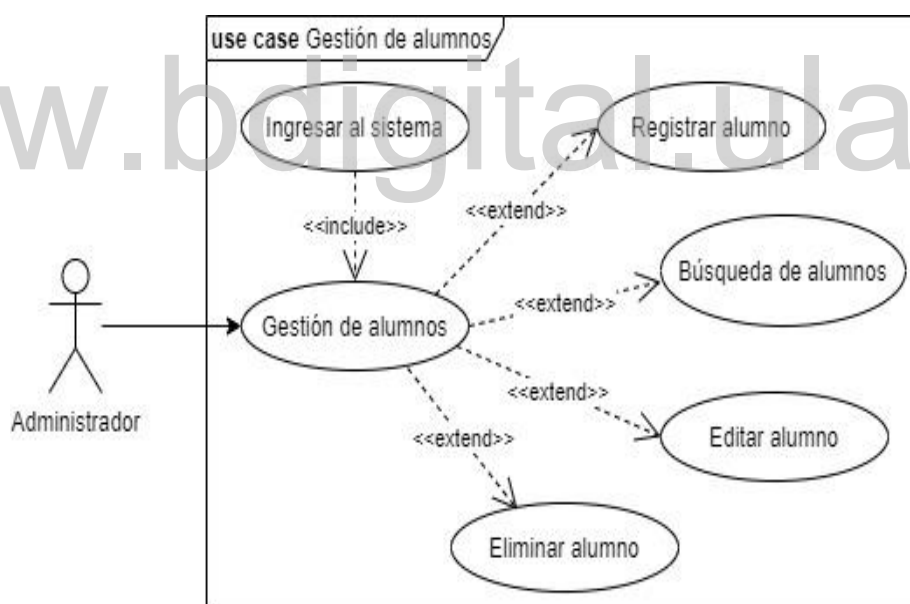


Figura 3.2: Diagrama caso de uso gestión de alumnos – Elaboración propia.

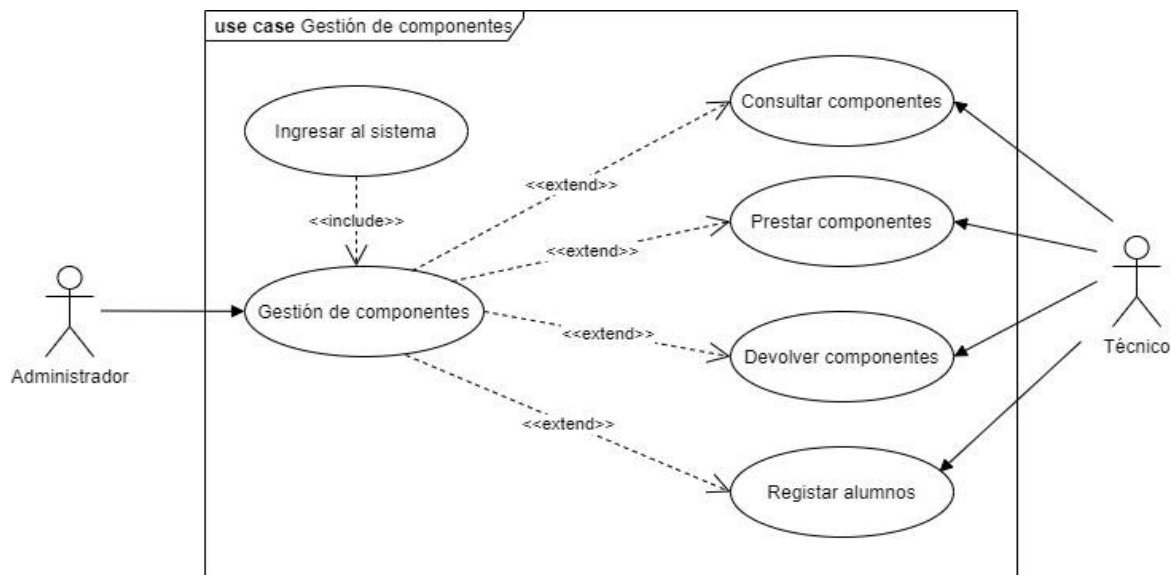


Figura 3.3: Diagrama caso de uso gestión de componentes – Elaboración propia.

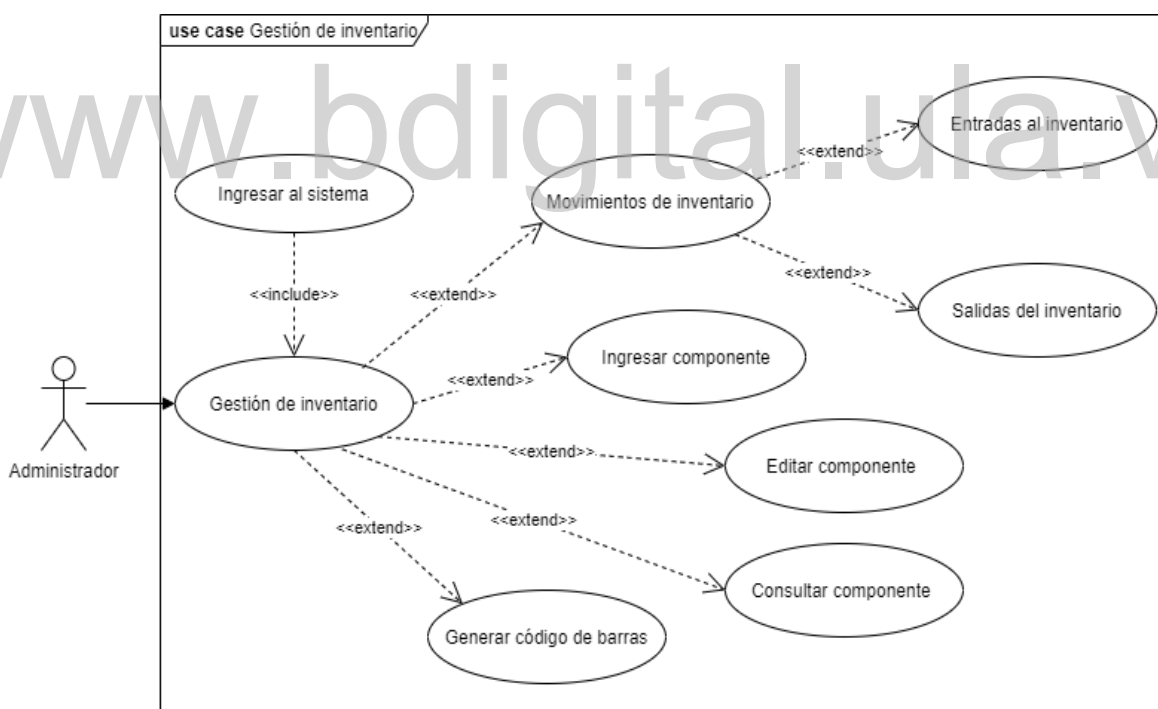


Figura 3.4: Diagrama caso de uso gestión de inventario – Elaboración propia.

3.4 DISEÑO DEL SOFTWARE

El propósito principal del diseño de software consta de determinar las interfaces de los usuarios, de la base de datos, la arquitectura y los elementos que componen el sistema, es decir, todo el modelado de la aplicación.

3.4.1 Diagrama de navegabilidad

El sistema se encuentra constituido de una serie de módulos que permite de acuerdo al rol que posea el usuario en el S.I.T.E.U.L.A, realizar funciones básicas de crear, editar, eliminar, consultar y funciones especiales como prestar y devolver.

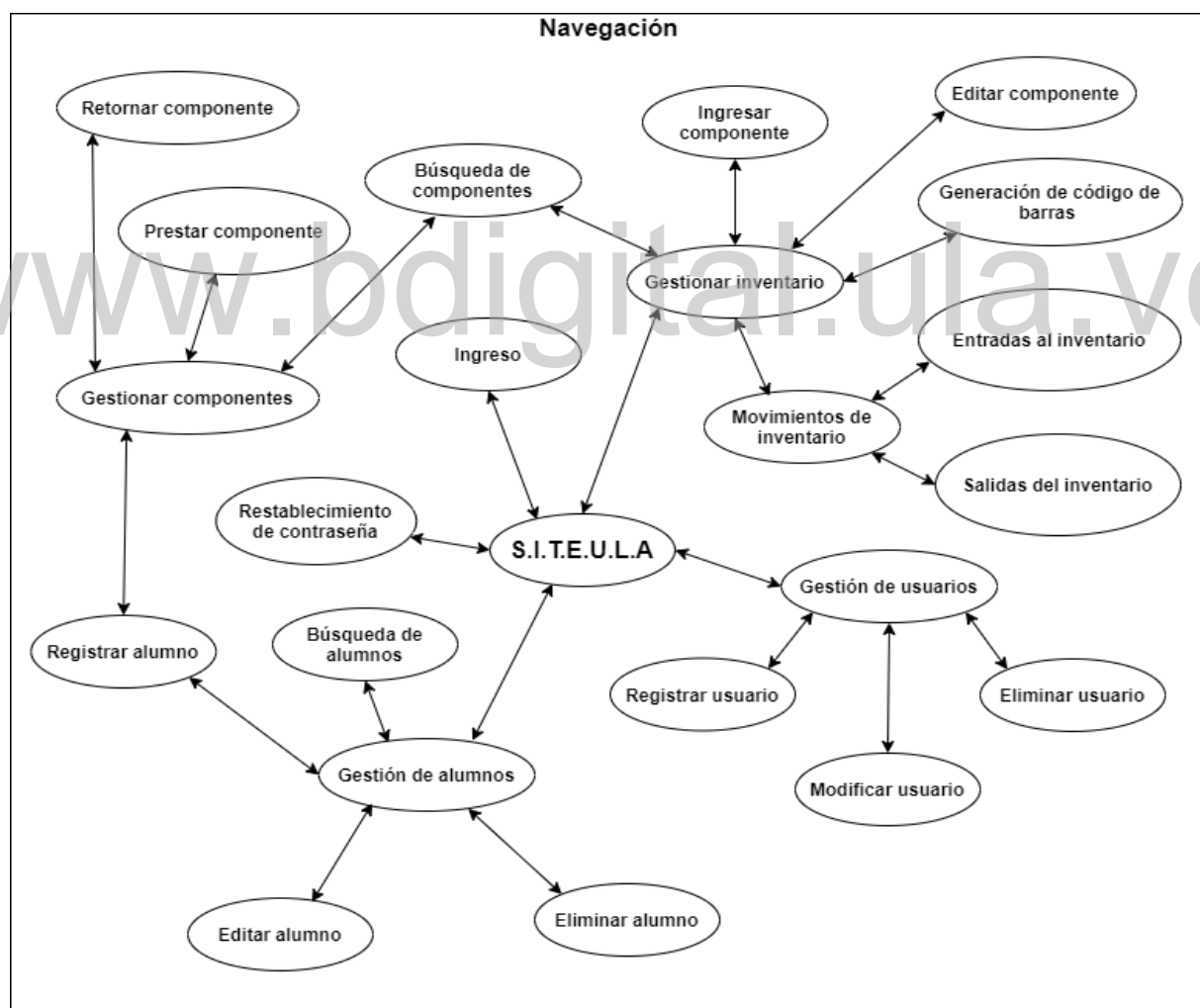


Figura 3.5: Diagrama de navegación del sistema – Elaboración propia.

3.4.2 Diseño arquitectónico del sistema

En la figura 3-6, se ilustra el esquema bajo el cual se conectará el cliente (usuario) con el servidor. Cabe destacar que el sistema de inventario web se alojará de forma local en un computador del taller de electrónica.

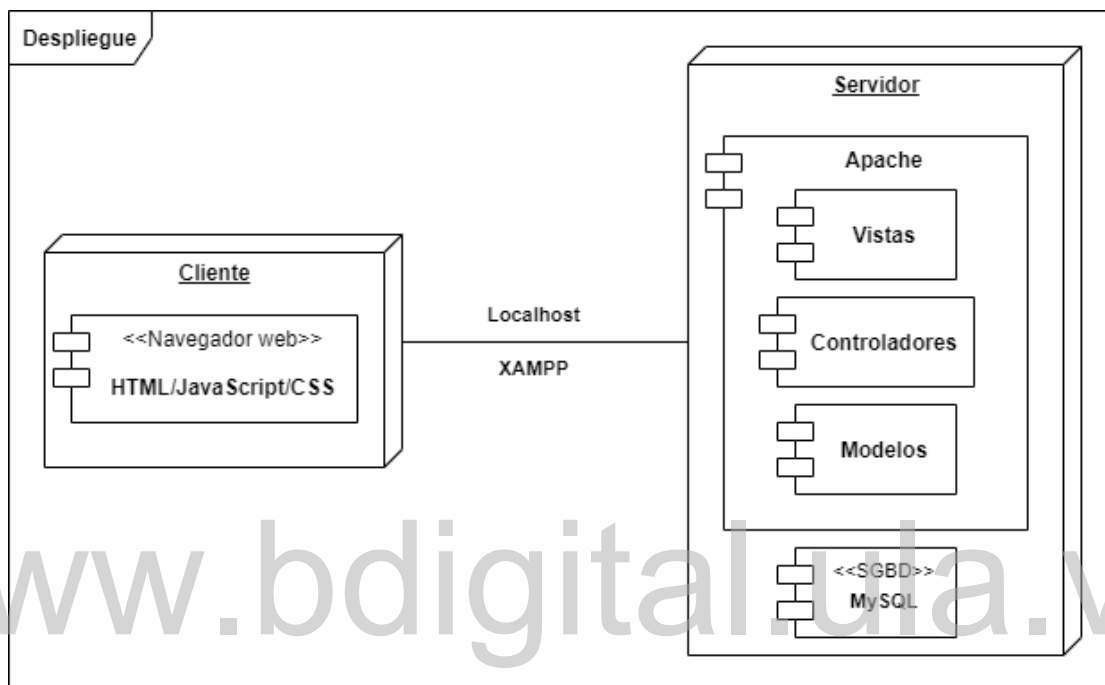


Figura 3.6: Diagrama del despliegue del sistema – Elaboración propia.

3.4.3 Descripción de componentes

a) Vistas

- *Auth*: carpeta que contiene el formulario de acceso al sistema (*login.blade.php*), el formulario de restablecimiento de contraseña (*resetpassword.blade.php*) y el modal de confirmación de restablecimiento de contraseña (*confirmreset.blade.php*).
- *Almacén*: carpeta que contiene la vista principal del módulo de gestión de inventario (*index.blade.php*), el formulario de registro de los componentes electrónicos (*store.blade.php*), el formulario de edición de los datos de los componentes registrados (*edit.blade.php*), la ventana que despliega el código de barras de los componentes

registrados (*barcode.blade.php*) y la barra de búsqueda incrustada en la vista principal del módulo de gestión de inventario (*search.blade.php*).

- Alumnos: carpeta que contiene la vista principal del módulo de gestión de alumnos (*index.blade.php*), el formulario de registro de alumnos (*store.blade.php*), el formulario de edición de datos de los alumnos registrados (*edit.blade.php*), la ventana de confirmación de eliminar alumnos (*delete.blade.php*) y la barra de búsqueda incrustada en la vista principal del módulo de gestión de alumnos (*search.blade.php*).
- Errors: carpeta que contiene las vistas de los errores de acceso no autorizado (*403.blade.php*), página no encontrada (*404.blade.php*). Además, contiene la plantilla que se utiliza para la construcción de estas vistas (*illustrated-layout.blade.php*).
- Usuarios: carpeta que contiene la vista principal del módulo de gestión de usuarios (*index.blade.php*), el formulario de registro de usuarios (*store.blade.php*), el formulario de edición de datos de los usuarios registrados (*edit.blade.php*), la ventana de la alerta de eliminar usuarios (*delete.blade.php*).
- Gestión: carpeta que contiene el formulario de préstamo de componentes (*préstamo.blade.php*), el formulario de retorno de componentes (*retorno.blade.php*) y un formulario de registro de alumnos (*storestudent.blade.php*).
- Layouts: carpeta que contiene la plantilla empleada en el *login* de acceso (*applogin.blade.php*).
- Recursos: carpeta que contiene la vista principal del módulo de gestión de componentes (*index.blade.php*), la vista del módulo de consulta de componentes (*consulta.blade.php*) y la barra de búsqueda de componentes (*search.blade.php*).
- Theme: carpeta que contiene el archivo de la cabecera (*header.blade.php*), la barra lateral (*aside.blade.php*), el archivo de pie de página (*footer.blade.php*) y el archivo con que maqueta toda la estructura de la plantilla de la aplicación (*layout.blade.php*).

- *Homepages*: carpeta que contiene la vista de inicio de la aplicación (*inicio.blade.php*).
- *Movinventario*: carpeta que contiene la vista principal del módulo de movimientos de inventario (*index.blade.php*), el formulario de entradas al inventario (*entradasblade.php*), el formulario de salidas del inventario (*salidas.blade.php*) y un formulario de registro de componentes (*storecomponent.blade.php*).

b) Controladores

- *Inicio*: controlador asociado a la vista de inicio de la aplicación (*InicioController.php*).
- *Auth/Login*: controlador encargado de procesar y validar la solicitud de acceso a la aplicación (*LoginController.php*).
- *Barcode*: controlador diseñado para la generación del código de barras de los componentes electrónicos (*BarcodeController.php*).
- *Gestión*: controlador asociado a la vista de inicio del módulo de gestión de componentes (*GestionController.php*).
- *Consulta*: controlador encargado de procesar las solicitudes de consulta de los componentes registrados en el sistema (*ConsultaController.php*).
- *Alumno*: controlador empleado para el procesamiento de todas las solicitudes de crear, editar, eliminar y consultar alumnos (*AlumnoController.php*).
- *Usuario*: controlador empleado para el procesamiento de todas las solicitudes de crear, editar y eliminar usuarios (*UsuarioController.php*).
- *Movimiento*: controlador encargado de manejar las solicitudes provenientes desde la sección de movimientos de inventario ubicada en el módulo de gestión de inventario (*MovimientoController.php*).

- Componente: controlador provisto con el objetivo de almacenar los componentes que se encuentran en el inventario existente, editarlos luego de registrarlos y consultarlos (*ComponenteController.php*)
- Préstamo: controlador diseñado para procesar todas las solicitudes de los préstamos estudiantiles (*PrestamoController.php*).
- Retorno: controlador diseñado para procesar todas las solicitudes de las devoluciones del material electrónico (*RetornoController.php*).
- Restablecer contraseña: controlador diseñado para la solicitud de restablecimiento de contraseña (*RestablecerContraseñaController.php*).

c) Modelos

- *User*: modelo asociado a la tabla de la base de datos denominada *users*.
- Alumno: modelo asociado a la tabla de la base de datos denominada alumnos.
- Préstamo: modelo asociado a la tabla de la base de datos denominada préstamos.
- Componente: modelo asociado a la tabla de la base de datos denominada componentes.
- Movimiento: modelo asociado a la tabla de la base de datos denominada movimientos.
- ComponenteMovimiento: modelo asociado a la tabla de la base de datos denominada Componente_Movimiento.
- ComponentePréstamo: modelo asociado a la tabla de la base de datos denominada Componente_Préstamo.

d) Public

La carpeta *public* es la única carpeta visible de la aplicación dentro de nuestro servidor.

- *css*: carpeta que contiene los recursos CSS de la aplicación.
- *js*: carpeta contenedora de los recursos JavaScript de la aplicación.
- *img*: carpeta que aloja todas las imágenes de la aplicación.
- *assets*: carpeta contenedora de las fuentes y la plantilla de la aplicación.

En la figura 3-7, se observan todos los componentes empleados en el *framework* Laravel para la implementación del sistema de inventario web.

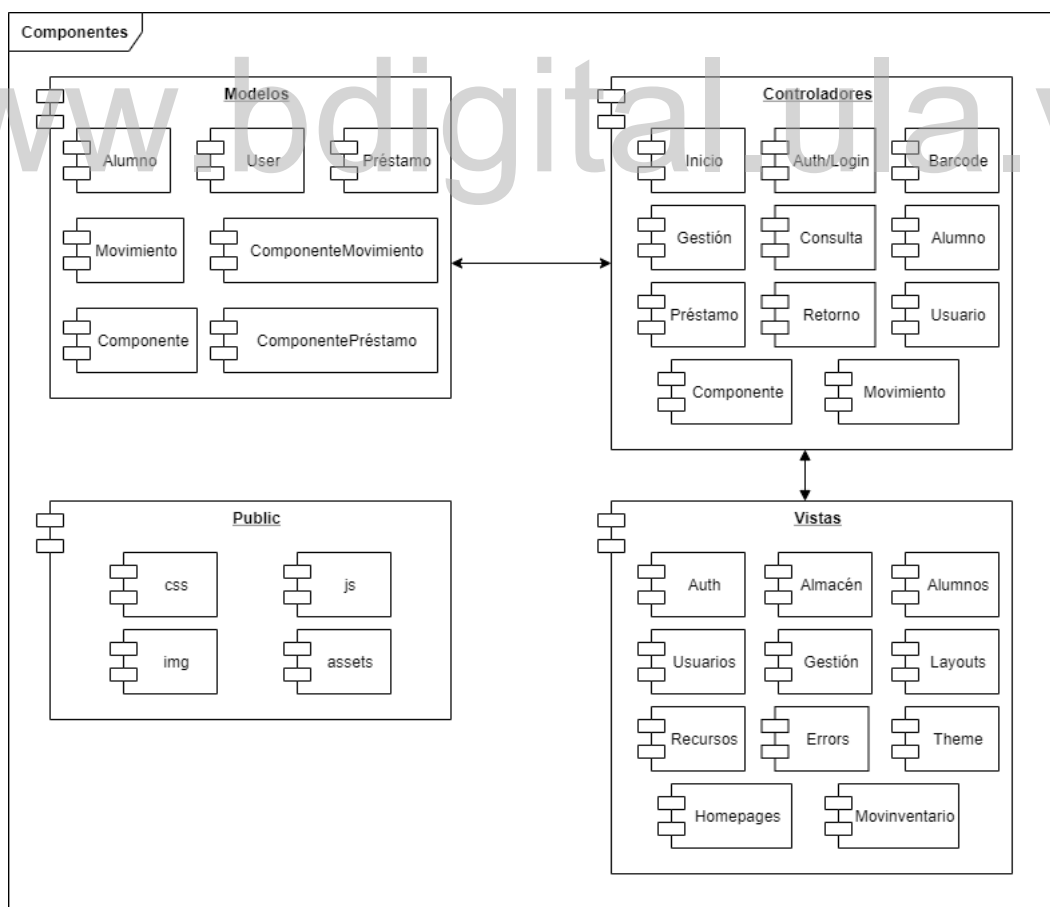


Figura 3.7: Diagrama de los componentes que constituyen el sistema – Elaboración propia.

3.5 MODELADO DE LA BASE DE DATOS

En la figura 3-8, se ilustra el modelo entidad-relación donde se muestran todas las tablas diseñadas para el sistema de inventario y las relaciones existentes entre ellas. Cabe destacar que el *framework* Laravel crea una tabla adicional que guarda un registro de las migraciones creadas, denominada *migrations*.

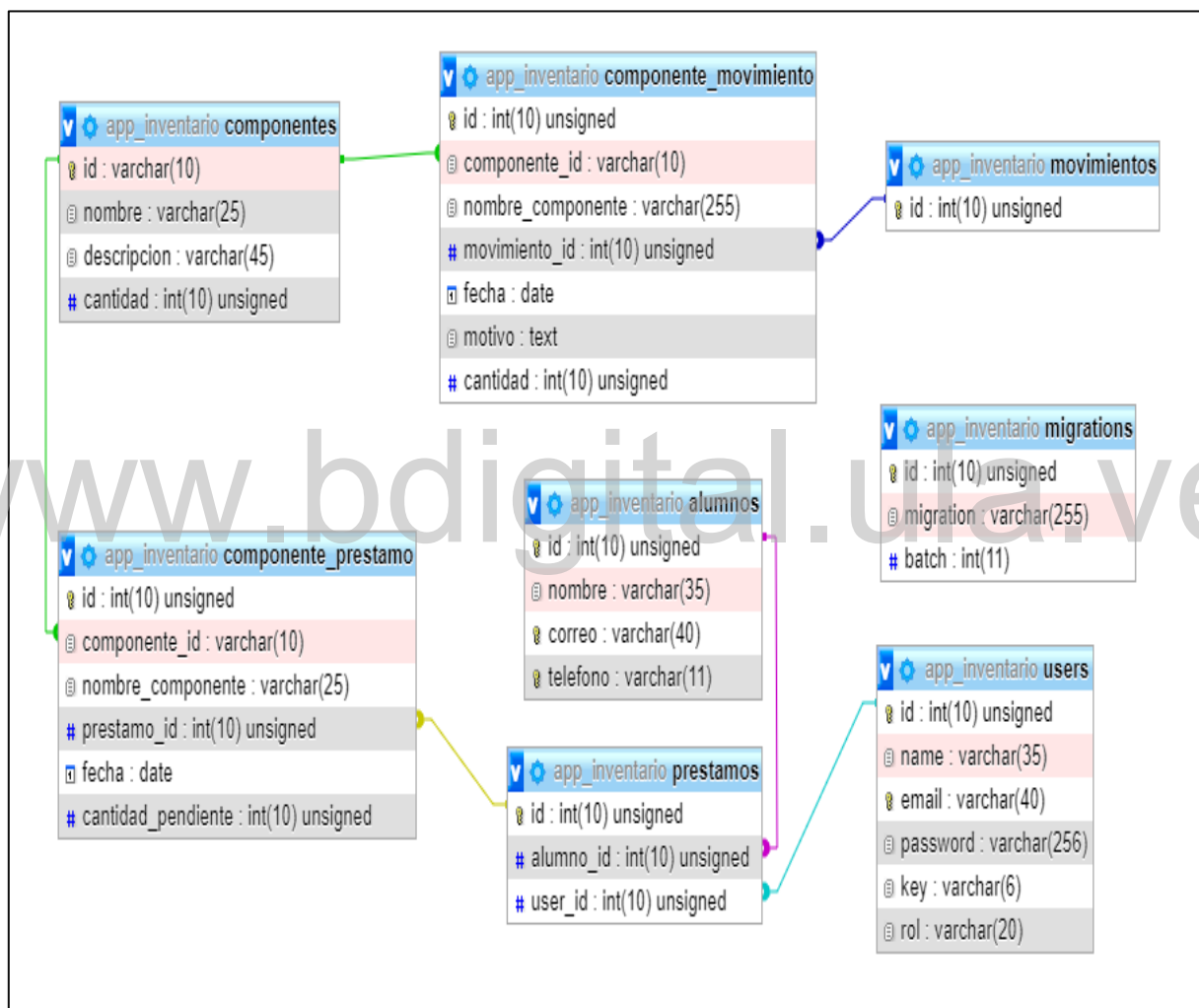


Figura 3.8: Modelo entidad-relación de la base de datos del sistema – Elaboración propia.

CAPÍTULO 4

IMPLEMENTACIÓN DEL SISTEMA

Este capítulo muestra cómo se construyó S.I.T.E.U.L.A. El aplicativo posee una Base de Datos MySQL que hace uso del lenguaje estructurado SQL. También se utilizó otros lenguajes como PHP, HTML, CSS, JavaScript.

4.1 HERRAMIENTAS DE DESARROLLO

Antes de iniciar con el desarrollo del aplicativo se configuraron las siguientes herramientas:

- XAMPP versión 7.2.30.
- Composer versión 2.0.7.
- Navegador web Google Chrome version 86.0.4240.198.
- Laravel versión 5.8.
- Como editor de código se usó Visual Studio Code versión 1.51.1 junto con los complementos requeridos para optimizar su funcionamiento.
- Git versión 2.27.0.

4.2 CONTROL DE VERSIONES

Un sistema de control de versiones es una herramienta que proporciona una gestión total de los cambios que puede sufrir un sistema de archivos o repositorio a lo largo del tiempo. El control de versiones protege el código fuente tanto de las catástrofes como del deterioro casual de los errores humanos y las consecuencias accidentales, por lo tanto, su uso se hace imprescindible en el trabajo colaborativo [28]. Para el desarrollo de este sistema web se utilizó el sistema de control de versiones Git, alojando la aplicación en el servicio web Bitbucket (nube), con el objetivo de mantener un registro de todas las operaciones sobre el proyecto en Laravel (configuraciones realizadas, creación, modificación y eliminación de componentes, corrección de errores y adaptación de librerías).

4.3 CREACIÓN DEL PROYECTO Y CONSTRUCCIÓN DE LA INTERFAZ

El proyecto se creó en Laravel a través del gestor de dependencias *Composer*, este *framework* permite tener componentes enlazados bajo el Modelo Vista Controlador, adicionalmente tiene su propio sistema de plantillas denominado *Blade*. La gran ventaja de construir las vistas bajo este concepto radica en la capacidad de tener un archivo PHP con código HTML, CSS y JavaScript incrustado, inclusive estructuras de decisión y estructuras iterativas.

Para el diseño de la interfaz se empleó la plantilla de administración de código abierto AdminLTE versión 2.4.5, esta plantilla basa su diseño en una arquitectura modular para la construcción de soluciones personalizadas. Además, se usaron librerías para la generación de los códigos de barras de los componentes (*JsBarcode*) y para la generación de las alertas que interactúan con el usuario (*sweetalerts*).

- Vistas



Figura 4.1: Login de acceso al aplicativo.

En la siguiente figura, se puede apreciar la estructura que tiene la aplicación web en todas sus vistas. El número 1, denota la cabecera, el número 2 denota la barra lateral, el número 3 el pie de página y el número 4 la sección con el contenido principal.



Figura 4.2: Vista principal del aplicativo.

Administración de usuarios + Nuevo usuario					
Id	Nombre	Correo	Clave especial	Rol	Acciones
20851317	Kevin Bottini	kbottini39@gmail.com	202020	Administrador	
20888050	Manuel Mendez	manuelmven@gmail.com	222222	Tecnico	

Figura 4.3: Modulo de gestión de usuarios.

Administración de alumnos + Nuevo alumno				
Cédula: <input type="text" value="Ingrese la cédula del alumno que desea buscar"/> <input type="button" value="Buscar"/>				
Id	Nombre	Correo	Teléfono	Acciones
5858642	Reinaldo Antonio Bottini Perez	rbotinp@gmail.com	04145897984	
19895192	Evely Paredes Perez	paredevely@gmail.com	04160776288	
20847017	Freya Cristina Gonzalez Maza	freyagmaza@gmail.com	04124898699	
20888050	Manuel Alejandro Mendez Albarran	manuelmven@gmail.com	04121715269	

< 1 2 >

Figura 4.4: Modulo de gestión de alumnos.

☰ Listado de componentes electrónicos existentes + Nuevo componente + Movimientos de inventario

Código:

Código	Nombre	Descripción	Stock	Acciones
7447	Decodificador	BCD a 7 segmentos con catodo comun	0	<input type="button" value="✎"/> <input type="button" value="📄"/>
74ls00	Compuerta NAND	Encapsulado de 4 compuertas de entradas	71	<input type="button" value="✎"/> <input type="button" value="📄"/>
74ls04	Compuerta not	Encapsulado de 6 compuertas	94	<input type="button" value="✎"/> <input type="button" value="📄"/>
74ls08	Compuerta and	Encapsulado de 4 compuertas de 2 entradas	67	<input type="button" value="✎"/> <input type="button" value="📄"/>

< 1 2 >

Figura 4.5: Modulo de gestión de inventario.



Figura 4.6: Generador de código de barras.

Registro de movimientos de inventario

Entradas Salidas

Id	Código componente	Nombre componente	Fecha	Motivo	Cantidad
1	74ls32	Compuerta or	2020-11-28	Ingreso por compra	95
2	74ls00	Compuerta nand	2020-11-28	Egreso por defecto de fábrica	1
3	pic18f4550	Microcontrolador pic	2020-11-28	Egreso por componente quemado	1
4	74ls08	Compuerta and	2020-11-28	Egreso por componente partido	1

< 1 2 >

Figura 4.7: Modulo de movimientos de inventario.

Solicitud

Motivo del ingreso:

-- Seleccionar --

Código del componente:

Acción: Cargar a la lista

Resumen

Ejecutar ingreso

Código	Nombre	Motivo	Unidades	Acciones
74ls00	Compuerta NAND	Ingreso por compra	1	Eliminar
74ls32	Compuerta or	Ingreso por donacion	1	Eliminar

Figura 4.8: Modulo de entradas al inventario.

Solicitud

Motivo del egreso:

Código del componente:


Acción:

Resumen

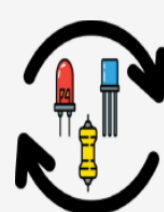
Código	Nombre	Motivo	Unidades	Acciones
74ls00	Compuerta NAND	Egreso por defecto de fábrica	1	<input type="button" value="Eliminar"/>
74ls04	Compuerta not	Egreso por componente quemado	1	<input type="button" value="Eliminar"/>
pic18f4550	Microcontrolador pic	Egreso por componente partido	1	<input type="button" value="Eliminar"/>

Figura 4.9: Modulo de salidas del inventario.


www.bdigital.ula.ve



Consulta de componentes
Listado de componentes electrónicos en existencia



Préstamo de componentes
Servicio de préstamo de componentes electrónicos



Retorno de componentes
Servicio de retorno de componentes electrónicos

Figura 4.10: Modulo de gestión de componentes.

Consulta de componentes

Código:

Código	Nombre	Descripción	Stock
7447	Decodificador	BCD a 7 segmentos con catodo comun	0
74ls00	Compuerta NAND	Encapsulado de 4 compuertas de entradas	71
74ls04	Compuerta not	Encapsulado de 6 compuertas	94
74ls08	Compuerta and	Encapsulado de 4 compuertas de 2 entradas	67

< 1 2 >

Figura 4.11: Modulo de consulta de componentes.

Solicitud

Cédula: Nombre del alumno:

Código del componente: Acción:

Resumen

Código	Nombre	Unidades	Acciones
74ls08	Compuerta and	1	<input type="button" value="Eliminar"/>
pic18f4550	Microcontrolador pic	1	<input type="button" value="Eliminar"/>
74ls32	Compuerta or	1	<input type="button" value="Eliminar"/>

Figura 4.12: Modulo de préstamo de componentes.

The screenshot displays two panels in a web application interface:

- Solicitud Panel:** Contains a search bar with 'Cédula:' (value: 20888050) and 'Nombre del alumno:' (value: Manuel Alejandro Mendez Albarran). Below is a table with columns: Id, Código, Nombre, Fecha(A/M/D), U.P, and Acción.

Id	Código	Nombre	Fecha(A/M/D)	U.P	Acción
3	74ls00	Compuerta nand	2020-11-28	1	Agregar
4	pic18f4550	Microcontrolador pic	2020-11-28	1	Agregar
- Resumen de retorno Panel:** Contains a table with columns: Id, Código, Nombre, U.R, and Acción.

Id	Código	Nombre	U.R	Acción
4	pic18f4550	Microcontrolador pic	1	Eliminar

Figura 4.13: Módulo de retorno de componentes.

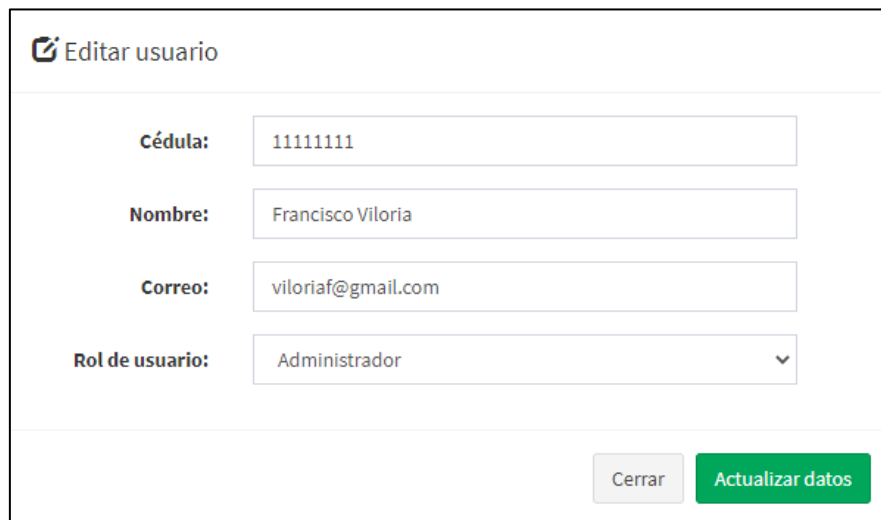
- Formularios

The screenshot shows a modal titled '+ Agregar nuevo usuario' with the following fields:

- Cédula:** Ingrese la cedula (max.8 digitos)
- Nombre:** Ingrese un nombre y un apellido (max. 35 caracteres)
- Correo:** Ingrese un correo electrónico (max. 40 caracteres)
- Contraseña:** Ingrese una contraseña de 6 digitos
- Clave especial:** Ingrese una clave de 6 digitos
- Rol:** -- Seleccionar rol --

Buttons at the bottom: Cerrar, Limpiar campos, Guardar datos.

Figura 4.14: Modal de registro de usuario.



Editar usuario

Cédula:

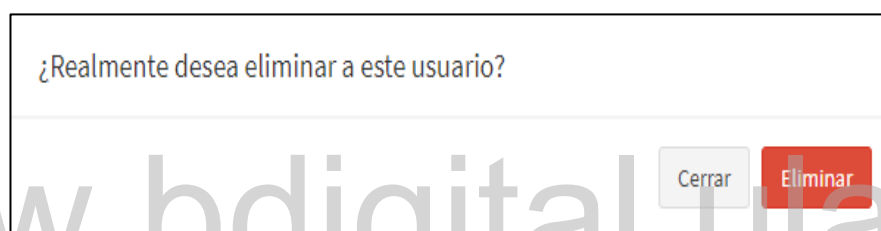
Nombre:

Correo:

Rol de usuario:

Cerrar Actualizar datos

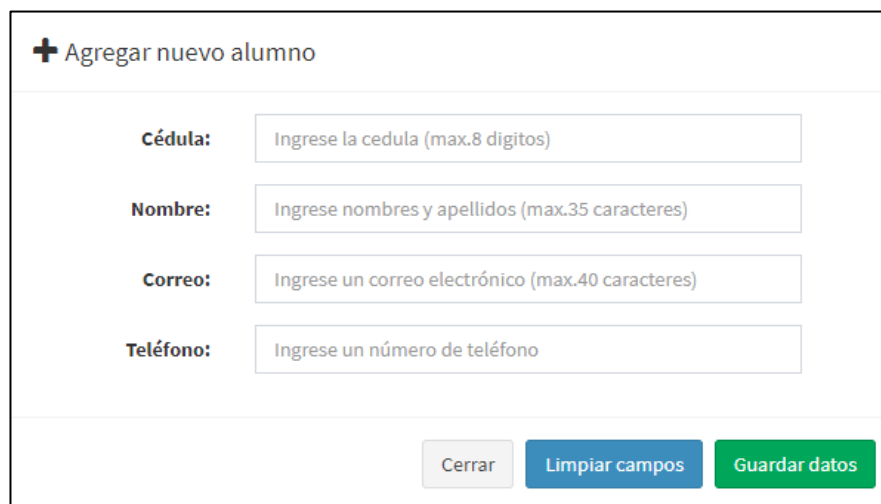
Figura 4.15: Modal de edición usuario.



¿Realmente desea eliminar a este usuario?

Cerrar Eliminar

Figura 4.16: Ventana de confirmación de eliminación de usuario.



+ Agregar nuevo alumno

Cédula:

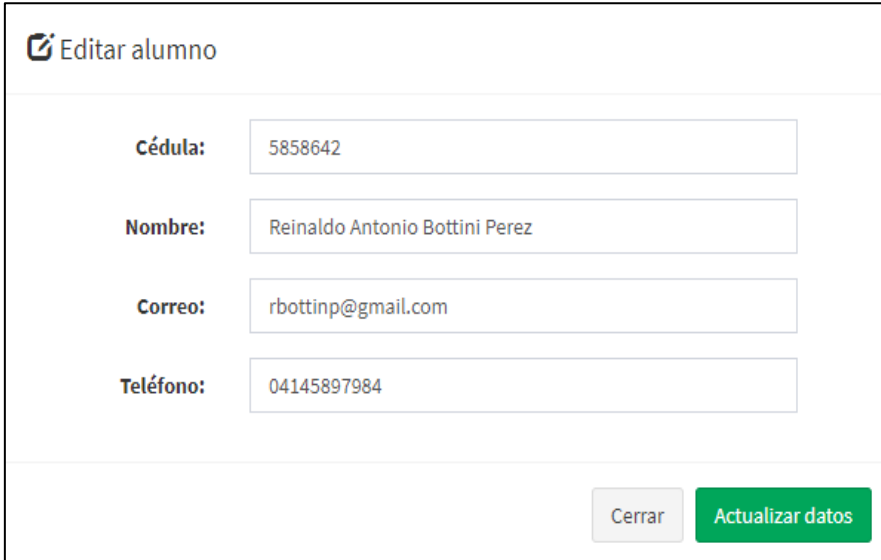
Nombre:

Correo:

Teléfono:

Cerrar Limpiar campos Guardar datos

Figura 4.17: Modal de registro de alumno.



Editar alumno

Cédula: 5858642

Nombre: Reinaldo Antonio Bottini Perez

Correo: rbottinp@gmail.com

Teléfono: 04145897984

Cerrar Actualizar datos

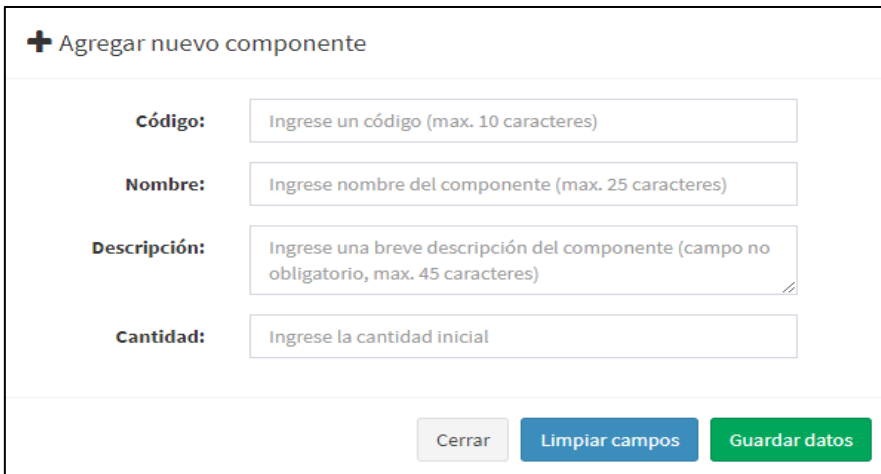
Figura 4.18: Modal de edición alumno.



¿Realmente desea eliminar a este alumno?

Cerrar Eliminar

Figura 4.19: Ventana de confirmación de eliminación de alumno.



+ Agregar nuevo componente

Código: Ingrese un código (max. 10 caracteres)

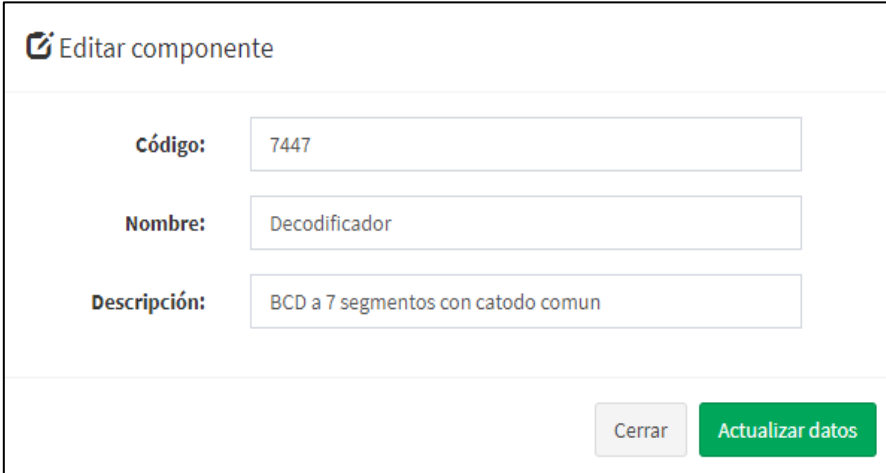
Nombre: Ingrese nombre del componente (max. 25 caracteres)

Descripción: Ingrese una breve descripción del componente (campo no obligatorio, max. 45 caracteres)

Cantidad: Ingrese la cantidad inicial

Cerrar Limpiar campos Guardar datos

Figura 4.20: Modal de registro de componente.



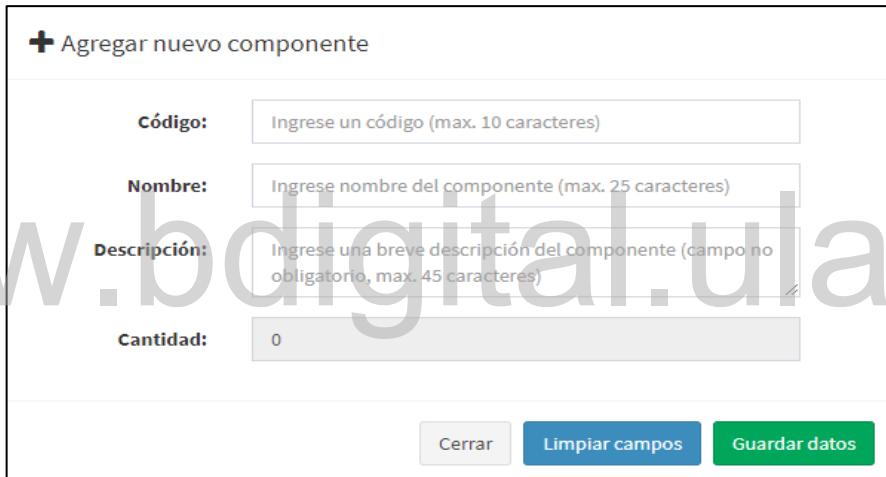
Editar componente

Código:

Nombre:

Descripción:

Figura 4.21: Modal de edición de componente.



+ Agregar nuevo componente

Código:

Nombre:

Descripción:

Cantidad:

Figura 4.22: Modal de registro de componente desde entradas al inventario.

4.4 INTERACCIÓN CLIENTE SERVIDOR

En las siguientes figuras, se describirán los procedimientos más complejos que se ejecutan en cada petición realizada por el usuario en sesión de acuerdo al rol que posee en el sistema de inventario.

- **Administrador**

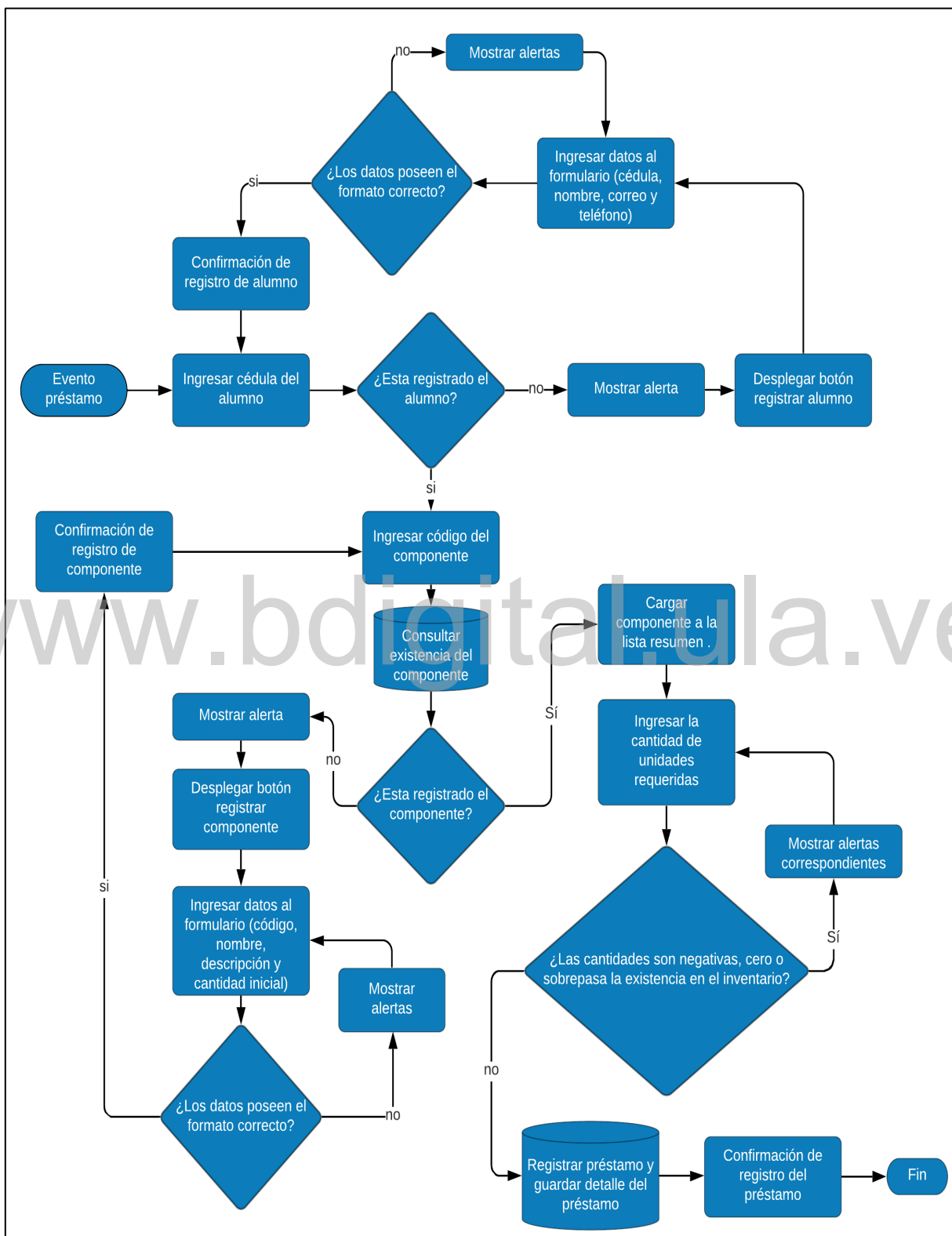


Figura 4.23: Diagrama realizar préstamo – Elaboración propia.

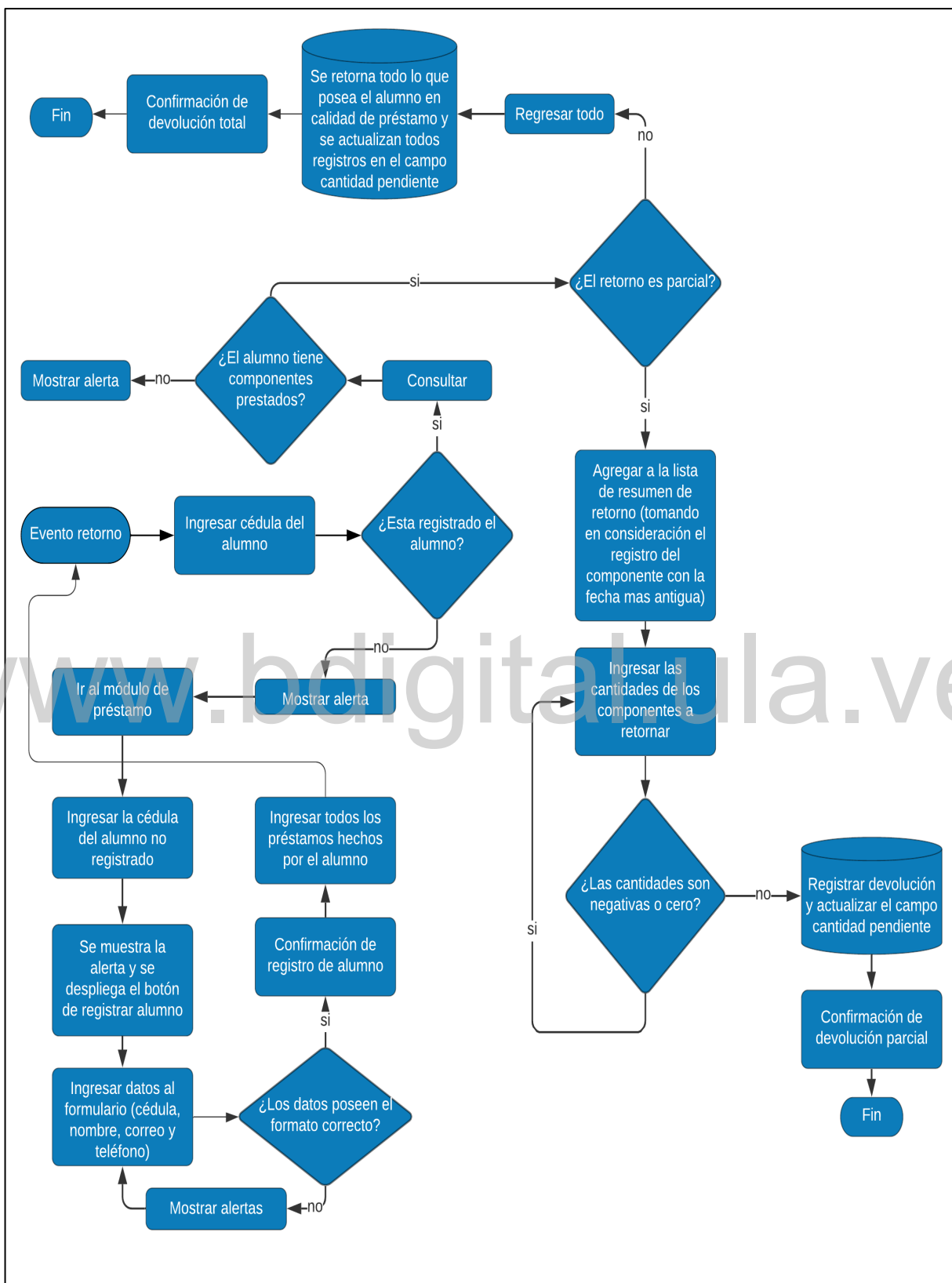


Figura 4.24: Diagrama realizar devolución – Elaboración propia.

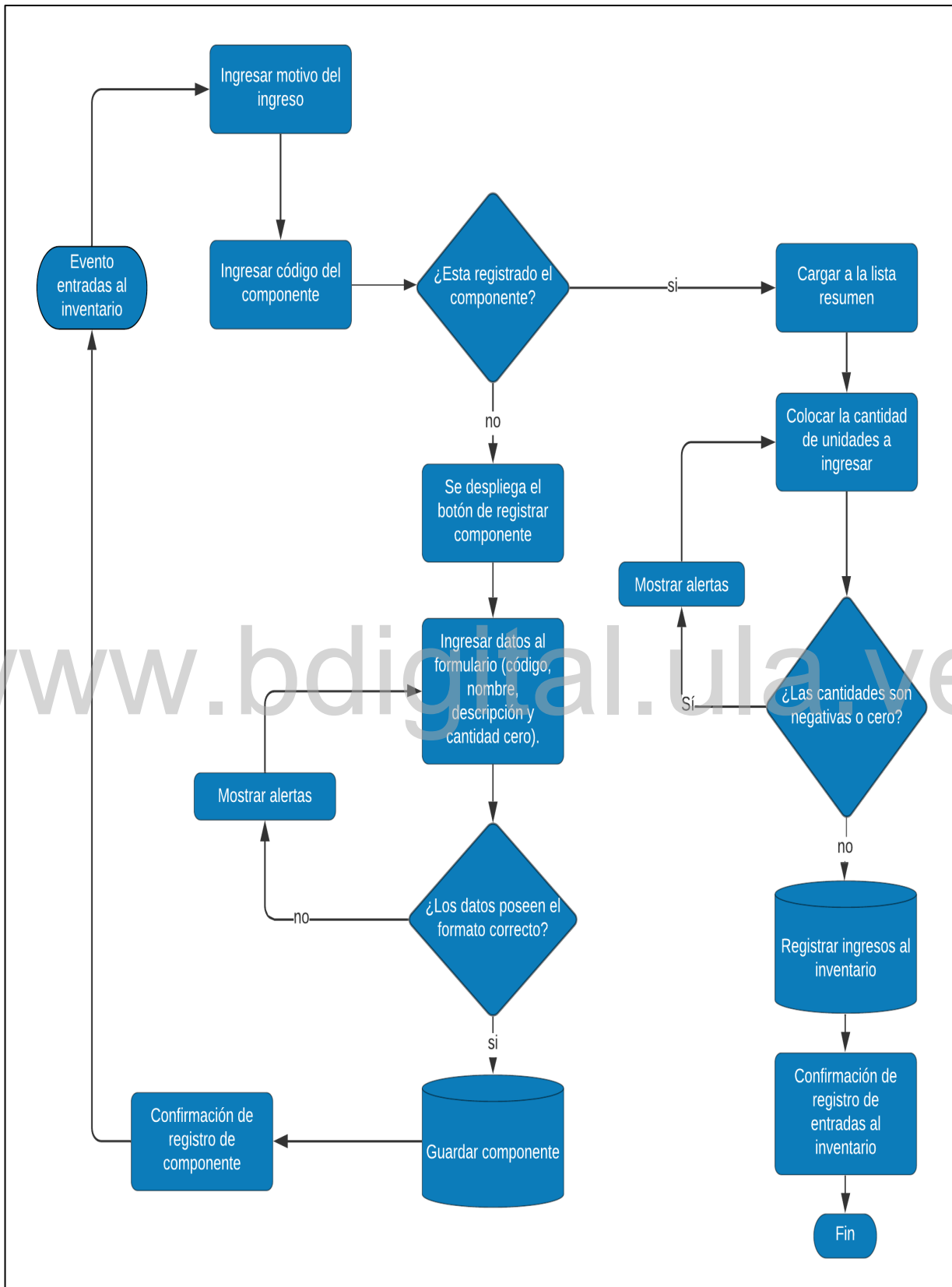


Figura 4.25: Diagrama entradas al inventario – Elaboración propia.

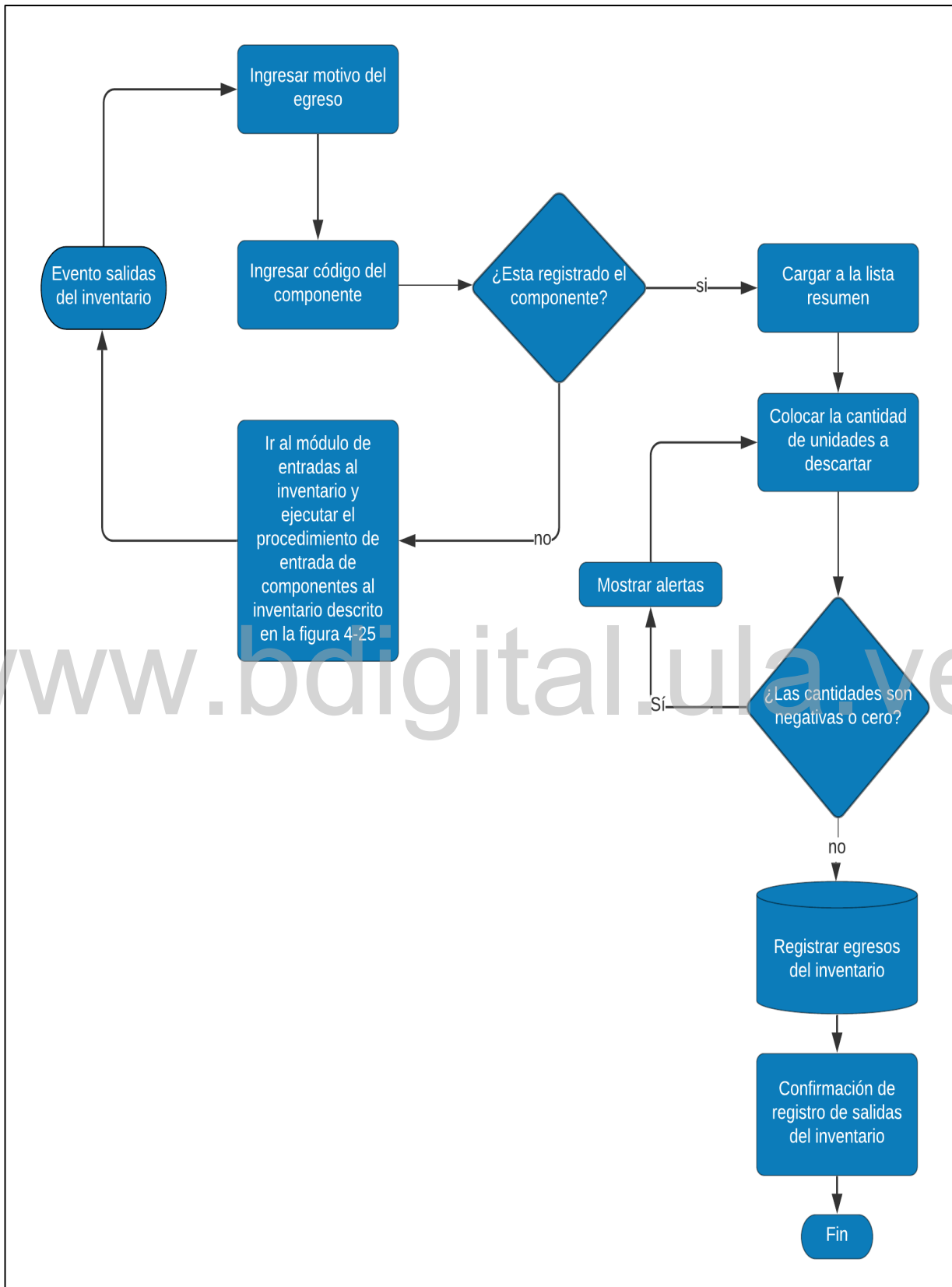


Figura 4.26: Diagrama salidas del inventario – Elaboración propia.

- Técnico

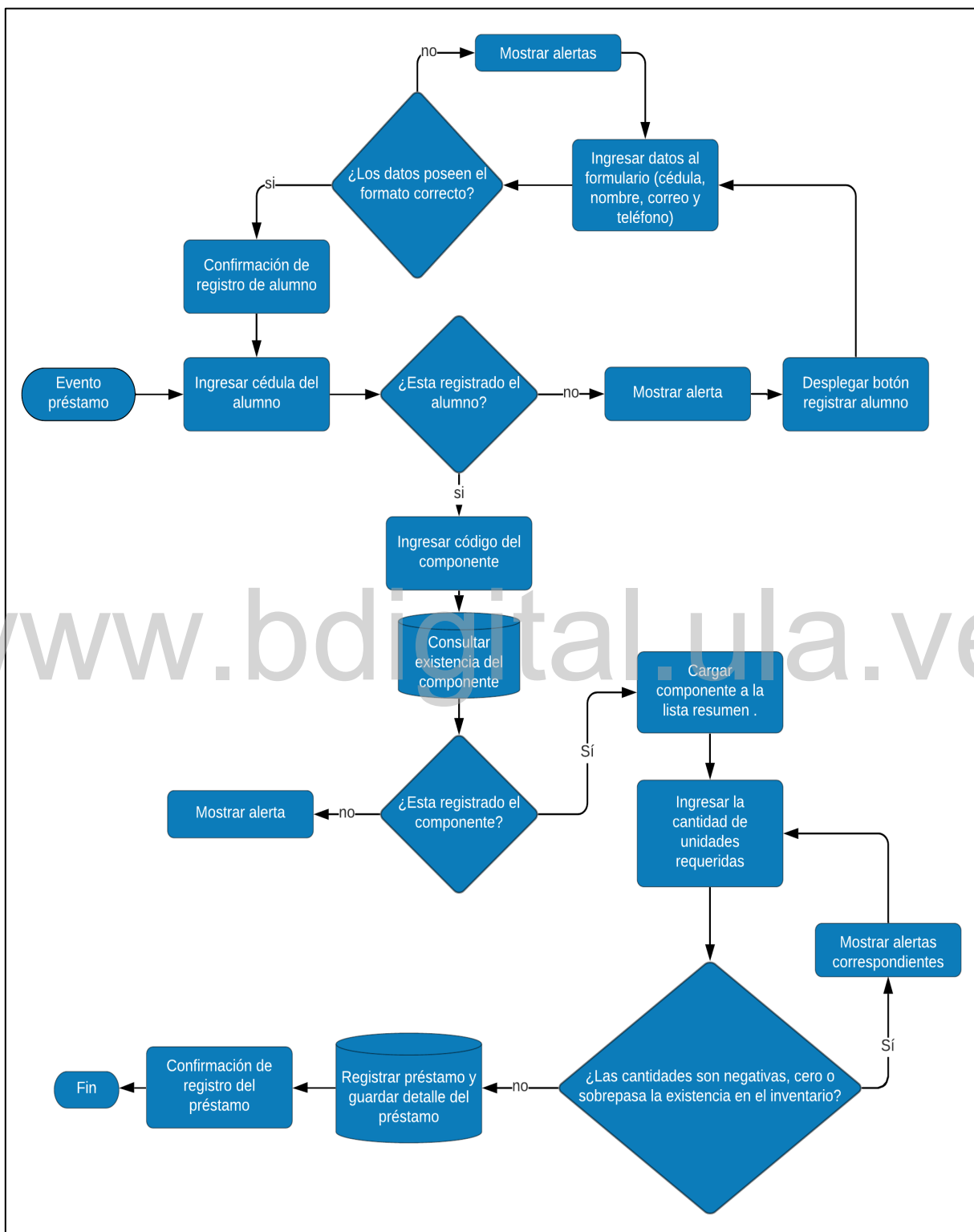


Figura 4.27: Diagrama realizar préstamo – Elaboración propia.

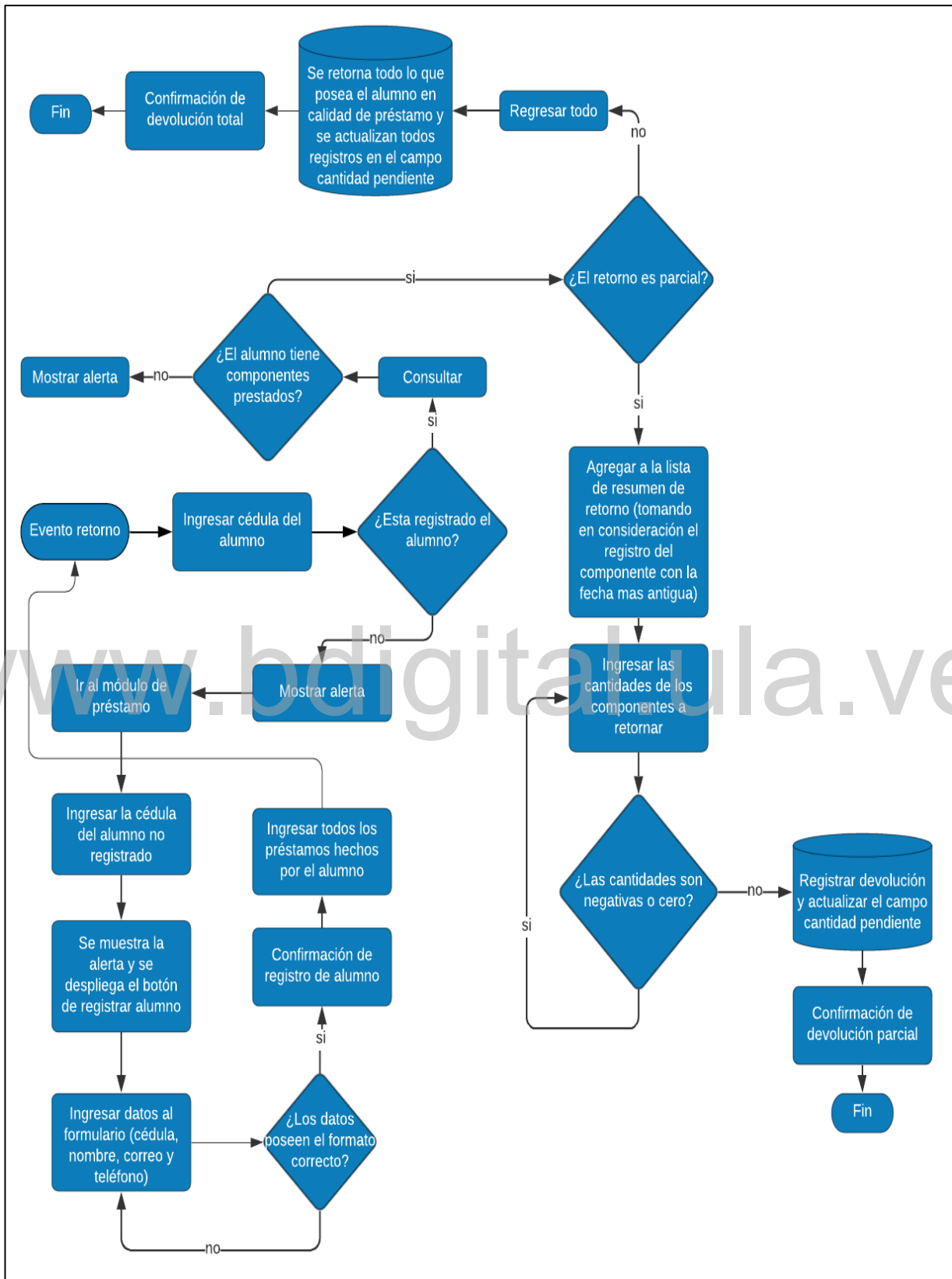


Figura 4.28: Diagrama realizar devolución – Elaboración propia.

4.5 RESTABLECIMIENTO DE CONTRASEÑA

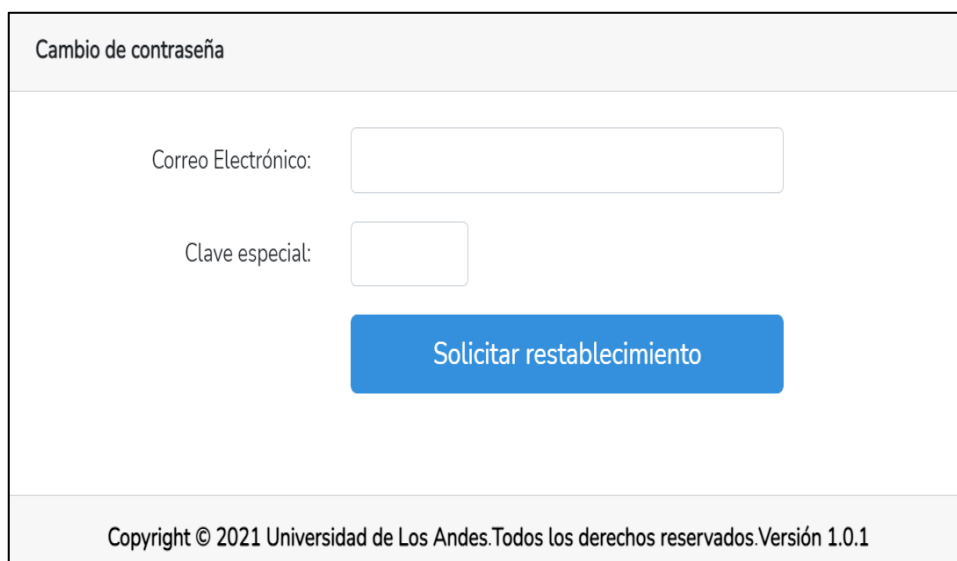
La aplicación S.I.T.E.U.L.A estará provista por un módulo de restablecimiento de contraseña de forma local, sin ataduras al acceso a internet o a la conexión telefónica. El funcionamiento de este módulo se describe a continuación:



Formulario de Login de acceso. Incluye campos para Correo Electrónico y Contraseña, un botón de Mostrar contraseña, un botón de Ingresar, y un enlace de ¿Ha olvidado su contraseña? resaltado con un recuadro rojo. El pie de página indica Copyright © 2021 Universidad de Los Andes. Todos los derechos reservados. Versión 1.0.1.

Figura 4.29: Formulario de Login

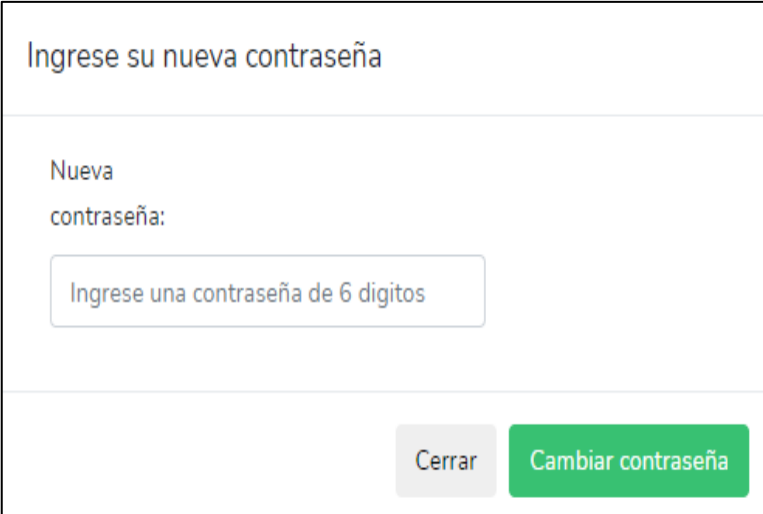
En la figura 4.29 se puede apreciar en el rectángulo de color rojo, un enlace que direcciona al formulario de solicitud de restablecimiento de contraseña.



Formulario de Cambio de contraseña. Incluye campos para Correo Electrónico y Clave especial, un botón de Solicitar restablecimiento, y el pie de página: Copyright © 2021 Universidad de Los Andes. Todos los derechos reservados. Versión 1.0.1.

Figura 4.30: Formulario de solicitud de restablecimiento de contraseña.

Para poder ingresar la nueva contraseña a través del modal de la figura 4.31, se deben ingresar el correo electrónico y la clave especial asociada a un usuario registrado previamente en el sistema.



Modal de cambio de contraseña. El modal tiene un título "Ingrese su nueva contraseña". Debajo del título, hay un campo de texto con el texto "Nueva contraseña:" y un campo de entrada con el texto "Ingrese una contraseña de 6 dígitos". En la parte inferior del modal, hay dos botones: "Cerrar" (gris) y "Cambiar contraseña" (verde).

Figura 4.31: Modal de cambio de contraseña.

www.bdigital.ula.ve

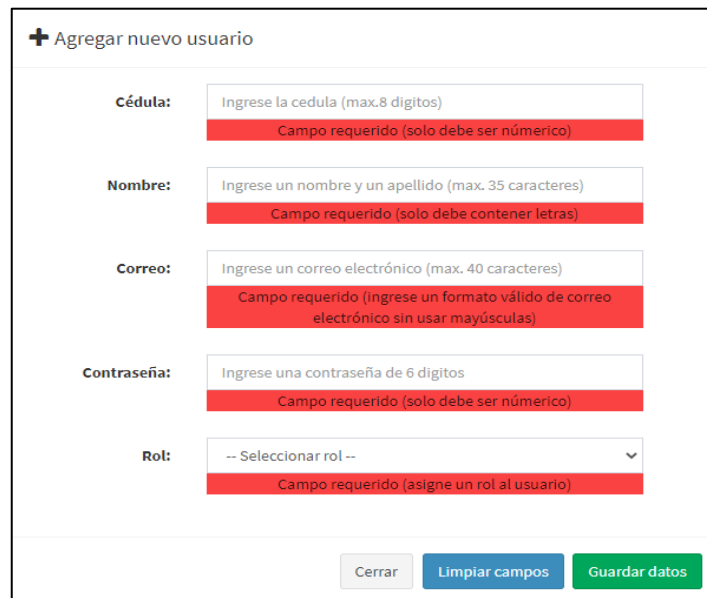
CAPÍTULO 5

VALIDACIONES

El desarrollo de cualquier aplicación web requiere verificar su funcionamiento y constatar que todo opera correctamente. En este capítulo, se describirán las pruebas de caja negra aplicadas al sistema, estas consisten en analizar la salida obtenida, ante una entrada dada y ver si el resultado es el esperado, de esta forma se logra detectar posibles fallas y corregirlas de inmediato. Adicionalmente, se mostrarán las alertas diseñadas para interactuar con el usuario y señalar errores esperados o confirmar operaciones realizadas correctamente.

5.1 ALERTAS

- Formularios



The image shows a web form titled '+ Agregar nuevo usuario'. It contains five input fields, each with a red error message below it:

- Cédula:** Ingrese la cedula (max.8 digitos). Alerta: Campo requerido (solo debe ser númeroico).
- Nombre:** Ingrese un nombre y un apellido (max. 35 caracteres). Alerta: Campo requerido (solo debe contener letras).
- Correo:** Ingrese un correo electrónico (max. 40 caracteres). Alerta: Campo requerido (ingrese un formato válido de correo electrónico sin usar mayúsculas).
- Contraseña:** Ingrese una contraseña de 6 digitos. Alerta: Campo requerido (solo debe ser númeroico).
- Rol:** -- Seleccionar rol --. Alerta: Campo requerido (asigne un rol al usuario).

At the bottom of the form are three buttons: 'Cerrar' (grey), 'Limpiar campos' (blue), and 'Guardar datos' (green).

Figura 5.1: Alertas del formulario de registro de usuario.

Editar usuario

Cédula: a
Campo requerido (solo debe ser numérico)

Nombre: 1
Campo requerido (solo debe contener letras)

Correo: viloriaf
Campo requerido (ingresar un formato válido de correo electrónico)

Rol de usuario: -- Seleccionar rol --
Campo requerido (asigne un rol al usuario)

Cerrar Actualizar datos

Figura 5.2: Alertas del formulario de edición de usuario.

+ Agregar nuevo alumno

Cédula: Ingrese la cedula (max.8 digitos)
Campo requerido (solo debe ser numérico)

Nombre: Ingrese nombres y apellidos (max.35 caracteres)
Campo requerido (solo debe contener letras)

Correo: Ingrese un correo electrónico (max.40 caracteres)
Campo requerido (ingresar un formato válido de correo electrónico)

Teléfono: Ingrese un número de teléfono
Campo requerido (solo debe ser numérico)

Cerrar Limpiar campos Guardar datos

Figura 5.3: Alertas del formulario de registro de alumno.

Editar alumno

Cédula: a
Campo requerido (solo debe ser numérico)

Nombre: 1
Campo requerido (solo debe contener letras)

Correo: rbotti
Campo requerido (Ingresar un formato válido de correo electrónico)

Teléfono: a
Campo requerido (solo debe ser numérico)

Cerrar Actualizar datos

Figura 5.4: Alertas del formulario de edición de alumno.

+ Agregar nuevo componente

Código: Ingrese un código (max. 10 caracteres)
Campo requerido (puede ser alfabético, numérico o alfanumérico)

Nombre: Ingrese nombre del componente (max. 25 caracteres)
Campo requerido (puede ser alfabético, numérico o alfanumérico)

Descripción: Ingrese una breve descripción del componente (campo no obligatorio, max. 45 caracteres)
Campo no obligatorio (puede ser alfabético, numérico o alfanumérico)

Cantidad: Ingrese la cantidad inicial
Campo requerido (solo debe ser numérico)

Cerrar Limpiar campos Guardar datos

Figura 5.5: Alertas del formulario de registro de componente.

Editar componente

Código:
 Campo requerido (puede ser alfabético, numérico o alfanumérico)

Nombre:
 Campo requerido (puede ser alfabético, numérico o alfanumérico)

Descripción:
 Campo no obligatorio (puede ser alfabético, numérico o alfanumérico)

Cerrar Actualizar datos

Figura 5.6: Alertas del formulario de edición de componente.

+ Agregar nuevo componente

Código:
 Campo requerido (puede ser alfabético, numérico o alfanumérico)

Nombre:
 Campo requerido (puede ser alfabético, numérico o alfanumérico)

Descripción:
 Campo no obligatorio (puede ser alfabético, numérico o alfanumérico)

Cantidad:

Cerrar Limpiar campos Guardar datos

Figura 5.7: Alertas del formulario de registro de componente en el módulo de entradas al inventario.

- Emergentes

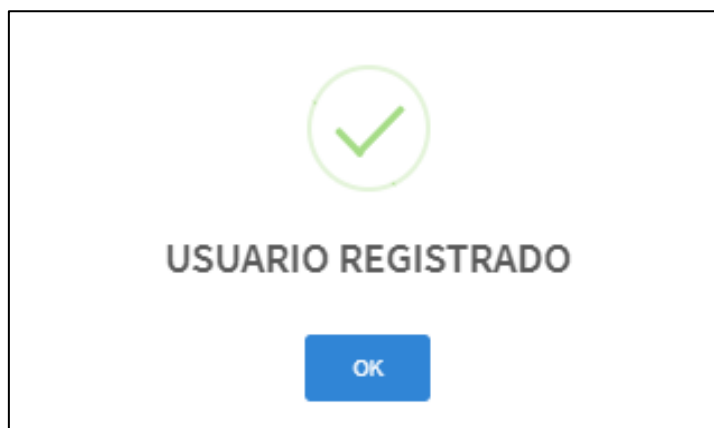


Figura 5.8: Alerta de confirmación de registro de usuario.



Figura 5.9: Alerta de confirmación de actualización de usuario.

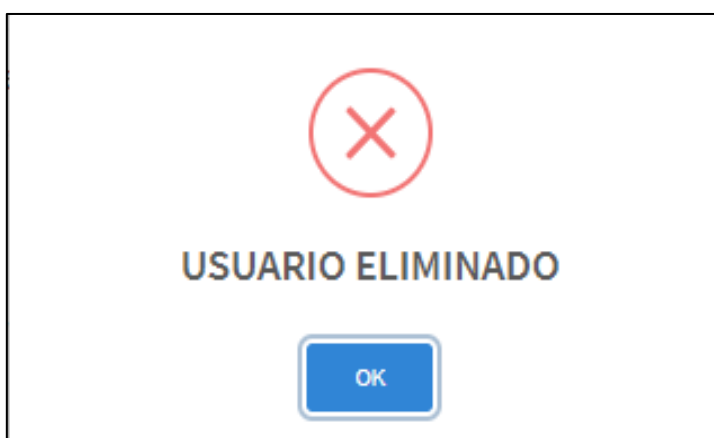


Figura 5.10: Alerta de confirmación de eliminación de usuario.

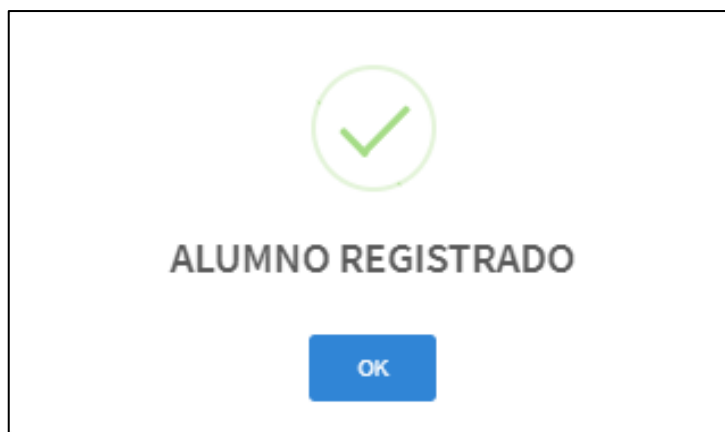


Figura 5.11: Alerta de confirmación de registro de alumno.



Figura 5.12: Alerta de confirmación de actualización de alumno.

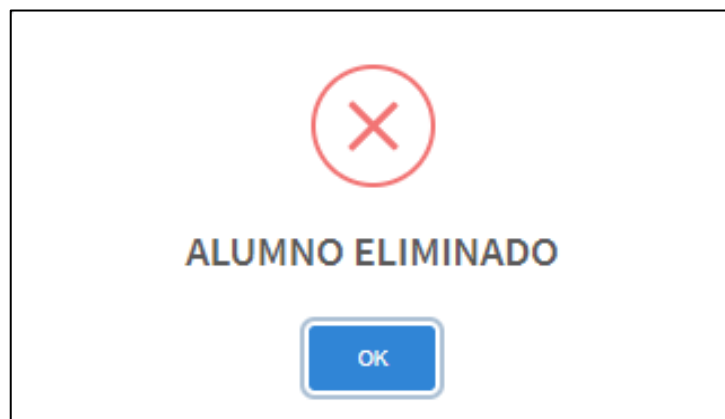


Figura 5.13: Alerta de confirmación de eliminación de alumno.

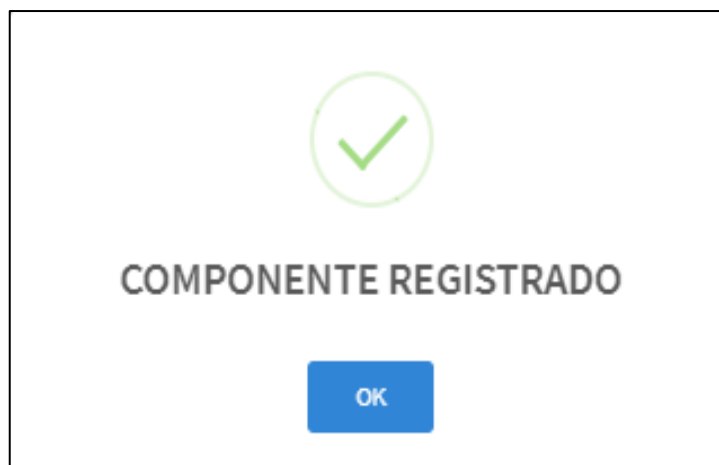


Figura 5.14: Alerta de confirmación de registro de componente.



Figura 5.15: Alerta de confirmación de actualización de componente.

5.2 PRUEBAS A LOS CASOS DE USO

Los casos de uso que comprenden el módulo de usuarios, alumnos e inventario son más sencillos y poseen procedimientos similares, en la siguiente tabla se muestran los resultados.

Tabla 5.1: Pruebas a los casos de uso básicos.

Caso de uso	Parámetros de entrada	Resultado esperado	Resultado obtenido
Registrar usuario	Click al botón nuevo usuario. Luego, se ingresa la cédula, nombre, correo y rol.	El sistema realiza el registro y despliega la alerta de la figura 5.8.	Satisfactorio

Tabla 5.1: Continuación de las pruebas a los casos de uso básicos.

Caso de uso	Parámetros de entrada	Resultado esperado	Resultado obtenido
Editar usuario	Click al botón editar usuario. Luego, se procede a modificar los campos de cédula, nombre, correo o rol.	El sistema realiza la actualización y despliega la alerta de la figura 5.9.	Satisfactorio
Eliminar usuario	Click al botón de eliminar usuario y luego al botón de eliminar.	El sistema elimina al usuario seleccionado y despliega la alerta de la figura 5.10.	Satisfactorio
Registrar alumno	Click al botón nuevo alumno. Luego, se ingresa la cédula, nombre, correo y teléfono.	El sistema realiza el registro y despliega la alerta de la figura 5.11.	Satisfactorio
Editar alumno	Click al botón editar alumno. Luego, se procede a modificar los campos de cédula, nombre, correo o teléfono.	El sistema realiza la actualización y despliega la alerta de la figura 5.12.	Satisfactorio
Eliminar alumno	Click al botón de eliminar alumno y luego al botón de eliminar.	El sistema elimina al alumno seleccionado y despliega la alerta de la figura 5.13.	Satisfactorio
Registrar componente	Click al botón nuevo componente. Luego, se ingresa el código, nombre, descripción (no obligatorio) y cantidad inicial.	El sistema realiza el registro y despliega la alerta de la figura 5.14.	Satisfactorio
Editar componente	Click al botón editar componente. Luego, se procede a modificar los campos de código, nombre o descripción.	El sistema realiza la actualización y despliega la alerta de la figura 5.15.	Satisfactorio
Generar código de barras	Click al botón generar código	El sistema realiza la actualización y despliega la ventana de la figura 4.6	Satisfactorio

5.2.1 Pruebas al caso de uso prestar componente

Para este caso de uso, se toman en consideración los siguientes escenarios:

Tabla 5.2: Pruebas prestar componente.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Alumno no registrado	El sistema muestra una alerta indicando que el alumno no se encuentra registrado y despliega un botón para realizar el registro (ver figura 5.16).	Satisfactorio
Componente no registrado	El sistema muestra una alerta indicando que el componente no se encuentra registrado y despliega un botón para realizar el registro, esto sucede solo si el usuario en sesión posee el rol de administrador. (ver figura 5.17).	Satisfactorio
Alumno registrado	El sistema muestra los datos del alumno en el campo de nombre de alumno.	Satisfactorio
Componente registrado	El sistema añade el componente ingresado a la lista de resumen para el préstamo.	Satisfactorio
Ingreso negativo o cero en el campo unidades	El sistema muestra una alerta indicando este evento (ver figura 5.18).	Satisfactorio
Cantidad de unidades requeridas mayor al número de unidades en existencia.	El sistema muestra una alerta indicando este evento (ver figura 5.19).	Satisfactorio
Componente sin unidades disponibles en el inventario	El sistema muestra una alerta indicando este evento (ver figura 5.20).	Satisfactorio
Registrar préstamo sin ingresar la cédula del alumno	El sistema muestra una alerta indicando este evento (ver figura 5.21).	Satisfactorio
Registrar préstamo	El sistema muestra una alerta indicando este evento (ver figura 5.22).	Satisfactorio

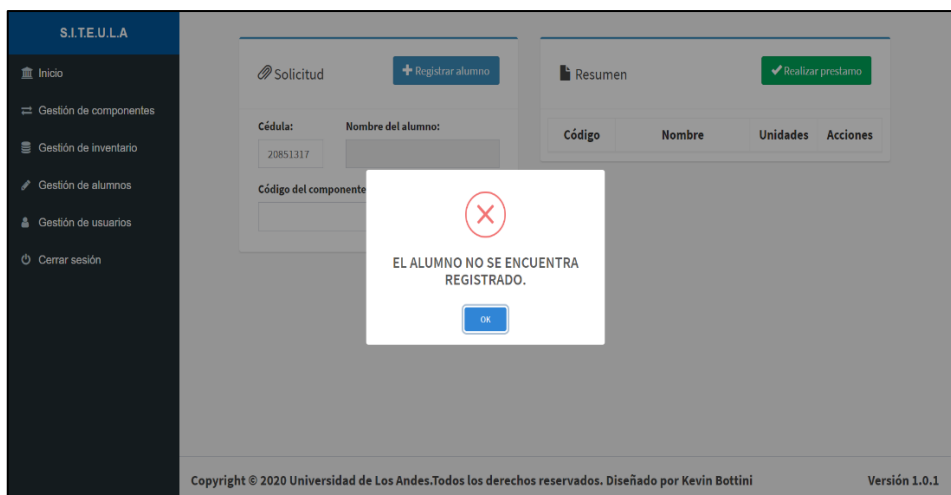


Figura 5.16: Alerta de alumno no registrado.

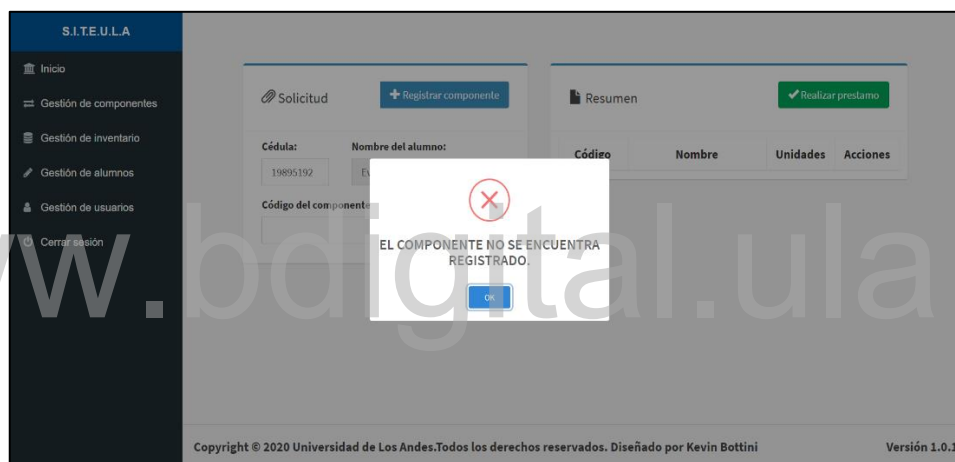


Figura 5.17: Alerta de componente no registrado.

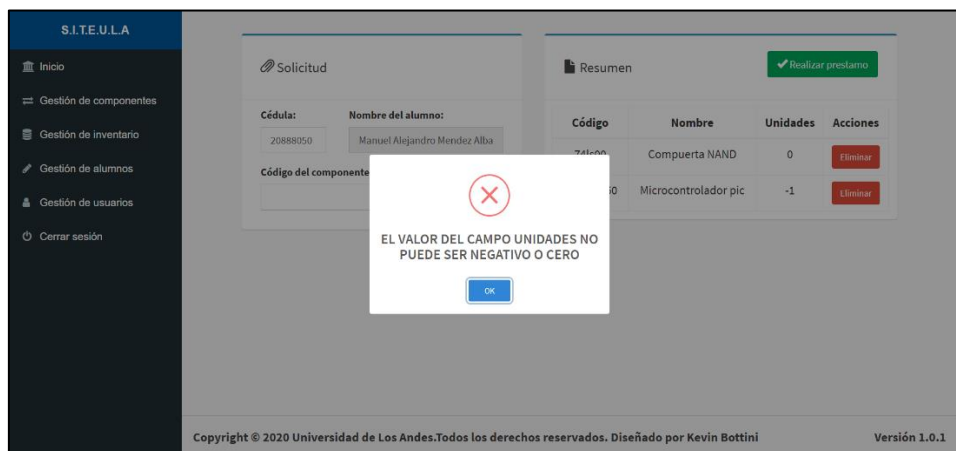


Figura 5.18: Alerta de valores negativos o cero en el campo unidades.

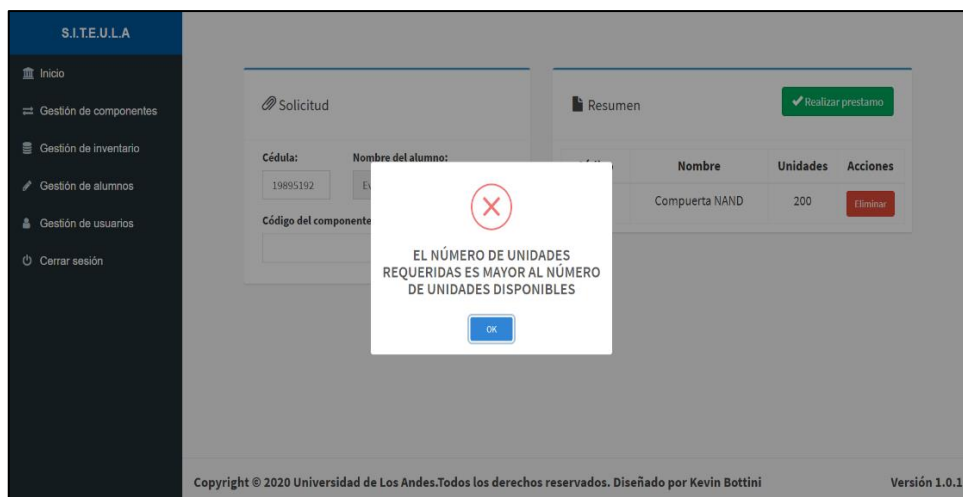


Figura 5.19: Alerta de exceso de unidades requeridas para el préstamo.

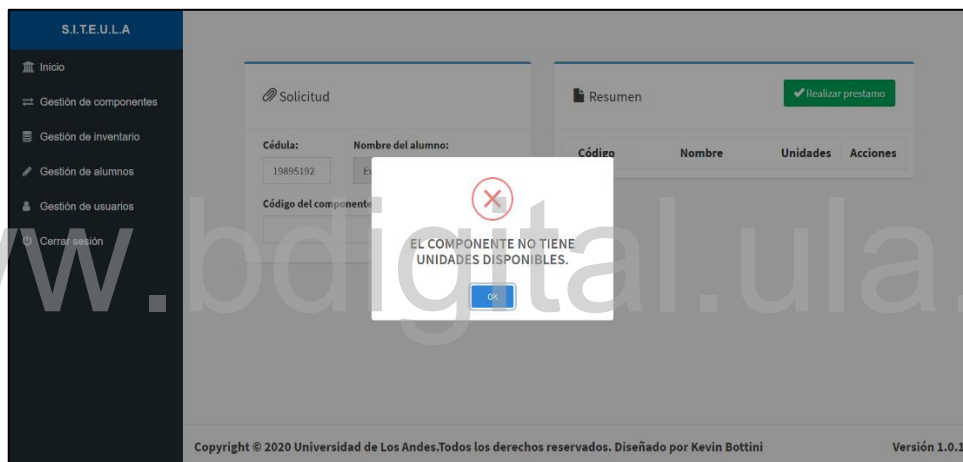


Figura 5.20: Alerta de componente sin unidades disponibles para el préstamo.

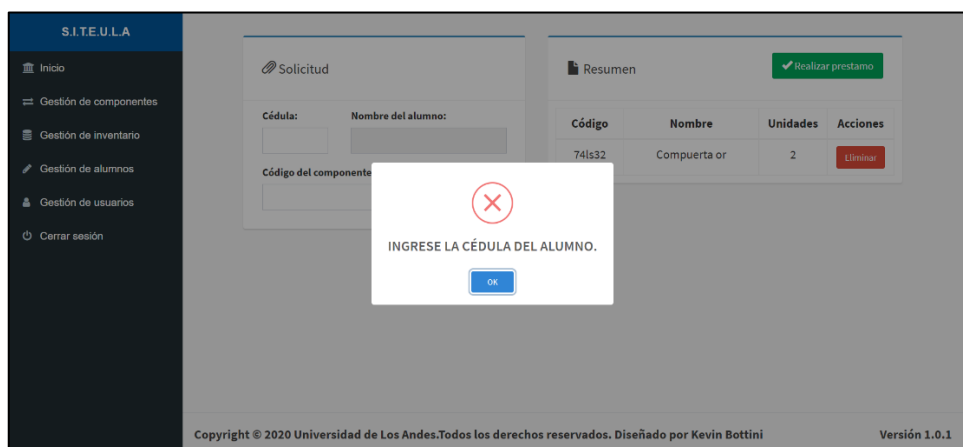


Figura 5.21: Alerta de intento de registro de préstamo sin indicar la cedula del alumno.

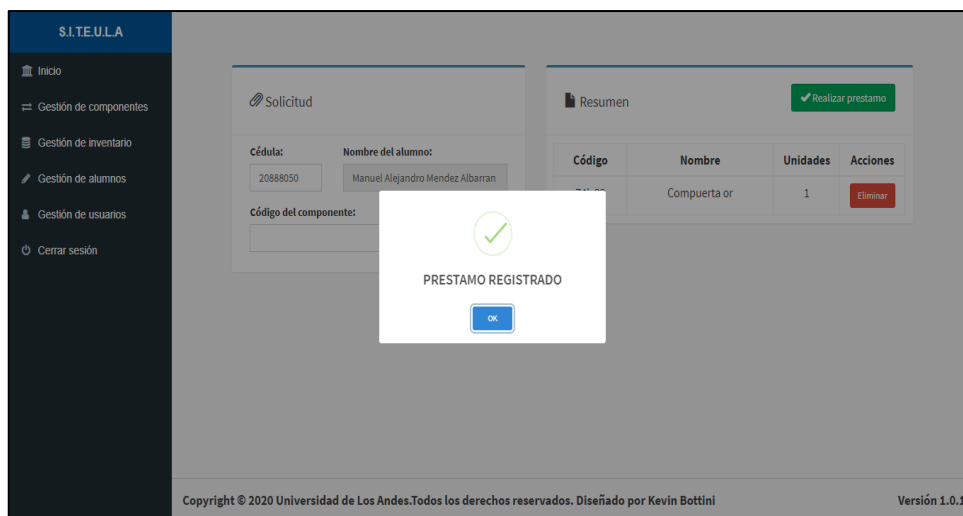


Figura 5.22: Alerta de registro de préstamo exitoso.

5.2.2 Pruebas al caso de uso retornar componente

Para este caso de uso, se toman en consideración los siguientes escenarios:

Tabla 5.3: Pruebas retornar componente.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Alumno no registrado	El sistema muestra una alerta igual que la del módulo de préstamo, indicando que el alumno no se encuentra registrado (ver figura 5.23).	Satisfactorio
Sin componentes prestados	El sistema muestra una alerta indicando que el alumno no posee componentes en calidad de préstamo (ver figura 5.24).	Satisfactorio
Consultar componentes prestados por el alumno	El sistema muestra en una tabla la lista de todos los componentes en posesión del estudiante.	Satisfactorio
Ingreso negativo o cero en el campo unidades	El sistema muestra una alerta indicando este evento (ver figura 5.25).	Satisfactorio
Devolución parcial de los componentes	Cada registro existente sobre un componente en préstamo, posee un botón que agrega a la lista de resumen (ver figura 5.26).	Satisfactorio

Tabla 5.3: Continuación pruebas retornar componente.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Devolución completa de los componentes (botón regresar todo).	Devolución de la totalidad de los componentes prestados (ver figura 5.27).	Satisfactorio

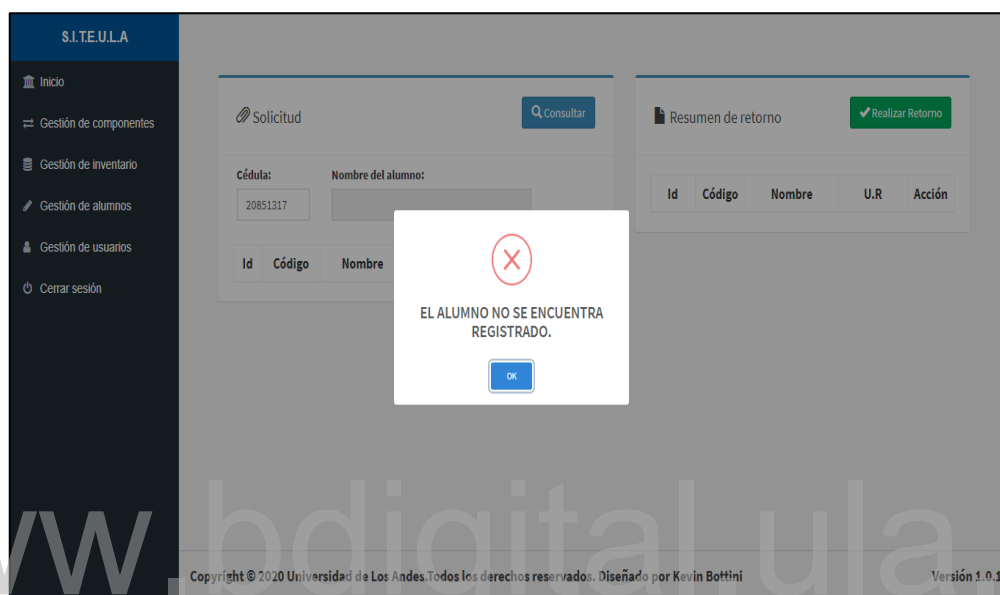


Figura 5.23: Alerta de alumno no registrado.

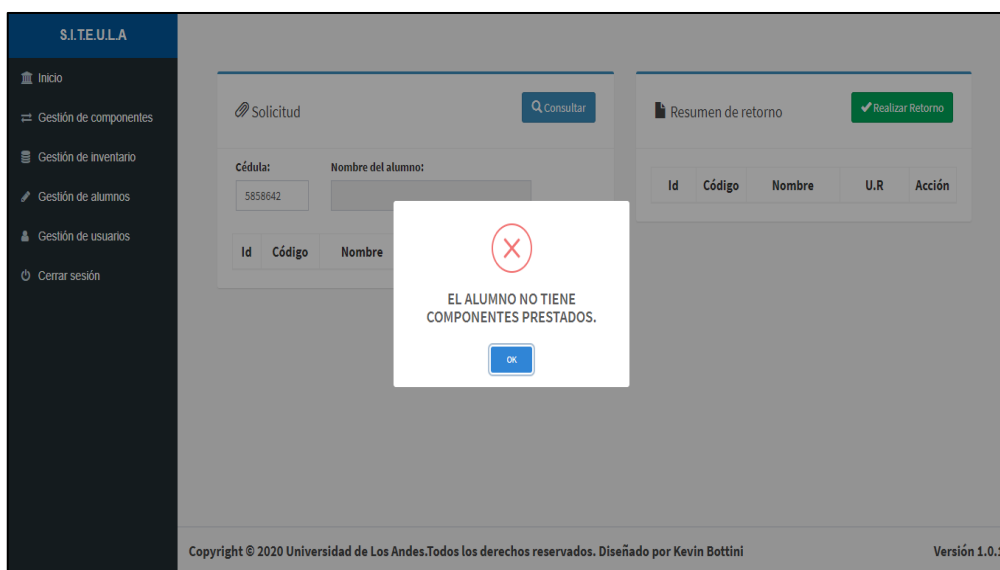


Figura 5.24: Alerta de alumno sin componentes prestados.

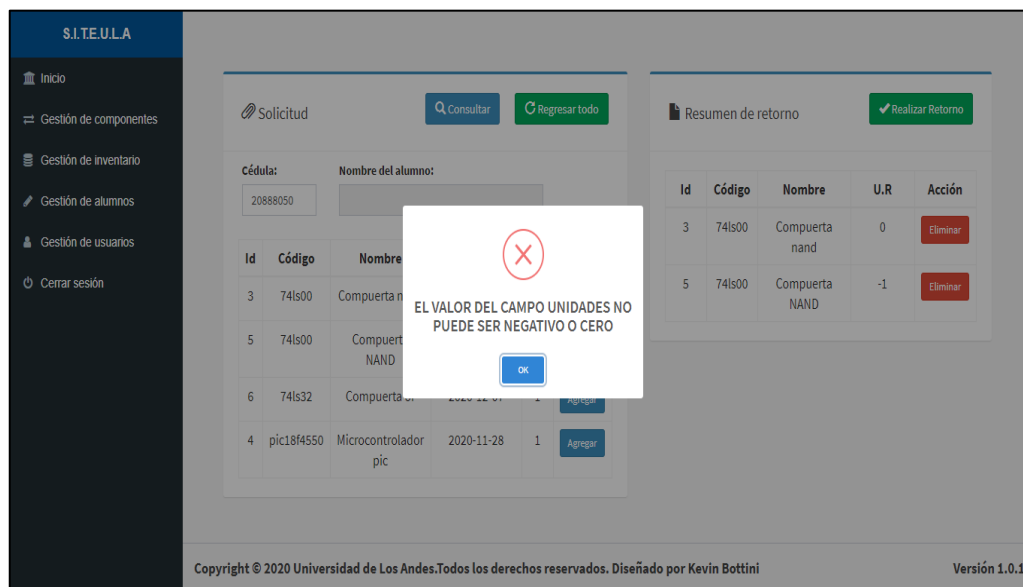


Figura 5.25: Alerta de valores negativos o cero en el campo unidades.

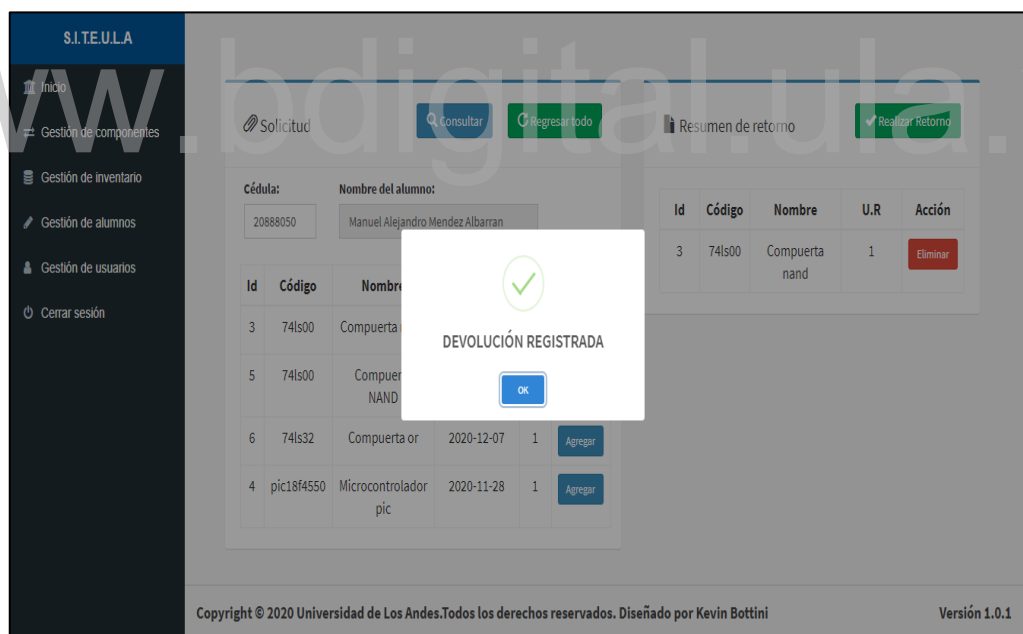


Figura 5.26: Alerta de devolución parcial.

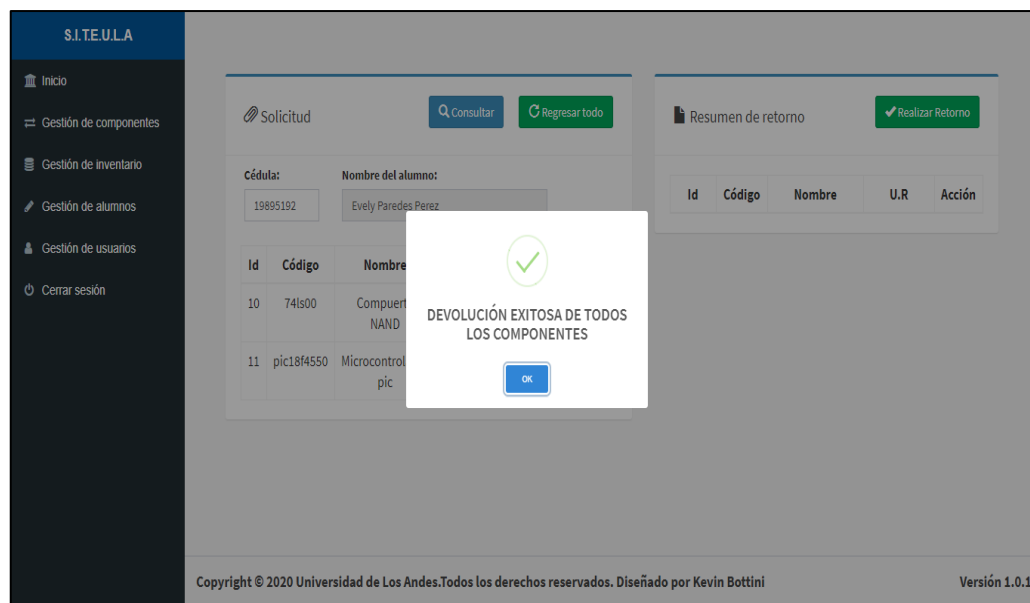


Figura 5.27: Alerta de devolución total.

5.2.3 Pruebas al caso de uso entradas al inventario

Para este caso de uso, se toman en consideración los siguientes escenarios:

Tabla 5.4: Pruebas entradas al inventario.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Componente registrado sin seleccionar motivo de ingreso.	El sistema muestra una alerta que indica que se debe ingresar el motivo de ingreso del componente, antes de cargarlo a la lista de resumen (ver figura 5.28).	Satisfactorio
Motivo de ingreso seleccionado y componente no registrado.	El sistema muestra una alerta indicando que el componente no se encuentra registrado, si el usuario es administrador, se despliega un botón para el registro del componente (ver figura 5.29).	Satisfactorio
Motivo de ingreso seleccionado y componente registrado.	El sistema añade el componente a la lista de resumen de entradas al inventario.	Satisfactorio

Tabla 5.4: Continuación pruebas entradas al inventario.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Ingreso negativo o cero en el campo unidades	El sistema muestra una alerta indicando este evento (ver figura 5.30).	Satisfactorio
Realizar ingreso	El sistema muestra una alerta que indica que el ingreso de componentes al inventario se ha realizado correctamente (ver figura 5.31).	Satisfactorio

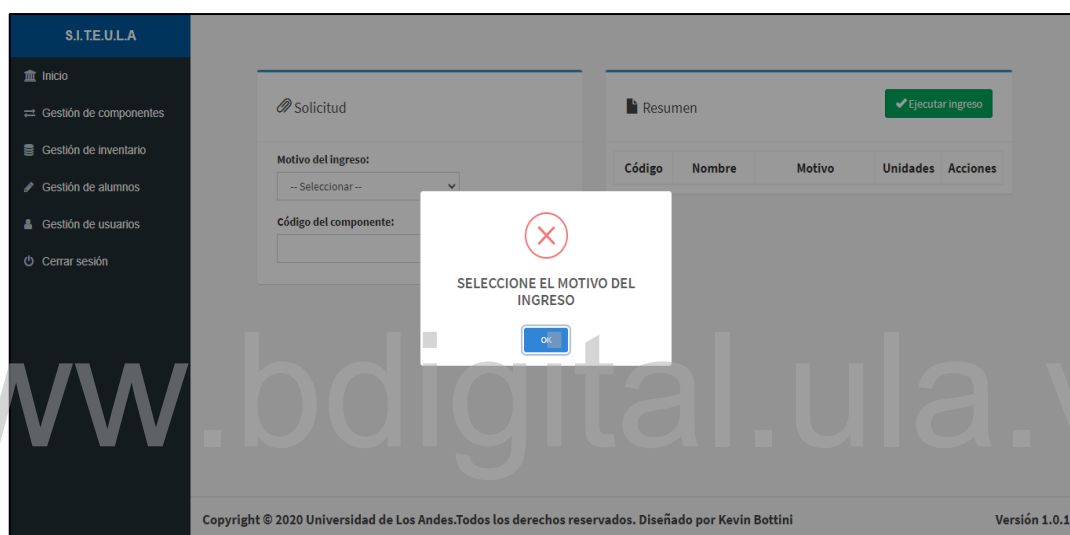


Figura 5.28: Alerta para recordar seleccionar el motivo de ingreso.

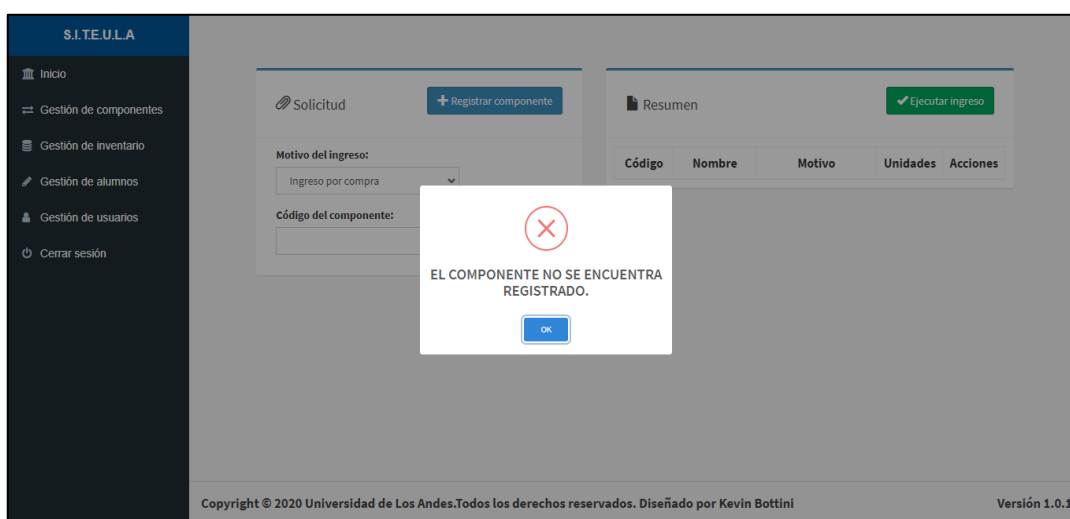


Figura 5.29. Alerta de componente no registrado.

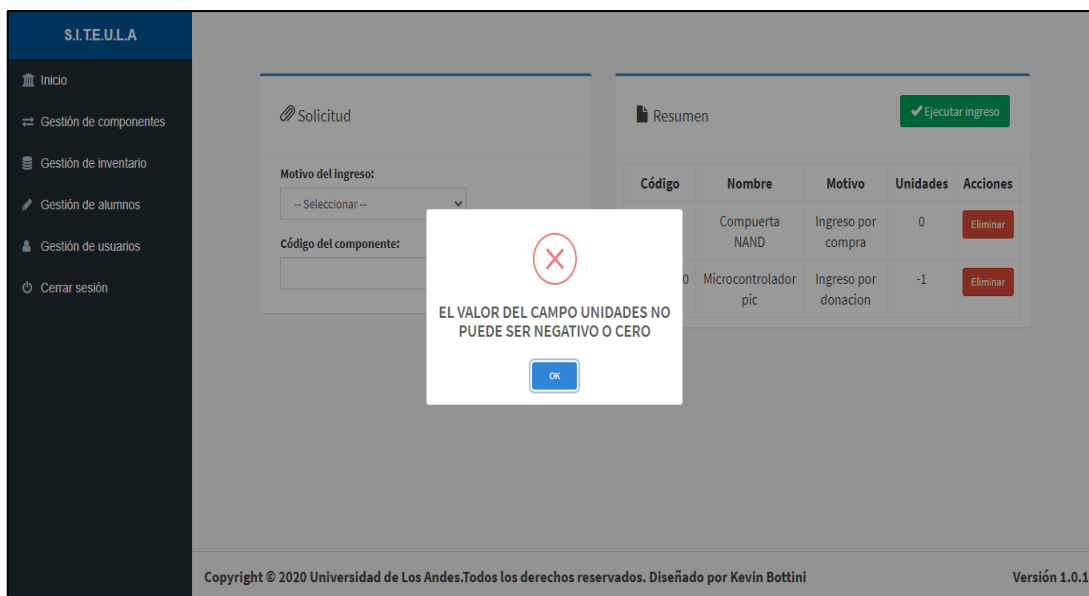


Figura 5.30: Alerta de valores negativos o cero en el campo unidades.

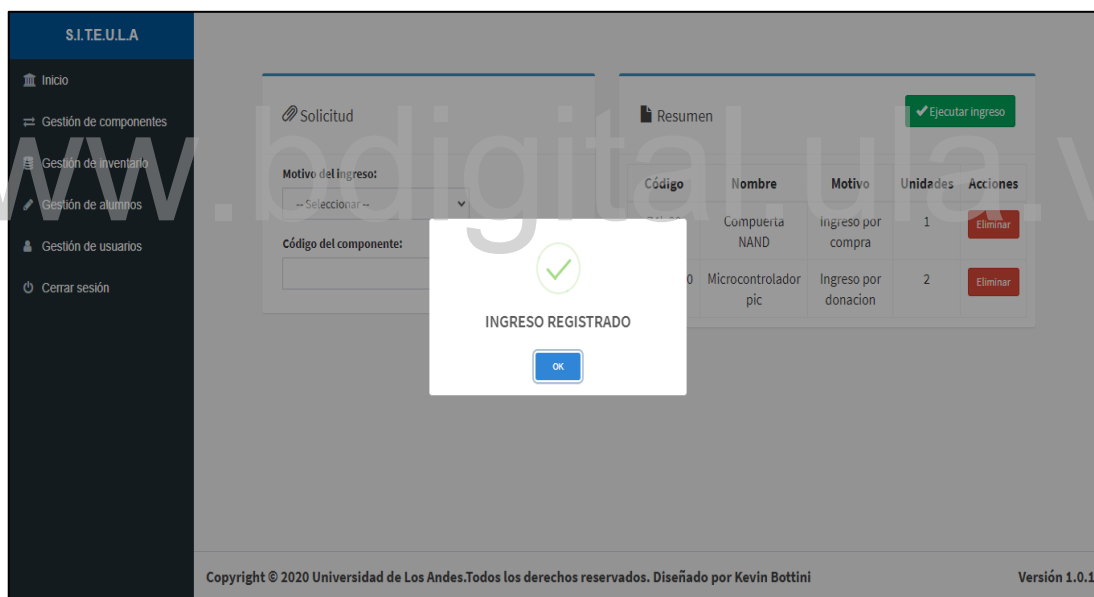


Figura 5.31: Alerta de confirmación de ingreso.

5.2.4 Pruebas al caso de uso salidas del inventario

Para este caso de uso, se toman en consideración los siguientes escenarios:

Tabla 5.5: Pruebas salidas del inventario.

Parámetros de entrada	Resultado obtenido	Resultado esperado
Componente registrado sin seleccionar motivo de ingreso.	El sistema muestra una alerta que indica que se debe ingresar el motivo de ingreso del componente, antes de cargarlo a la lista de resumen (ver figura 5.32).	Satisfactorio
Motivo de ingreso seleccionado y componente no registrado.	El sistema muestra una alerta indicando que el componente no se encuentra registrado (ver figura 5.33).	Satisfactorio
Motivo de ingreso seleccionado y componente registrado.	El sistema añade el componente a la lista de resumen de entradas al inventario.	Satisfactorio
Ingreso negativo o cero en el campo unidades	El sistema muestra una alerta indicando este evento (ver figura 5.34).	Satisfactorio
Realizar egreso	El sistema muestra una alerta que indica que el ingreso de componentes al inventario se ha realizado correctamente (ver figura 5.35).	Satisfactorio

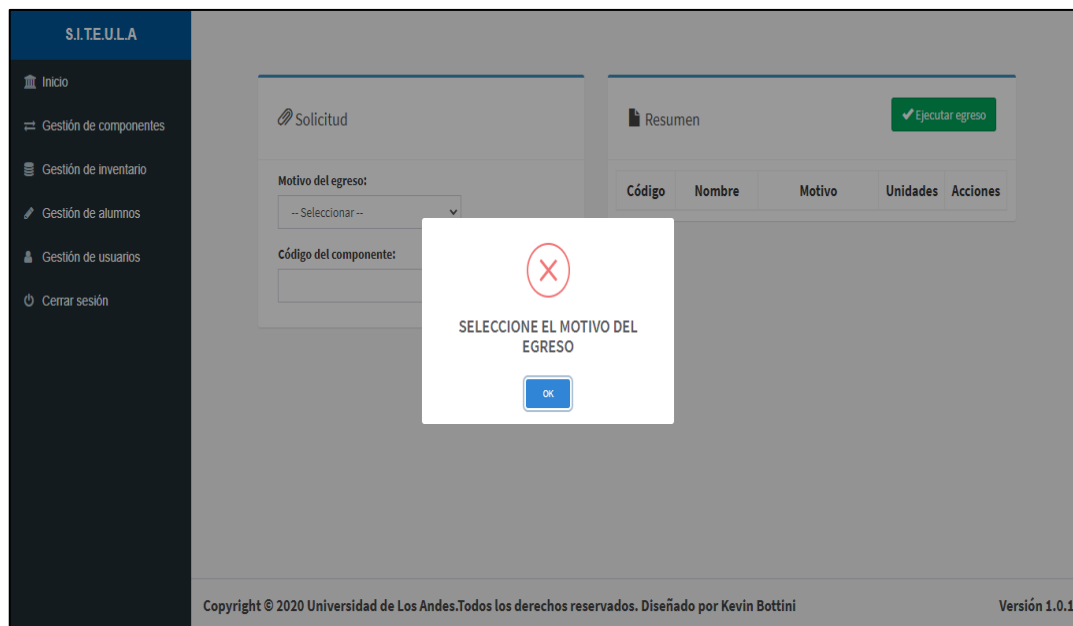


Figura 5.32: Alerta para recordar seleccionar el motivo de egreso.

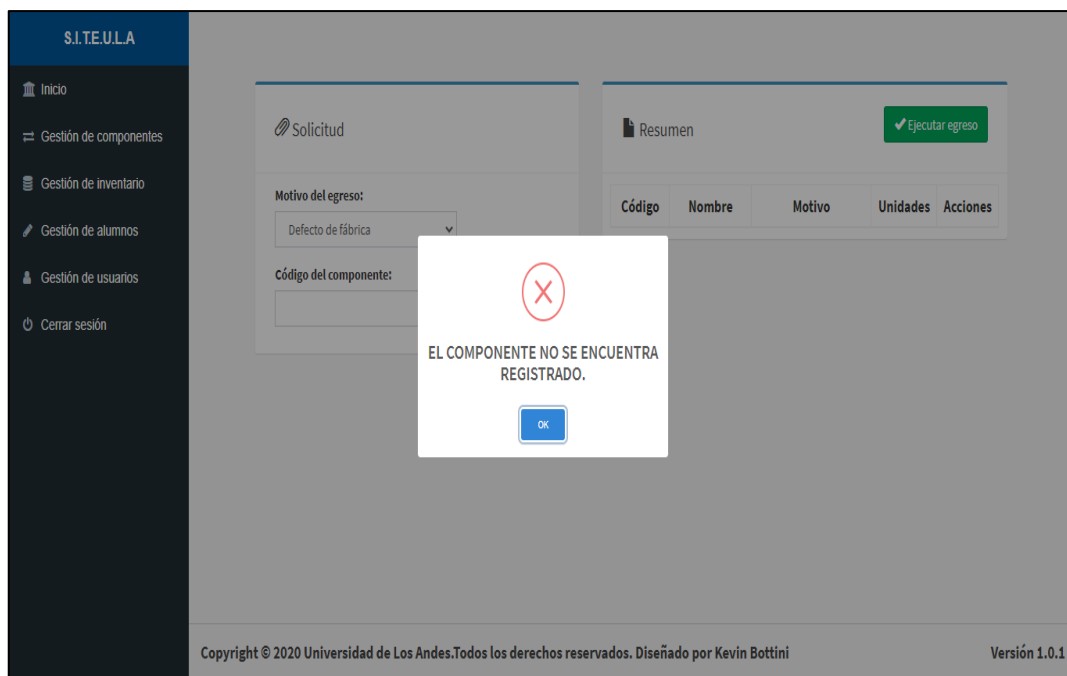


Figura 5.33: Alerta de componente no registrado.

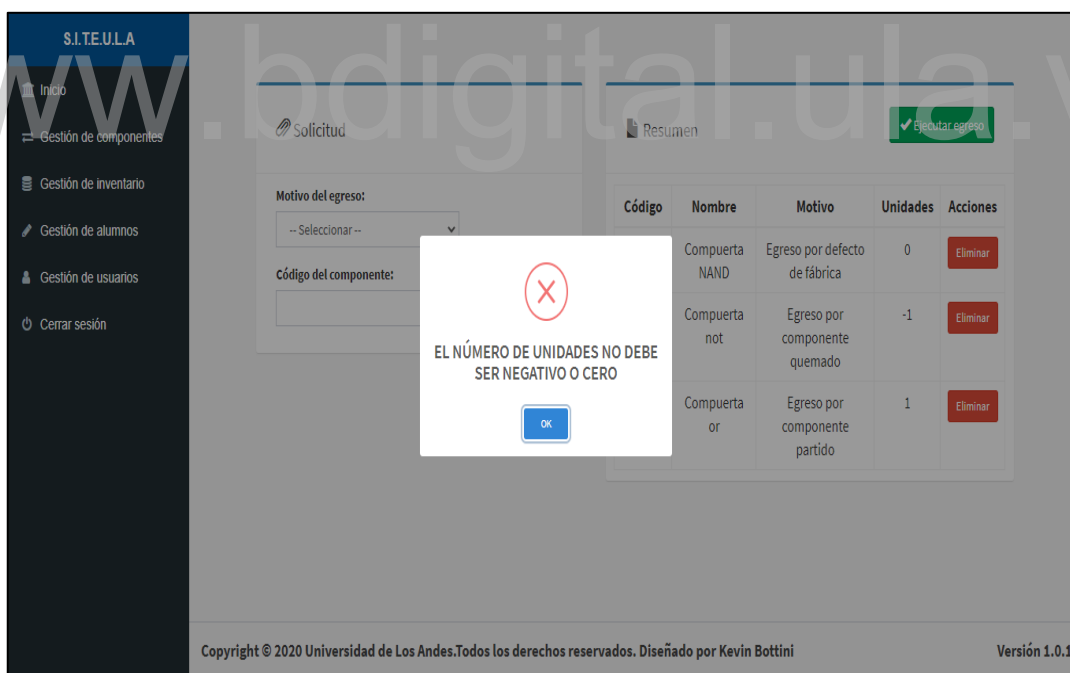


Figura 5.34: Alerta de valores negativos o cero en el campo unidades.

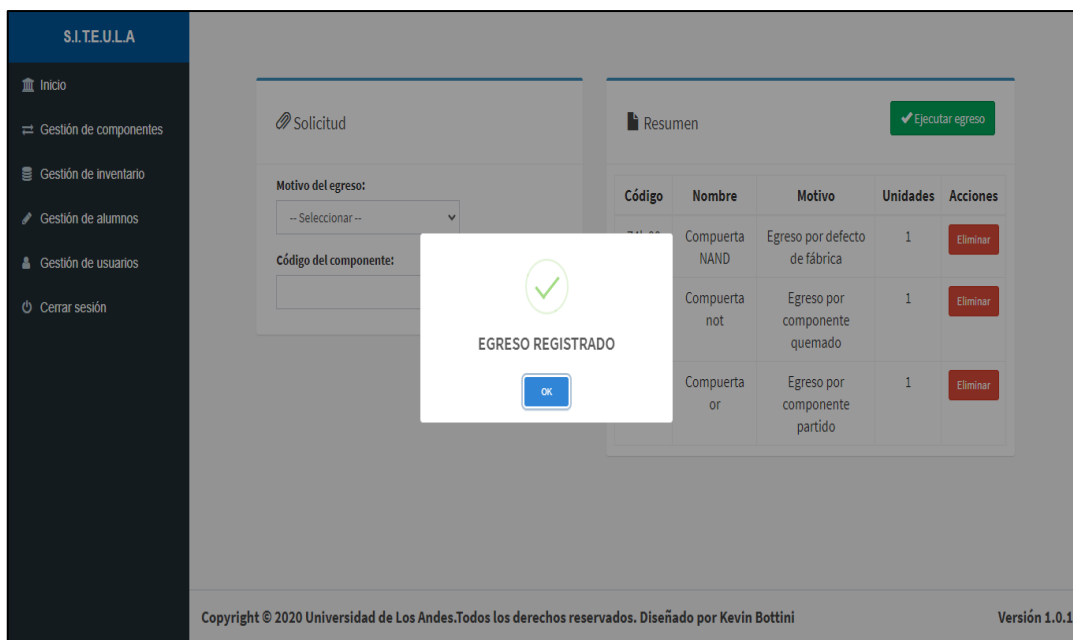


Figura 5.35: Alerta de confirmación de egreso.

5.3 ROLES DE USUARIO

En la siguiente figura, se muestra una página de error para indicarle al usuario con el perfil de técnico que no posee acceso para ingresar a determinados módulos del sistema.

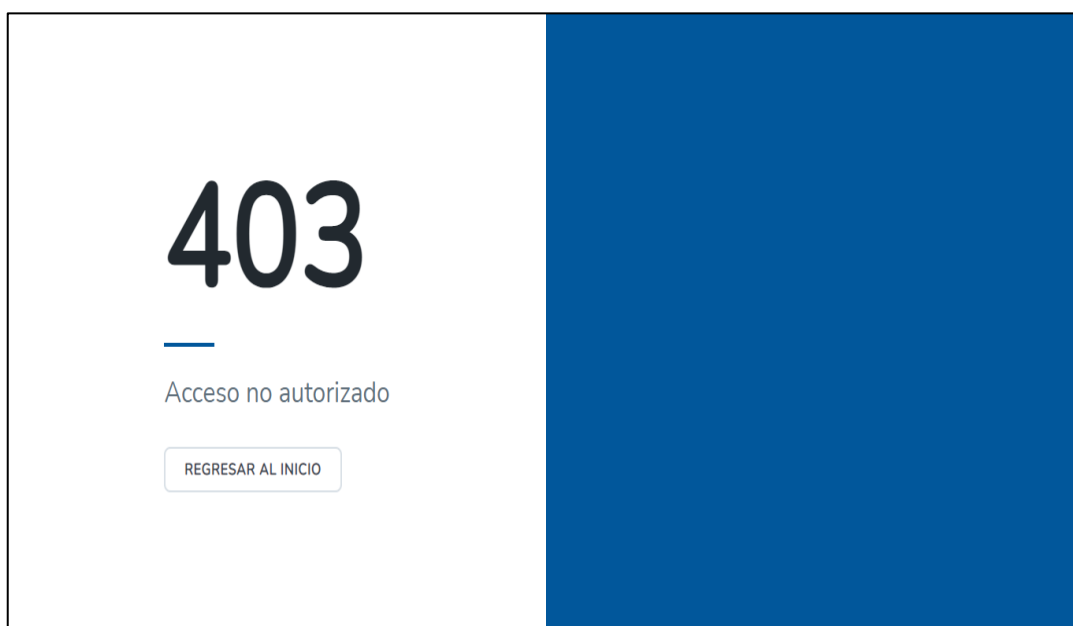


Figura 5.36: Pagina de acceso no autorizado.

CAPÍTULO 6

CONCLUSIONES Y MEJORAS FUTURAS

6.1 CONCLUSIONES

La metodología ágil *Scrumban*, proporciona un método más rápido para el desarrollo de sistemas web que requiere de alto grado de funcionalidades, en otras palabras, establece lineamientos bastante claros sobre como ejecutar las distintas tareas que se plantean en el levantamiento de requerimientos (*Product Backlog*), para este proyecto todos los objetivos funcionales y no funcionales fueron alcanzados satisfactoriamente. Por su parte, el listado de casos de uso elegidos, permitió diseñar una aplicación de inventario web con múltiples módulos de administración, asegurando escalabilidad con los datos que se guardaran en la base de datos.

El uso de este aplicativo por parte de los trabajadores del Taller de Electrónica U.L.A, recaba toda la información significativa en un solo lugar, evitando la redundancia de los datos, perdida de la información referente a préstamos y retornos, componentes defectuosos en el inventario que afectan los proyectos realizados por los estudiantes y largas colas para las operaciones a realizar por las personas que hacen uso de este servicio.

La interfaz de usuario implementada, cumple varios de los fundamentos de diseño de interfaces sugeridos universalmente como la uniformidad, estructura, contraste, composición, alineación, balance, semejanza, interacción, etc.

Finalmente, se ejecutaron correctamente todas las pruebas al *software*, obteniendo resultados satisfactorios que permite concluir que el sistema funciona como se esperaba.

6.2 MEJORAS FUTURAS

A continuación, se listan las posibles adiciones para el Sistema de Inventario del Taller de Electrónica U.L.A y poder hacer un *software* más completo:

- Módulo estadístico que permita generar reportes sobre la demanda asociada a la cantidad de alumnos que utilizan este servicio, ya sea diariamente, semanalmente o mensualmente. De igual forma, que permita generar reportes sobre los componentes más demandados por los estudiantes y conocer los que se deben adquirir con más prioridad.
- Enviar el listado de componentes prestados y el registro de devoluciones hechas, al correo registrado de los estudiantes.
- Agregar la ubicación física de los componentes electrónicos dentro del taller.
- Agregar las hojas de datos (*datasheets*) en el registro de los componentes electrónicos.
- Un módulo de generación de solvencias al momento del cierre del expediente de grado.
- Un módulo de consulta de los componentes disponibles con acceso directo para los estudiantes.

REFERENCIAS

- [1] A. R. Herrera Povis, "Sistemas de inventarios," Trabajo de Grado, Fac. Ciencias Matemáticas, Dpto. Investigación Operativa, U.N.M.S.M, Lima, Perú, 2006. Disponible en: http://sisbib.unmsm.edu.pe/bibvirtualdata/monografias/basic/herrera_pa/cap3.pdf. Ultimo acceso: 05 de enero de 2021.
- [2] A. I. Laureano Castro, (2018, Sept. 09). "Metodología de Desarrollo de Software ". Disponible en: <https://studentplace98.blogspot.com/2018/09/metodologia-de-desarrollo-de-software.html>. Ultimo acceso: 08 de enero de 2021.
- [3] L. A. Ramon Villegas, "Implementación de un sistema de inventarios para lograr un mejor servicio y eficientizar la producción de la planta de tubería," Maestría, Fac. Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, Monterrey, México, 2001. Disponible en: <http://eprints.uanl.mx/1029/1/1020148451.PDF>. Ultimo acceso: 16 de enero de 2021.
- [4] H. Sy Corvo, (2019). "Sistemas de inventarios: tipos, métodos y ejemplos" [En línea]. Disponible en: <https://www.lifeder.com/sistema-de-inventarios/>. Ultimo acceso: 23 de enero de 2021.
- [5] E. Morales Feliz, "Diseño de una aplicación de gestión de préstamo de material electrónico para los laboratorios UPC", Trabajo de Grado, Dpto. Electrónica, U.P.C, Barcelona, España, 2016. Disponible en: <https://upcommons.upc.edu/handle/2117/91489>. Ultimo acceso: 04 de febrero de 2021.
- [6] J. Velamazán Sánchez, "Diseño de inventario web de productos químicos para la facultad de química," Trabajo de Grado, Dpto. de Matemáticas Aplicadas y Análisis, U.B, Barcelona, España, 2015. Disponible en: http://diposit.ub.edu/dspace/bitstream/2445/67906/2/memoria_.pdf. Ultimo acceso: 09 de febrero de 2021.
- [7] Y. Orduz Navarrete, "Análisis, diseño e implementación de un sistema de información para la gestión de alquiler y mantenimiento de vehículos," Trabajo de Pasantía, Fac. Ingeniería, U.D.F.J.C, Bogotá, Colombia, 2016. Disponible en: <http://repository.udistrital.edu.co/bitstream/11349/3701/1/OrduzNavarreteYesid2016.pdf>. Ultimo acceso: 13 de febrero de 2021.
- [8] M. B. Ochando, (2014, Feb. 11). "Fundamentos y Diseño de Bases de Datos". Disponible en: <http://ccdoc-basesdedatos.blogspot.com/2014/02/concepto-definicion-y-aspectos-basicos.html>. Ultimo acceso: 18 de febrero de 2021.

- [9] M. B. Ochando, (2014, Feb. 11). "Fundamentos y Diseño de Bases de Datos". [En línea]. Disponible en: <http://ccdodoc-basesdedatos.blogspot.com/2014/02/concepto-definicion-y-aspectos-basicos.html>. Ultimo acceso: 25 de febrero de 2021.
- [10] A. Silberschatz, H. F. Korth y S. Sudarshan, "Fundamentos de Bases de Datos", 4ta ed. Madrid: S.A. MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, 2002. Cap.1, pp.1. Disponible en: http://artemisa.unicauca.edu.co/~cardila/Libro_Silberschatz.pdf. Ultimo acceso: 18 de diciembre de 2020.
- [11] MySQL (2019). Disponible en: <https://hostingpedia.net/mysql.html>. Ultimo acceso: 18 de noviembre de 2020. Ultimo acceso: 22 de noviembre de 2020.
- [12] M. A. Alvarez, (2014, Ene. 02). "Teoría de la programación orientada a objetos: ¿Qué es MVC?". Disponible en: <https://desarrolloweb.com/articulos/que-es-mvc.html>. Ultimo acceso: 19 de noviembre de 2020.
- [13] J.J Valdivia Caballero. (2016). "Modelo de procesos para el desarrollo del front-end de aplicaciones web," *Interfases*, pp. 187-208. Disponible en: <https://revistas.ulima.edu.pe/index.php/Interfases/article/view/1245>. Ultimo acceso: 12 de septiembre de 2020.
- [14] I. Alvarado Diaz, (2014, Abr. 12). "¿Qué es FrontEnd y BackEnd en la programación web?". Disponible en: <https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>. Ultimo acceso: 25 de febrero de 2021.
- [15] M. A. López Torrealba, (2015, Ene. 15). "Definición de aplicación web". Disponible en: <http://mialtoweb.es/definicion-de-aplicacion-web/>. Ultimo acceso: 25 de febrero de 2021.
- [16] R. Zepeda, (2015, Nov. 22). "¿Qué es xampp?". Disponible en: <http://blogdelinformatico-reizer.blogspot.com/2015/11/que-es-xampp.html>. Ultimo acceso: 16 de febrero de 2021.
- [17] R. Juárez, (2019, Oct. 09). "¿Qué es xampp y por qué lo utilizan los desarrolladores PHP?". Disponible en: <https://www.rbjuares.com/xampp-servidor-local-php-y-mysql/>. Ultimo acceso: 16 de febrero de 2021.
- [18] PHP (s.f). Disponible en: <https://www.ecured.cu/PHP>. Ultimo acceso: 02 de marzo de 2021.
- [19] J. Isacc, (2017, Jul. 11). "¿Qué es HTML, CSS y JavaScript?". Disponible en: <https://grupo4herramientasinformatica.blogspot.com/2017/07/que-es-html-css-y-javascript.html>. Ultimo acceso: 03 de marzo de 2021.
- [20] Á. Robledano, (2019, Ago. 12). "¿Qué es JavaScript?". Disponible en: <https://openwebinars.net/blog/que-es-javascript/>. Ultimo acceso: 04 de marzo de 2021.
- [21] P. Dominguez, (2018, Mar. 27). "¿Qué es jQuery?". Disponible en: <https://openclassrooms.com/en/courses/4309491-simplifica-tus-proyectos-con-jquery/4309498-que-es-jquery>. Ultimo acceso: 12 de marzo de 2021.

- [22] K. Lathiya, (2019, Nov. 01). "How to use AJAX in Laravel. Disponible en: <https://appdividend.com/2018/02/07/laravel-ajax-tutorial-example/>. Ultimo acceso: 22 de marzo de 2021.
- [23] A. E. Ortiz, (2018, Abr. 05). "¿Qué es un framework en informática o programación?". Disponible en: <https://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/>. Ultimo acceso: 10 de abril de 2021.
- [24] S. Bonito, (2017, Jun. 28). "¿Por qué Laravel?". Disponible en: <https://www.drauta.com/porque-laravel>. Ultimo acceso: 28 de febrero de 2021.
- [25] D. Palacios, (2017). "Curso de Laravel desde cero". Disponible en: <https://styde.net/laravel-5/>. Ultimo acceso: 08 de enero de 2021.
- [26] A. Schiaffarino, (2019, Mar. 12). "Modelo cliente servidor". Disponible en: <https://blog.infranetworking.com/modelo-cliente-servidor/>. Ultimo acceso: 15 de marzo de 2021.
- [27] M. Vega, (2010, Oct.). "Casos de uso UML". Disponible en: <https://lsi.ugr.es/~mvega/docis/casos%20de%20uso.pdf>. Ultimo acceso: 02 de diciembre de 2020.
- [28] Sistema de control de versiones (s.f). Disponible en: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>. Ultimo acceso: 05 de abril de 2021.