

UNIVERSIDAD PERUANA UNIÓN

FACULTAD DE INGENIERÍA Y ARQUITECTURA

Escuela Profesional de Ingeniería de Sistemas



Una Institución Adventista

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca

Tesis para obtener el Título Profesional de Ingeniero de Sistemas

Por:

Mirian Calcina Vilcapaza

Asesor:

Ing. Eder Gutierrez Quispe

Juliaca, febrero de 2022

DECLARACIÓN JURADA DE AUTORÍA DEL INFORME DE TESIS

Ing. Eder Gutierrez Quispe, de la Facultad de Ingeniería y Arquitectura, Escuela Profesional de Ingeniería de Sistemas, de la Universidad Peruana Unión.

DECLARO:

Que el presente informe de investigación titulado: **“Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca”** constituye la memoria que presenta el Bachiller **Mirian Calcina Vilcapaza** para obtener el título de Profesional de Ingeniero de Sistemas, cuya tesis ha sido realizada en la Universidad Peruana Unión bajo mi dirección.

Las opiniones y declaraciones en este informe son de entera responsabilidad del autor, sin comprometer a la institución.

Y estando de acuerdo, firmo la presente declaración en Juliaca, a los 25 días del mes de mayo del año 2022



Ing. Eder Gutierrez Quispe
Asesor



ACTA DE SUSTENTACIÓN DE TESIS

En Puno, Juliaca, Villa Chullunquiari, a 24 día(s) del mes de febrero del año 2022 siendo las 19:30 horas, se reunieron en el Salón de Grados y Títulos de la Universidad Peruana Unión, Filial Juliaca, bajo la dirección del Señor Presidente del jurado: Msc Freddy Abel Huanca Torres el secretario: Mg. Abel Angel Sullon Macalupu y los demás miembros: Mtro. Lenin Henry Lantigua Julca - Msc. Benigno Francis Herrera Lucra y el asesor Ing. Eder Gutierrez Quispe

con el propósito de administrar el acto académico de sustentación de la tesis titulada: Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NextJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca

de el(los)/la(las) bachiller(es): a) Mirian Lalcina Vilcapaza
 b) _____
 conducente a la obtención del título profesional de Ingeniera de Sistemas
 (Nombre del Título Profesional)

con mención en _____
 El Presidente inició el acto académico de sustentación invitando al (los)/a(la)/las) candidato(a)s hacer uso del tiempo determinado para su exposición. Concluida la exposición, el Presidente invitó a los demás miembros del jurado a efectuar las preguntas, y aclaraciones pertinentes, las cuales fueron absueltas por el(los)/la(las) candidato(a)s. Luego, se produjo un receso para las deliberaciones y la emisión del dictamen del jurado. Posteriormente, el jurado procedió a dejar constancia escrita sobre la evaluación en la presente acta, con el dictamen siguiente:

Candidato (a): Mirian Lalcina Vilcapaza

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	
<u>Aprobado</u>	<u>15</u>	<u>B-</u>	<u>Buena</u>	<u>Muy Buena</u>

Candidato (b): _____

CALIFICACIÓN	ESCALAS			Mérito
	Vigesimal	Literal	Cualitativa	

(*) Ver parte posterior

Finalmente, el Presidente del jurado invitó al(los)/a(la)/las) candidato(a)s a ponerse de pie, para recibir la evaluación final y concluir el acto académico de sustentación procediéndose a registrar las firmas respectivas.

[Firma]
 Presidente
[Firma]
 Asesor

[Firma]
 Miembro

[Firma]
 Secretario
[Firma]
 Miembro

 Candidato/a (a)

 Candidato/a (b)

DEDICATORIA

A Dios, por brindarme siempre sabiduría y fuerza necesaria para seguir adelante. A mi familia por ser una motivación para seguir adelante y su constante apoyo.

AGRADECIMIENTOS

A Dios por permitirme cumplir mis metas, gracias al estado peruano por darme la oportunidad de ser parte de PRONABEC (Programa Nacional de Becas y Créditos) - Beca 18 y gracias a mi compañero de investigación por brindarme todo su apoyo.

Gracias al Ing. Eder Gutierrez Quispe por compartir su experiencia profesional, y por acompañarme en esta investigación en calidad de Asesor.

ÍNDICE GENERAL

DEDICATORIA	iv
AGRADECIMIENTOS	v
ÍNDICE GENERAL	vi
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS	xii
ÍNDICE DE GRÁFICOS	xiv
ÍNDICE DE ANEXOS	xv
SÍMBOLOS USADOS	xvi
RESUMEN	xviii
ABSTRACT.....	xix
CAPÍTULO I. El problema.....	20
1.1. Identificación del problema	20
1.2. Justificación	21
1.3. Objetivos	23
1.3.1. Objetivo general.....	23
1.3.2. Objetivos específicos.....	23
CAPÍTULO II. Revisión de la literatura	24
2.1. Antecedentes de la investigación	24
2.2. Definición de marco teórico.....	26
2.2.1. Sistema web	26
2.2.2. Arquitectura de software	26
2.2.3. Cliente-Servidor (REST).....	27
2.2.4. Desarrollo del lado del servidor (Backend)	29
2.2.5. Desarrollo del lado del cliente (Frontend)	32
2.2.6. Ciclo de vida de desarrollo de software	40
2.2.7. Metodología en el desarrollo del software	43
CAPÍTULO III. Materiales y Métodos	47

3.1. Descripción del lugar de ejecución	47
3.2. Materiales e insumos.....	47
3.3. Tipo de investigación	48
3.4. Diseño de investigación	49
3.5. Metodología de la investigación	49
3.5.1. Planificación.....	50
3.5.2. Diseño	50
3.5.3. Codificación	50
3.5.4. Pruebas	52
3.6. Desarrollo del sistema.....	52
3.6.1. Planificación.....	52
3.6.2. Diseño	64
3.6.3. Desarrollo.....	70
3.6.4. Pruebas	95
3.6.5. Despliegue.....	103
CAPÍTULO IV. RESULTADOS Y DISCUSIÓN	105
4.1. Resultado del objetivo específico 1	105
4.1.1. Discusión.....	105
4.2. Resultado del objetivo específico 2	105
4.2.1. Discusión.....	107
4.3. Resultado del objetivo específico 3	107
4.3.1. Discusión.....	108
4.4. Resultado del objetivo específico 4	109
4.4.1. Análisis de tiempo de atención	109
4.4.2. Análisis de satisfacción de uso del sistema web	111
4.4.3. Discusión.....	115
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	116

5.1. Conclusiones	116
5.2. Recomendaciones.....	117
REFERENCIAS.....	118
ANEXOS	124

ÍNDICE DE FIGURAS

Figura 1. Mejores frameworks frontend (State of JS, 2020).....	26
Figura 2. Arquitectura Cliente – Servidor (Raj y Subramanian, 2018).	28
Figura 3. LoopBack (StrongLoop, 2020).	31
Figura 4. Presentación de Aplicación de una sola página (SPA) - (Frijlink, 2020).	33
Figura 5. Arquitectura de Vuex - (Vuex, 2020).	36
Figura 6. Ciclo de funcionamiento de Nuxt.js – (Nuxt.js, 2020).	38
Figura 7. Diseño anatómico – (Frost, 2018).	39
Figura 8. Metodología XP - Elaboración propia.	50
Figura 9. Arquitectura de solución de la investigación - Elaboración propia.	51
Figura 10. Login del sistema web - Elaboración propia.	65
Figura 11. Pantalla principal del sistema - Elaboración propia.	65
Figura 12. Gestión de equipos - Elaboración propia.	66
Figura 13. Vista de entrega de línea y/o equipo - Elaboración propia.	67
Figura 14. Asignación de línea y/o equipo - Elaboración propia.	67
Figura 15. Formato generado de entrega de una línea y/o equipo - Elaboración propia. ...	68
Figura 16. Diseño de modelo de la base de datos - Elaboración propia.	69
Figura 17. Diseño de la arquitectura del sistema web - Elaboración propia.	70
Figura 18. Interfaz de MongoDB Atlas – Elaboración propia.	71
Figura 19. Estructura de MongoDB - Elaboración propia.	72
Figura 20. Comandos disponibles en LoopBack - Elaboración propia.	72
Figura 21. Creación de una API REST – (StrongLoop, 2021).	73

Figura 22. Crear un modelo (lb4 model) - Elaboración propia.....	74
Figura 23. Conexión a base de datos (lb4 datasource) - Elaboración propia.....	75
Figura 24. Crear un repositorio (lb4 repository) - Elaboración propia.	76
Figura 25. Crear un controlador (lb4 controller) - Elaboración propia.....	76
Figura 26. Controlador de persona - Elaboración propia.....	77
Figura 27. Explorador de APIs de Loopback - Elaboración propia.....	78
Figura 28. Estructura de carpetas Nuxt.js - Elaboración propia.....	79
Figura 29. Componente Actions - Elaboración propia.....	80
Figura 30. Utilización de componente Actions - Elaboración propia.....	80
Figura 31. Mixin - Elaboración propia.....	82
Figura 32. Funciones (getItems(), saveItem(), deleteItem() clear() - Elaboración propia. .	83
Figura 33. Uso de mixin en componentes - Elaboración propia.	84
Figura 34. routing en Nuxt.js - Elaboración propia.	85
Figura 35. Archivo nuxt.config.js - Elaboración propia.	86
Figura 36. Archivo nuxt.config.js (Configuración de Nuxt Auth) - Elaboración propia. ...	87
Figura 37. Pantalla principal del sistema web – Elaboración propia.....	88
Figura 38. Administrar campaña - Elaboración propia.....	88
Figura 39. Módulo de gestión de persona - Elaboración propia.	89
Figura 40. Gestión de líneas - Elaboración propia.....	90
Figura 41. Agregar una línea - Elaboración propia.....	90
Figura 42. Agregar planes - Elaboración propia.	91
Figura 43. Gestión de equipos - Elaboración propia.....	92
Figura 44. Agregar modelo - Elaboración propia.	92

Figura 45. Nuevo equipo - Elaboración propia.....	93
Figura 46. Asignación de línea y/o equipo - Elaboración Propia.	94
Figura 47. Formato de asignación de línea y/o equipo - Elaboración Propia.	94
Figura 48. Dashboard - Pantalla principal del sistema web - Elaboración propia.	103
Figura 49. Visualización de reportes a detalle - Elaboración propia.	103
Figura 50. Diagrama de despliegue - Elaboración propia.	104
Figura 51. Arquitectura de sistema web – Elaboración propia.	107

ÍNDICE DE TABLAS

Tabla 1. Herramientas utilizadas - Elaboración propia.....	47
Tabla 2. Requerimientos funcionales - Elaboración propia.....	53
Tabla 3. Requerimientos no funcionales - Elaboración propia.....	53
Tabla 4. Roles del sistema web – Elaboración propia.....	53
Tabla 5. Nivel de Complejidad – Elaboración propia.....	54
Tabla 6. Historia de Usuario N° 1: Gestionar personas – Elaboración propia.....	54
Tabla 7. Historia de Usuario N° 2: Gestionar roles y permisos – Elaboración propia.....	55
Tabla 8. Historia de Usuario N° 3: Asociar un plan a una línea – Elaboración propia.....	55
Tabla 9. Historia de Usuario N° 4: Gestionar equipos – Elaboración propia.....	56
Tabla 10. Historia de Usuario N° 5: Gestionar personas – Elaboración propia.....	56
Tabla 11. Historia de Usuario N° 6: Asignación de línea y/o equipo–Elaboración propia	57
Tabla 12. Historia de Usuario N° 7: Gestionar las devoluciones – Elaboración propia.....	58
Tabla 13. Historia de Usuario N° 8: Gestionar pérdidas y robos - Elaboración propia.....	59
Tabla 14. Historia de Usuario N° 9: Mostrar estadísticas generales - Elaboración propia.	60
Tabla 15. Historia de Usuario N° 10: Gestionar venta de equipos – Elaboración propia...	60
Tabla 16. Historia de Usuario N° 11: Visualización de reportes – Elaboración propia.....	61
Tabla 17. Planificación de iteraciones – Elaboración propia.....	62
Tabla 18. Tareas a desarrollar en la Iteración 1 – Elaboración propia.....	62
Tabla 19. Tareas a desarrollar en la Iteración 2 – Elaboración propia.....	63
Tabla 20. Tareas a desarrollar en la Iteración 3,4,5,6,7- Elaboración propia.....	64
Tabla 21. Prueba de aceptación de la historia número 1 – Elaboración propia.....	95

Tabla 22. Prueba de aceptación de la historia número 2 – Elaboración propia.	96
Tabla 23. Prueba de aceptación de la historia número 3 – Elaboración propia.	96
Tabla 24. Prueba de aceptación de la historia número 4 – Elaboración propia.	97
Tabla 25. Prueba de aceptación de la historia número 5 – Elaboración propia.	98
Tabla 26. Prueba de aceptación de la historia número 6 – Elaboración propia.	99
Tabla 27. Prueba de aceptación de la historia número 7 – Elaboración propia.	100
Tabla 28. Prueba de aceptación de la historia número 8 – Elaboración propia.	101
Tabla 29. Prueba de aceptación de la historia número 9, 10 – Elaboración propia.	102
Tabla 30. Prototipos realizados - Elaboración propia.	106
Tabla 31. Atención actual en la entrega de equipo - Elaboración propia.....	109
Tabla 32. Atención actual de devolución de equipo – Elaboración propia.....	110
Tabla 33. Atención con el sistema web en la entrega de equipos-Elaboración propia.	110
Tabla 34. Atención con el sistema web - Elaboración propia.	111
Tabla 35. Resultados obtenidos con el sistema web - Elaboración propia.	111
Tabla 36. Pregunta N° 1 - La interfaz es amigable/intuitiva - Elaboración propia.	111
Tabla 37. Pregunta N° 2 - El tiempo de respuesta es óptimo - Elaboración propia.....	112
Tabla 38. Pregunta N° 3 - Me brinda información actualizada - Elaboración propia.....	113
Tabla 39. Pregunta N° 4 - Me ayuda a reducir el material físico - Elaboración propia....	113
Tabla 40. Pregunta N° 5 - Me ayuda a gestionar con facilidad - Elaboración propia.....	114

ÍNDICE DE GRÁFICOS

Gráfico 1. Análisis de satisfacción, Pregunta 1 - Elaboración propia.....	112
Gráfico 2. Análisis de satisfacción, Pregunta 2 - Elaboración propia.....	112
Gráfico 3. Análisis de satisfacción, Pregunta 3 - Elaboración propia.....	113
Gráfico 4. Análisis de satisfacción, Pregunta 4 - Elaboración propia.....	114
Gráfico 5. Análisis de satisfacción, Pregunta 5 - Elaboración propia.....	114

ÍNDICE DE ANEXOS

Anexo A. Fase de Planificación: Historia de usuario y criterios de aceptación.....	124
Anexo B. Diseño de prototipo del sistema web.	127
Anexo C. Código de Mixin – Nuxtjs	128
Anexo D. Archivo Excel, formatos utilizados para la gestión de celulares corporativos.	131
Anexo E. Formato encuestas.....	132
Anexo F. Encuesta de satisfacción del sistema web.....	134
Anexo G. Prueba de confiabilidad y validación de los instrumentos utilizados.	135

SÍMBOLOS USADOS

UI: Abreviatura inglés de Interfaz de usuario.

UX: Abreviatura en inglés de Experiencia de Usuario.

JSON: Abreviatura en inglés de Notación de Objetos de JavaScript.

BSON: Representa documentos JSON en formato Binario.

XML: Abreviatura en inglés de lenguaje de marcado extensible.

CSS: Abreviatura en in inglés de hojas en estilo cascada.

HTML: Abreviatura en inglés de lenguaje de marcas de hipertexto

SQL: Abreviatura en inglés de lenguaje de consulta estructurada.

NOSQL: Hace referencia al lenguaje de consulta no estructurado.

CLI: Abreviatura en inglés de interfaz de línea de comandados.

SPA: Aplicación de una sola página.

SSR: Renderización del Lado del Servidor o Aplicaciones universales.

HTTP: Abreviatura de Protocolo de transferencia de Hipertexto

INEI: Instituto Nacional de Estadística e Informática

XP: Abreviatura inglés de Programación Extrema.

PHP: Siglas en inglés Hypertext Pre-Processor, Lenguaje de Programación Interpretado.

ISO: Abreviatura en inglés de Organización Internacional de Normalización.

API: Abreviatura en inglés de Interfaz de Programación de Aplicaciones.

URI: Abreviatura en inglés de Identificador Uniforme de Recursos.

URL: Abreviatura en inglés de Localizador de Recursos Uniforme.

NPM: Siglas en inglés Node Package Manager, gestor de paquetes.

SOAP: Abreviatura en inglés de Protocolo Simple de Acceso a Objetos.

DOM: Abreviatura en inglés de Modelo de Objetos del documento.

RUP: Abreviatura en inglés de Proceso de Desarrollo Unificado.

ORM: Abreviatura en inglés de Mapeo Objeto Relacional (Object Relational Mapping).

ODM: Abreviatura en inglés Mapeo de Documentos de Objetos (Object Document Mapping).

UPeU: Universidad Peruana Unión

DTI: Dirección de Tecnologías de Información

RESUMEN

En la actualidad el desarrollo y la implementación de sistemas web a medida se han convertido en la base tecnológica de las empresas. Las empresas se apoyan cada vez más en adoptar un sistema que automatice, mejore y permita simplificar sus procesos y tareas. Sustituye procedimientos manuales en el cual se invierte mucho tiempo. En la Universidad Peruana Unión, campus Juliaca, la gestión y control de inventario de equipos, asignación de equipos, devoluciones, y registro de pérdidas se registran de manera manual, apoyado en archivos Excel, genera un flujo de información poco confiable, inoportuno e inexacto, que llevan a cometer errores o acciones tardías. Motivo por el cual se planteó desarrollar un sistema web utilizando el Framework LoopBack 4 y NuxtJS, para esta investigación se optó por una metodología ágil, Programación Extrema (XP), con el fin de analizar, cada detalle del negocio de la empresa, diseñar, y desarrollar el sistema web. Para el desarrollo se utilizó full stack JavaScript, no hubo necesidad de cambiar entre diferentes lenguajes de programación. En el lado del servidor, Framework LoopBack 4, para la construcción de las APIs, y consumirlas desde un sistema web. Para la construcción de la interfaz de usuario Nuxt.js, vuesax en el lado cliente; los datos se gestionan mediante el gestor de base de datos MongoDB, que utiliza el formato JSON, formato utilizado por los desarrolladores de JavaScript, ya que es fácil de transformar los datos y trabajar con ellos. Al desarrollar bajo la tecnología de código abierto se reduce el tiempo, existen menores costos de desarrollo, no hay tarifas de licencia y mantenimiento, el cual permite completar el trabajo más rápido, usando módulos y complementos con menos errores y la capacidad de actualizar en un futuro.

Se concluyó con el desarrollo del sistema web que permite gestionar y controlar el inventario de equipos celulares, asignación de equipos, devoluciones, registro de pérdidas, permitiendo realizar un trabajo más eficiente, ordenada, y reduciendo el riesgo de pérdida de información, ya que esta, esta almacenada en una base de datos que permite centralizar toda la información.

Palabras clave: Sistema web, LoopBack, Nuxt.js, MongoDB, Metodología XP.

ABSTRACT

At present the development and implementation of custom web systems have become the technological base of companies. Companies are increasingly relying on adopting a system that automates, improves and simplifies their processes and tasks. It replaces manual procedures in which a lot of time is invested. At the Universidad Peruana Unión, Juliaca campus, the management and control of equipment inventory, equipment allocation, returns, and loss records are recorded manually, supported by Excel files, generating an unreliable, inopportune and inaccurate flow of information, which lead to mistakes or late actions. Reason why it was proposed to develop a web system using the LoopBack Framework 4 and NuxtJS, for this research an agile methodology was chosen, Extreme Programming (XP), in order to analyze, every detail of the business of the company, design, and develop the web system. Full stack JavaScript was used for development, there was no need to switch between different programming languages. On the server side, LoopBack 4 Framework, for building APIs, and consuming them from a web system. For the construction of the Nuxt.js user interface, vuesax on the client side; the data is managed by the MongoDB database manager, which uses the JSON format, a format used by JavaScript developers, as it is easy to transform and work with the data. By developing under open source technology, time is reduced, there are lower development costs, there are no license and maintenance fees, which allows to complete the work faster, using modules and plugins with less errors and the ability to update in a future.

It was concluded with the development of the web system that allows to manage and control the inventory of cellular equipment, allocation of equipment, returns, registration of losses, allowing a more efficient, orderly work, and reducing the risk of loss of information, since this, is stored in a database that allows you to centralize all the information.

Keywords: Web system, LoopBack, Nuxt.js, MongoDB, XP Methodology.

CAPÍTULO I. El problema

1.1. Identificación del problema

En la actualidad, el desarrollo de sistemas de información viene creciendo, ya que con su implementación permite administrar, gestionar, procesar, almacenar y mostrar datos relevantes para cada organización que ayuda en la toma de decisiones, en este sentido, un sistema de información se caracteriza principalmente por la eficiencia que procesa los datos en relación al área de acción (Barrios, 2019). En el Perú se tiene el problema de la demora de trámites, en especial con el Estado, provocando el malestar de los usuarios ocasionándoles pérdida de tiempo al momento de realizar algún trámite, según el Banco Interamericano de Desarrollo (BID). En nuestro país se tiene el peor promedio en la parte de tramites con el estado (RPP Noticias, 2019), por lo que nace la necesidad de que la gestión pública pueda ser digitalizada para mejorar en la atención al usuario.

Los problemas de gestión no solamente se presentan en el estado, según (Amao, 2016) en su investigación encontró deficiencias en una empresa comercial y de servicios, como la pérdida de equipos informáticos, el registro de forma manual, falta de control del ingreso y salida de los productos, información desactualizada y la demora en la obtención de la búsqueda de los productos debido a la aglomeración de los documentos, ocasionando el malestar de los clientes por la falta de eficiencia en la atención. Mencionamos también otra investigación, según (Aiquipa, 2018), realizada a una empresa naviera dedicada al transporte marítimo que contaba con equipos informáticos, identificó deficiencias con los equipos informáticos que no habían pasado mantenimiento en su tiempo programado lo que generaba a futuro incidencias, no se contaba con un inventario actualizado e información centralizada de todas las sedes debido a que se trabajaba con el registro de ficheros Excel lo que generaba lentitud en la administración. Revisando estos casos nació la necesidad de poder mejorar en esos aspectos.

Según (Ylquimiche, 2018), en su informe de suficiencia profesional acerca del manejo y gestión de los laboratorios de cómputo de la Facultad de Ciencias Empresariales de la Universidad Peruana Unión Campus Lima, menciona que, para todos esos procesos y tareas, se realizaban informes mediante documentos, adjuntando el formato impreso de tipo MS Excel proporcionado por el área, a su vez hace mención que el área no cuenta con un sistema especializado para que el personal pueda usarlo para llevar a cabo el control de este proceso;

solo existen formatos de apoyo y archivadores para almacenar los cargos de entrada y salida de equipos, señalando así que, al trabajar de esa manera se tiende a cometer errores como la redundancia de datos en el registro de las características del equipo y provocando la falta de información rápida del estado de los equipos, afirmando que la atención del personal no es efectiva y propone que se implemente un sistema control de inventario para mejorar las necesidades del personal.

Actualmente en la Universidad Peruana Unión en el área de Sub-Dirección de Tecnologías de Información (DTI) se realiza el servicio de préstamo de equipos de celulares corporativos para cada cargo, el cual, desde la adquisición, el inventariado, la asignación al personal y posterior venta, se registran de manera manual, utilizando el programa de Microsoft Excel y archivos de documentos, Anexo D, lo que conlleva a no tener un buen control del inventario e inversión de mayor tiempo, creando duplicidad innecesaria y redundancia de datos a la hora de registrar, y al momento de que la administración solicite reportes de los equipos asignados por área y por cargo, el estado de los equipos, la disponibilidad de equipos y líneas, entre otros, no se cuentan a la mano, por lo cual realizar los reportes solicitados toman su tiempo para su preparación ya que la información que se maneja no se tiene centralizada ni automatizada, puesto que se tiene que revisar archivadores físicos para su elaboración, y al contar con varios lugares donde registrar los datos, la información es inconsistente y toma mayor tiempo a la hora de tener la información actualizada. Por lo que nace la necesidad de contar con un sistema web que pueda ser utilizado para llevar un mejor control del proceso de gestión de celulares corporativos, facilite el trabajo y ayude a mejorar la atención al usuario.

1.2. Justificación

La presente investigación se realiza con la finalidad de aportar al conocimiento sobre la gestión de equipos móviles con el desarrollo de un sistema web mediante la aplicación de una metodología y uso de nuevas herramientas tecnológicas de desarrollo para el control del inventario, se espera que los resultados obtenidos puedan ayudar a similares trabajos de investigación, ya que está demostrado que la implementación del sistemas web en la gestión de inventarios mejora la eficiencia en su administración, según definiciones de (Microsoft, 2020) nos menciona que “El software de gestión de inventarios automatiza tareas que solían requerir mucho tiempo, como el proceso manual de contabilizar los productos uno a uno y

registrarlos en papel. Gracias a la digitalización de este proceso, no solo se ha conseguido mayor rigurosidad, sino también un valioso ahorro de tiempo”. Desde este punto de vista ya nos afirma que se estaría mejorando en ese proceso que se busca desarrollar.

El uso de metodologías para el proceso de desarrollo de sistemas añade valor en la calidad del producto, tal es el caso de la metodología XP, una metodología que promueve el trabajo colaborativo en equipo, constante aprendizaje de los desarrolladores y la retroalimentación frecuente del código (Maida & Pacienza, 2015). El uso de esta metodología permite que el área Subdirección de Tecnologías de Información (Cliente) se involucre en el proceso del desarrollo de manera constante con el objetivo de contemplar y validar los requerimientos funcionales y no funcionales establecidos del sistema web.

El sistema web es desarrollado bajo la arquitectura de cliente-servidor, el cual provee una clara separación de funciones con base en la idea de servicio, donde el cliente es el consumidor de servicios y el servidor, la de proveer el servicio, gestionando el acceso a los datos y realizando funciones de control y seguridad. Esta arquitectura permite distribuir físicamente los procesos y los datos en forma más eficiente, (Márquez & Zulaica, 2004). Se desarrolla con el lenguaje de programación JavaScript full stack, con el cual no es necesario cambiar entre diferentes lenguajes de programación. Para la construcción de las APIs, se eligió el Framework LoopBack en su versión 4, el cual nos facilita la creación de APIs dinámicas mediante la línea de comandos, esto hace que crear modelos y conectarlos a una o varias bases de datos sea increíblemente sencillo. Para la construcción de la interfaz de usuario se escogió Nuxt.js que está basado en Vue.js. Se tomó en cuenta básicamente porque es un Framework completo, posee varios módulos incluidos como la autenticación básica, oauth2 y con redes sociales, el cual nos provee una estructura de proyecto bien definida, permite la contratación de aplicaciones universales y SPA, y es muy flexible para incluir cualquier librería de componentes para la interfaz; en general está diseñado para construir aplicaciones de gran escala, y agilizar y automatizar algunos procesos en el desarrollo Frontend. Además, se logra crear componentes reutilizables, esto permitirá que el tiempo del desarrollo de los siguientes módulos sean más cortos, de manera que no se dupliquen innecesariamente.

El desarrollo del sistema web permitirá a la Universidad Peruana Unión, específicamente al área de Sub dirección de Tecnologías de la Información a sistematizar la gestión y control de teléfonos celulares corporativos, permitiendo a los usuarios (Coordinador de Servicios TI, Operador de Soporte Técnico y Mesa de Ayuda) a que pueda realizar un trabajo eficiente,

atención más rápida y oportuna, al momento de realizar la renovación o adquisición, en el registro y control del inventario, de igual manera a gestionar la asignación de préstamos, devoluciones y contar con reportes actualizados. Al contar con el sistema web la información estará centralizada y unificada para su uso fácil, optimizar el proceso, disminuir errores, optimizar las tareas, reducir el tiempo para disponer de ello, lo que significa que le proporcionará un acceso más rápido a la información requerida, gracias a la centralización de la información, y evitar la pérdida de la información.

1.3. Objetivos

1.3.1. Objetivo general.

Desarrollar un sistema web utilizando el Framework LoopBack 4 y Nuxt.js para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca.

1.3.2. Objetivos específicos.

- Analizar los requerimientos para el desarrollo del sistema web para la gestión y control de celulares corporativos.
- Realizar diseño de prototipo y arquitectura del sistema web para la gestión y control de celulares corporativos.
- Construir el sistema web para la gestión y control de celulares corporativos usando el Framework LoopBack 4 y Nuxt.js.
- Mostrar la mejora en la gestión y control de los equipos celulares corporativos mediante el sistema web.

CAPÍTULO II. Revisión de la literatura

2.1. Antecedentes de la investigación

Cenas (2018), en su Tesis “Sistema informático web para el control del proceso de alquiler de GPS en la empresa HENPHONE S.R.L.” tiene como objetivo desarrollar un sistema informático web, aplicando la metodología Programación extrema (XP). Por el lado del servidor utilizó el lenguaje de programación PHP y JavaScript, HTML5, y CSS3 en el lado del cliente, aplicando un diseño responsivo. Como resultados se concluyó con el desarrollo del sistema, realizando las pruebas de funcionamiento se verifica que el sistema de información ayuda en gran medida a reducir el tiempo, principalmente en los reportes y consultas de la información, permitiendo un control más ordenado y eficaz, los datos que antes eran almacenados en archivos Excel ahora se centralizan en una base de datos consistente y concurrente que contiene información actualizada en todo momento y de esta manera se aborda la problemática de la empresa, con ello, aporta valor en las mejoras de los procesos y demuestra la necesidad de contar con su sistema en la empresa.

En la tesis realizado por Falcón (2017) con el título de estudio, “Desarrollo de un aplicativo web para la gestión de equipos informáticos en el área de soporte técnico de informática – INEI Huánuco”. Desarrolló e implementó un aplicativo web usando tecnologías web para agilizar la gestión de equipos informáticos en el área de soporte técnico de informática – INEI Huánuco, utilizando la metodología de Programación Extrema (XP) y las tecnologías web (herramientas Netbeans, XAMPP, el lenguaje de programación PHP, HTML5, CSS3 y JavaScript) en el desarrollo de sistemas de información basados en web para agilizar procesos operativos e integrarlos. Concluyó con el desarrollo del aplicativo web, ayudando a reducir el tiempo empleado en, en un 71% al momento de registrar un equipo informático, un 42% el tiempo empleado en generar el reporte de inventarios, también se redujo el tiempo empleado en generar el reporte de mantenimientos en un 62% y de un 43% en generar el reporte de movimientos, agilizando de esta manera la gestión de equipos informáticos.

Otra investigación realizada por (Tacilla, 2018), en su Tesis “Sistema informático web de gestión de incidencias usando el framework AngularJS y NodeJS para la empresa REDTEAM SOFTWARE LLC”. El objetivo, es brindar una solución mediante el desarrollo de un sistema de información web utilizando los Frameworks Angular.js y Node.js, realizando el análisis de la situación actual del proceso de gestión de incidencias con la finalidad de identificar los

requerimientos principales, el cual, es la base para el desarrollo del sistema, aplicando la metodología ICONIX, permitiéndole tener un desarrollo ágil y rápido mostrando el análisis, diseño e implementación del sistema de información web. Con la implementación del sistema web logró reducir el tiempo en dar solución a las incidencias reportadas de 129.46 horas (100%) a 69.83 horas (53.93%), logrando aumentar el porcentaje de atención de incidencias y el aumento del nivel de satisfacción del cliente.

De Liz (2020) en su investigación titulada “Aplicación web progresiva para un red social de aficionados a los videojuegos”, tiene como objetivo el desarrollo de la aplicación web para ello, se realizó un análisis inicial, mediante encuestas a posibles futuros usuarios y el análisis competitivo de otras redes sociales con la intención de obtener los requisitos iniciales, en segundo lugar, se llevó a cabo la fase de diseño, arquitectura Baas (Backend as a Service) el cual consiste en implementar el backend de la aplicación utilizando un proveedor externo, a través de la utilización de su API, que hace de puente entre el frontend, y en tercer lugar se realizó el desarrollo de la aplicación, frontend, comenzando con la creación del proyecto con el framework Nuxt.js y su integración con el proveedor externo de backend, seguido de la implementación del modelo de datos con Vuex y la interfaz de usuario con Vue y Vuetify. Concluyendo satisfactoriamente con la puesta en producción de la aplicación.

Elegir un framework basada en Node.js para su aplicación puede ser difícil, hay tantos de ellos, con sus propias fortalezas y debilidades, pero al fin del día, la elección que realizará dependerá de su entorno de programación y de lo que quiera que haga su aplicación. El primer framework que generalmente viene a la mente es Express. Si desea la libertad de elegir sus propias librerías e interacciones de terceros, LoopBack, es el segundo Framework API REST más popular, según Github (Shipton, 2021). Ganó el premio API 2019 en la categoría “Mejor en API middleware” en API World 2019 de conferencias y exposición sobre API y microservicios más grande del mundo (Mundo API, 2019) (Whiteley, 2019). Basado en Express que le permite crear rápidamente API y microservicios. El desarrollo de API REST y microservicios con Express es posible, pero después de un cierto nivel de complejidad, puede convertirse en un ejercicio repetitivo plagado de errores para cada nuevo proyecto. El uso de un marco de API REST como LoopBack reduce el tiempo de desarrollo y reduce el dolor de cabeza de mantenimiento. Según State of JS (2020), Svelte, React y

Vue.js son los principales framework usados para el desarrollo frontend, ver figura 1. Nuxt.js, un framework frontend basado en Vue.js.

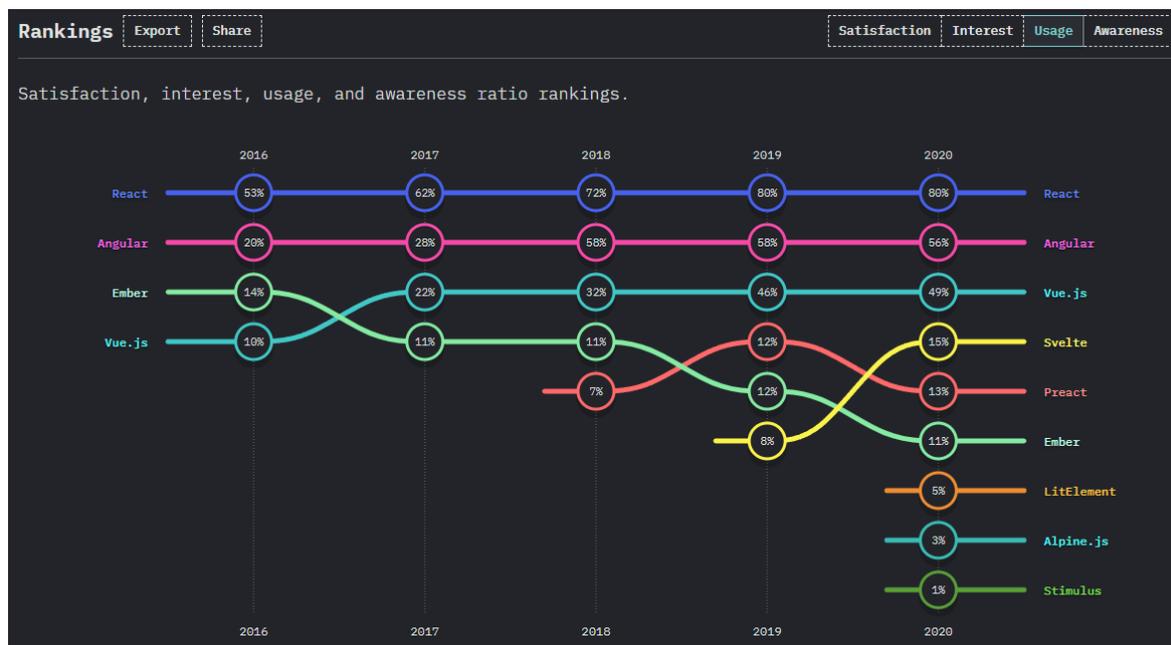


Figura 1. Mejores frameworks frontend (State of JS, 2020)

2.2. Definición de marco teórico

2.2.1. Sistema web

Según Mora (2002), un sistema web “es una aplicación web desarrollada para satisfacer necesidades específicas y resolver problemas mediante el análisis de la lógica de los procesos o también llamado "lógica de negocios", automatizando procesos en línea. Una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP). En las aplicaciones web suelen distinguirse tres niveles (como en la arquitectura cliente/servidor): el nivel superior que interacciona con el usuario (el cliente web, normalmente un navegador), el nivel inferior que proporciona los datos (la base de datos) y el nivel intermedio que procesa los datos (el servidor web)”.

2.2.2. Arquitectura de software

Según (Bass, Clements, & Kazman, 2013) define, “La arquitectura de software de un programa o de un sistema computacional es la estructura del sistema, que abarca componentes

de software, las características externamente visibles de esos componentes, y las relaciones entre ellas”. Por otro lado, (Garlan & Shaw, 1996) declararon que “la arquitectura de software implica la descripción de los elementos de los que se compone el sistema, las interacciones entre esos elementos, los patrones que guían su composición y las restricciones en esos patrones”.

La arquitectura de software es el conjunto de decisiones de diseño importantes para organizar el software y promover los atributos de calidad deseados y otras propiedades (Keeling, 2017). En cuanto a las decisiones de diseño se define cuáles van a guiar; por otro lado, tenemos los atributos de calidad, describen propiedades externas visibles de un sistema de software, permite definir qué es lo que va más allá de lo funcional que va garantizar que el sistema funcione. Así mismo la arquitectura promueve otras propiedades. Por ejemplo, la arquitectura correcta para el trabajo, le permitirá enviar a tiempo, dentro del presupuesto y sin requerir demasiadas horas extras.

2.2.3. Cliente-Servidor (REST)

Transferencia de estado representacional (REST) término introducido por Fielding (2000) en su tesis doctoral, define a “REST es un intento de mostrar cómo debe comportarse una aplicación Web bien diseñada: una red de páginas Web (una máquina de estados virtual) donde el usuario progresará seleccionando enlaces (transiciones de estado) que devuelven la página siguiente (el siguiente estado de la máquina) que el usuario manipula a su gusto”.

(Subramanian & Raj, 2019) REST es un estilo arquitectónico de software que facilita pautas para que los sistemas distribuidos se comuniquen directamente utilizando los principios y protocolos existentes de la web. Se utiliza principalmente para desarrollar servicios web livianos, rápidos, escalables y fáciles de mantener y API (Interfaz de programación de aplicaciones) (Kumar, 2018). Para REST el principio fundamental para la comunicación de datos es utilizar protocolo.

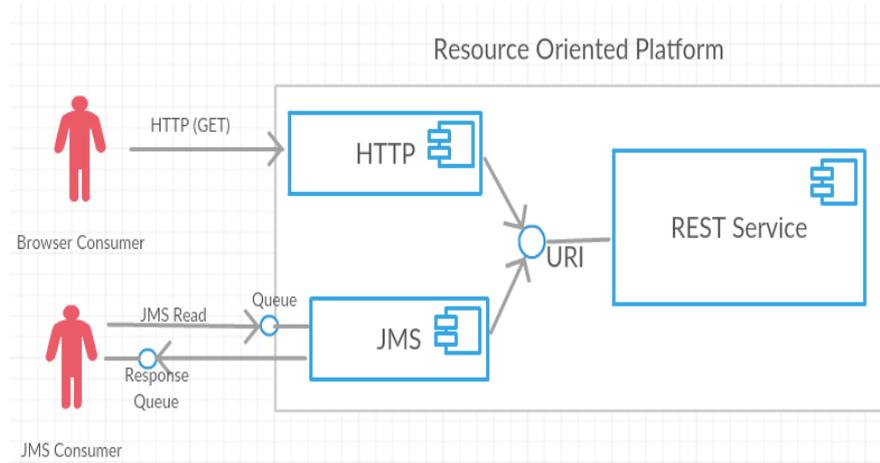


Figura 2. Arquitectura Cliente – Servidor (Raj y Subramanian, 2018).

La arquitectura REST provee el acceso a recursos para que el cliente acceda y represente en el lado del cliente. En estilo REST, URI o IDs globales ayudan a identificar cada recurso. REST cuenta con un conjunto de limitaciones arquitectónicas que tratan de minimizar las comunicaciones de red y la latencia, al mismo tiempo, la escalabilidad de las implementaciones de los componentes y la independencia. Describe seis principios:

- **Cliente-servidor:** requiere que un servicio ofrezca una o más operaciones, y que los servicios esperen que los clientes soliciten dichas operaciones.
- **Sin estado:** requiere que no tengan estado las comunicaciones entre el consumidor del servicio (cliente) y el proveedor del servicio (servidor).
- **Caché:** requiere se etiqueten claramente como con caché o sin caché las respuestas.
- **Interfaz uniforme:** requiere que compartan una sola interfaz común para todas las operaciones todos los proveedores del servicio y consumidores dentro de una arquitectura que cumpla con REST
- **Sistema en capas:** requiere la capacidad de agregar o quitar intermediarios en el tiempo de ejecución sin alterar el sistema.
- **Código según la demanda (opcional):** permite que la lógica dentro de los clientes se actualice independientemente de la lógica del servidor utilizando un código ejecutable enviado por los proveedores del servicio a los consumidores.

2.2.3.1.HTTP.

HTTP es la base de cualquier intercambio de datos en la web, con operaciones de solicitud/respuesta. El cliente establece una conexión con un servidor y envía un mensaje con

los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan (Mora, 2015).

HTTP tiene métodos de solicitud que permiten la comunicación entre múltiples aplicaciones. Los más utilizados son:

- **GET:** Recupera información sobre el recurso de objeto.
- **POST:** Crea un recurso de objeto.
- **PUT:** Actualiza la información de un objeto específico.
- **DELETE:** Elimina un objeto específico.

2.2.3.2. SOCKET.

Un protocolo de capa de aplicación diseñado para facilitar la comunicación bidireccional (el cliente o el servidor pueden enviar un mensaje a la otra parte siempre que haya un mensaje disponible) y la comunicación full-duplex (tanto el cliente como el servidor pueden enviarse mensajes entre sí simultáneamente) entre el lado del servidor y el lado del cliente en tiempo real (Prusty, 2016), basada en HTTP para su protocolo de enlace inicial. Socket es un protocolo binario, basado en texto y diseñado para facilitar la comunicación bidireccional

2.2.4. Desarrollo del lado del servidor (Backend)

Interfaz de programación de aplicaciones (API) es un conjunto de especificaciones que permiten que un programa informático interactúe con otro. Un cliente solicita algunos datos, esos datos viajan a una aplicación de servidor web a través del Protocolo HTTP, la aplicación de servidor web toma las solicitudes y procesa datos, luego los datos se envían al cliente a través de HTTP (Scott, 2020). Las API proporcionan una interfaz para que los datos fluyan desde una base de datos a las aplicaciones, es una forma de describir la forma en que los programas o los sitios webs intercambian datos, el formato de intercambio de datos normalmente es JSON o XML.

2.2.4.1. Node.js.

Node.js es entorno de ejecución de código abierto y multiplataforma para ejecutar código JavaScript. Según la documentación oficial de Node.js (2020) la define de la siguiente manera: “Node.js es un entorno de ejecución para JavaScript construido con el motor JavaScript V8 de Chrome”. Significa que para transformar el texto JavaScript en código ejecutable para que la computadora lo procese, utiliza V8, el motor JavaScript de código abierto creado por Google para alimentar los navegadores Chromium y Chrome.

Node.js tiene un ecosistema de paquetes en NPM (Node Package Manager), es el ecosistema abierto más grande del mundo, con más de 350,000 paquetes. Le permite usar el mismo lenguaje de programación para el frontend y el backend, con la ayuda de los frameworks Node.js como Express, LoopBack y entre otros, usar diferentes herramientas, consultar diferentes pautas y también recomendar prácticas que finalmente le ahorrarán mucho tiempo.

2.2.4.1.1. Express

Express, un marco de aplicación web, que proporciona un conjunto robusto de características para construir aplicaciones web híbridas y de una sola página o múltiples páginas. Express fue lanzado por primera vez en 2009 por TJ Holowaychuk (Holmes & Harber, 2019), describe Express como un framework web inspirado en Sinatra, que es un framework web basado en Ruby (Brown, 2019).

Según la documentación oficial la define “es un marco de aplicación web Node.js mínimo y flexible que proporciona un conjunto robusto de características para aplicaciones web y móviles. Con una gran cantidad de métodos de utilidad HTTP y middleware a su disposición, crear una API.” (StrongLoop, Express, 2020)

2.2.4.1.2. LoopBack

“LoopBack es un framework Node.js de código abierto altamente extensible basado en Express que le permite crear rápidamente API y microservicios compuestos de sistemas de fondo como bases de datos y servicios SOAP o REST” (StrongLoop, 2020). Viene con un conjunto de módulos Node.js que se pueden usar de forma independiente y en conjunto para construir API REST para aplicaciones frontend (web y móviles). LoopBack tiene la posibilidad de conectarse a múltiples fuentes de datos, escribir lógica de negocios en typescript. En la

figura 2 se aprecia cómo LoopBack sirve como un puente de composición entre las solicitudes entrantes y las integraciones salientes. Del mismo modo, muestra las diferentes personas que están interesadas en varias capacidades proporcionadas por LoopBack.

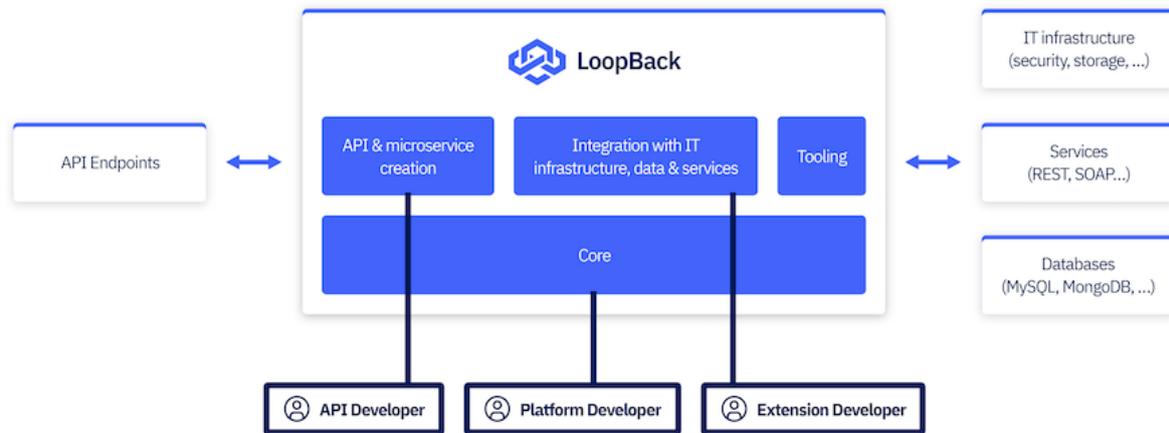


Figura 3. LoopBack (StrongLoop, 2020).

Ventajas:

- Crear API REST de extremo a extremo con poca o ninguna codificación mediante la línea de comandos.
- Acceder a datos de las bases de datos y servicios como MongoDB, SOAP y API REST.
- Incorporar relaciones de modelo y controles de acceso para API complejas.
- Componentes separables para almacenamiento de archivos, inicio de sesión de terceros y OAuth.

2.2.4.2. Base de datos.

Jiménez (2018) define como “una colección de datos relacionados, que guarda grandes cantidades de información de forma organizada, y que posteriormente se pueda encontrar y utilizar de manera fácil”.

2.2.4.2.1. MongoDB

MongoDB, una base de datos de código abierto, de almacenamiento de documentos NoSQL escrita en C ++. MongoDB usa JavaScript como lenguaje de consulta. Desarrollado por MongoDB Inc., de uso gratuito. Los datos se almacenan en forma de colecciones. Cada colección consta de documentos. Esto significa que los campos pueden variar de un

documento a otro y que los modelos de datos pueden evolucionar con el tiempo en respuesta a los requisitos cambiantes de la aplicación (Holque, 2018). Según Giamas (2017) las características clave de MongoDB:

- **Base de datos de uso general:** a diferencia de otras bases de datos NoSQL que se crean para fines específicos (por ejemplo, base de datos de gráfico), mongoDB puede servir cargas heterogéneas y múltiples propósitos dentro de una aplicación.
- **Diseño de esquema flexible:** los enfoques orientados a documentos con atributos no definidos que se pueden modificar sobre la marcha son un contraste clave entre MongoDB y las bases de datos relacionales.
- **Rico en funciones:** al ofrecer la gama completa de operadores equivalentes de SQL, junto con características como MapReduce, marco de agregación, Time to Live y colecciones limitadas, e indexación secundaria, mongoDB puede adaptarse a muchos casos de uso, sin importar cuán diversos sean los requisitos.
- **Escalabilidad y equilibrio de carga:** está construido a escala, tanto verticalmente como lo más importante horizontalmente. Mediante el uso de fragmentos, un arquitecto puede compartir la carga entre diferentes instancias y lograr escalabilidad de lectura y escritura.
- **Replicación nativa,** los datos se replicarán en un conjunto de réplicas sin una configuración complicada.
- Objetos JSON (BSON, BSON JSON) para almacenar y transmitir documentos. JSON se usa ampliamente en la web para la comunicación frontend y API y, como tal, es más fácil cuando la base de datos usa el mismo protocolo.
- **MapReduce,** a pesar de que el motor MapReduce no es tan avanzado como lo es en frameworks dedicados, sin embargo, es una gran herramienta para construir canalizaciones de datos.

2.2.5. Desarrollo del lado del cliente (Frontend)

2.2.5.1. Aplicación de una sola página (SPA).

Es una aplicación web de una sola página, es decir, la interacción de la aplicación es en una página. En una SPA el HTML, JavaScript, CSS se carga una sola vez. SPA está compuesta de componentes individuales que pueden reemplazarse o actualizarse de forma independiente, sin

actualizar o volver a cargar toda la página, no es necesario volver a cargar toda la página en cada acción del usuario. (Jadhav, Balkrishna, & Anushree, 2015, pág. 2877).

En un SPA, las vistas no son páginas HTML completas, son simplemente porciones del DOM (Modelo de Objetos del documento) que conforman las áreas visibles de la pantalla. Si se necesita una nueva vista, se genera localmente en el navegador y se adjunta dinámicamente al DOM a través de JavaScript. Sin embargo, una aplicación web tradicional, cada nueva vista (página HTML) se construye en el servidor, es decir, cada solicitud de una nueva vista da como resultado un viaje de ida y vuelta al servidor. Cuando se necesita datos nuevos en el lado del cliente, la solicitud se envía al lado del servidor, el servidor las construye en cada petición. (Scott, 2020). El formato de intercambio de datos suele ser la notación de objetos JavaScript (JSON).

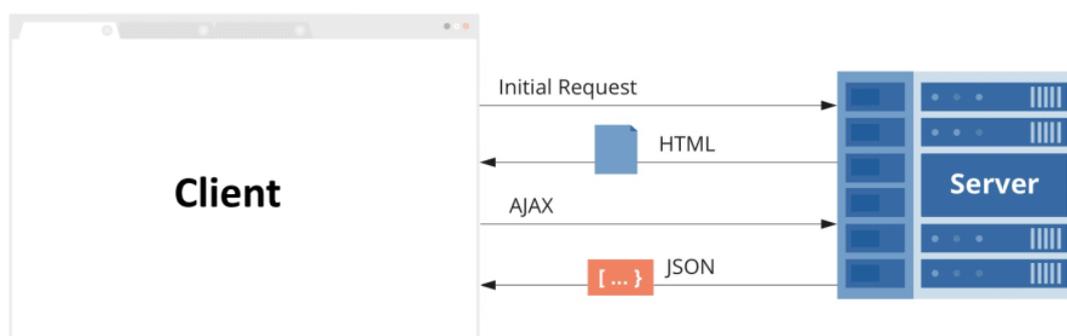


Figura 4. Presentación de Aplicación de una sola página (SPA) - (Frijlink, 2020).

2.2.5.2. JavaScript.

JavaScript, lenguaje de programación diseñado para crear aplicaciones web interactivas, es una de las columnas principales del desarrollo web. El primer prototipo de JavaScript fue escrito por Brendan Eich en solo 10 días, se creó por primera vez para el navegador web Netscape en 1995 (Shute, 2019).

La popularidad de JavaScript está creciendo, la creación de aplicaciones web con JavaScript como único lenguaje de programación. Hay muchas tecnologías para elegir para el frontend y el backend al desarrollar una aplicación. Del lado de backend tenemos: Express.js, LoopBack, Meteor.js y entre otros, del mismo modo, para el frontend, las tecnologías incluyen Vue.js, React, Angular, Backbone y muchas más (Sharma, 2018).

Los frameworks web de JavaScript permiten el desarrollo ágil y rápido de aplicaciones web. Sirven de estructura para SPA, permiten a los desarrolladores preocuparse menos de la estructura del código y el mantenimiento, y centrarse en la funcionalidad. El desarrollo es más escalable, la curva de aprendizaje es relativamente fácil ya que están basadas en JavaScript, del mismo modo, permite agregar diferentes animaciones a un sitio web para que sea agradable de usar, cada frameworks puede ser utilizado dependiendo a la necesidad.

2.2.5.2.1. *React*

React es una biblioteca de JavaScript para crear interfaces de usuario basado en componentes (Facebook Inc., 2020). Esto significa que podemos construir una interfaz de usuario componiendo elementos llamados componentes. Un componente es un elemento que contribuye a construir una interfaz de usuario; puede ser un cuadro de texto, un botón, un formulario completo, un grupo de componentes, etc. Incluso toda la interfaz de usuario de la aplicación es un componente (Chiarelli, 2018). Crear componentes en React permite dividir la interfaz de usuario en piezas reutilizables e independientes, lo que hace que el componente de su aplicación sea reutilizable (Singh & Tanna, 2018).

2.2.5.2.2. *Vue.js*

Vue es un framework progresivo para construir interfaces de usuario, diseñado desde cero para ser gradualmente adoptable. La biblioteca principal se centra solo en la capa de vista y es fácil de recoger e integrar con otras bibliotecas o proyectos existentes. Vue también es perfectamente capaz de impulsar aplicaciones sofisticadas de una sola página (You, Vue.js, 2020).

Vue fue diseñado para crear interfaces de usuario basadas en componentes, la idea es que puede estructurar su aplicación en torno a elementos independientes y reutilizables que representan una parte discreta de la interfaz (Franklin, Wanyoike, & Bouchefra, 2019). A diferencia de React o Angular, uno de los beneficios de Vue es la salida HTML limpia que produce. Otras bibliotecas de JavaScript tienden a dejar el HTML disperso con atributos y clases adicionales en el código, mientras que Vue los elimina para producir una salida semántica limpia (Street, Passaglia, & Halliday, 2018).

La biblioteca principal de Vue.js se enfoca solo en la capa de vista. Con el tiempo, a medida que crecen los requisitos, puede adaptar bibliotecas adicionales para la funcionalidad y la idea

de agregar características a la aplicación que está creando con el tiempo no limita el uso de herramientas de desarrollo más complejas. ¿Necesita agregar un enrutador? No hay problema; use el Vue-Router, una opción de un tercero, o utilice el suyo propio. ¿Buscas gestionar el estado de la memoria? Puede usar un Objeto JavaScript simple, un patrón de tienda o el Vuex específico de Vue.js (Brett, 2018).

A. Vue Router

Un router es la parte que sincroniza la vista que se muestra actualmente con el contenido de la barra de direcciones del navegador, es decir, es el componente de una aplicación web moderna que activa la actualización de la vista cuando se produce un cambio en la dirección URL (Copes, 2021). Vue Router es el enrutador oficial de Vue. Se integra profundamente con el núcleo de Vue para facilitar la creación de aplicaciones de una sola página con Vue (You, Vue.js, 2020).

Con Vue.js, ya estamos componiendo nuestra aplicación con componentes, al agregar Vue Router, todo lo que tenemos que hacer es asignar nuestros componentes a las rutas. Algunas de sus características son: ruta anidada / mapeo de vista, ver los efectos de transición impulsados por el sistema de transición de Vue.js, enlaces con clases de CSS activas automáticas, comportamiento de desplazamiento personalizable y parámetros de ruta, consulta, comodines.

B. Vuex

Según (Vuex, 2020) define como “patrón de gestión de estado + biblioteca para aplicaciones Vue.js. Sirve como un almacén centralizado para todos los componentes de una aplicación, con reglas que aseguran que el estado sólo pueda mutar de manera predecible”.

- Un patrón de gestión del estado, es una aplicación autónoma con las siguientes partes:
- El estado, la fuente de verdad que impulsa la aplicación
- La vista, un mapeo declarativo del estado
- Las acciones, las posibles formas en que el estado podría cambiar en reacción a las entradas del usuario desde la vista.

De acuerdo a (Vuex, 2020), la arquitectura empleada por Vuex tiene el siguiente flujo: los componentes pueden desencadenar acciones, que a su vez pueden o no realizar solicitudes a las API en el backend (opcionalmente dependiendo de la aplicación). Tal acción puede comprometer datos, lo que llamará una mutación. En este momento, el evento se inicia en Vue DevTools y se puede observar. Finalmente, la mutación almacena efectivamente los datos en el estado. Se puede acceder a este estado a través de un getter sin ningún componente de la aplicación (Lima, Petrucelli, & Espíritu Santo, 2019). El resumen de esta arquitectura se puede ver en la siguiente figura 5.

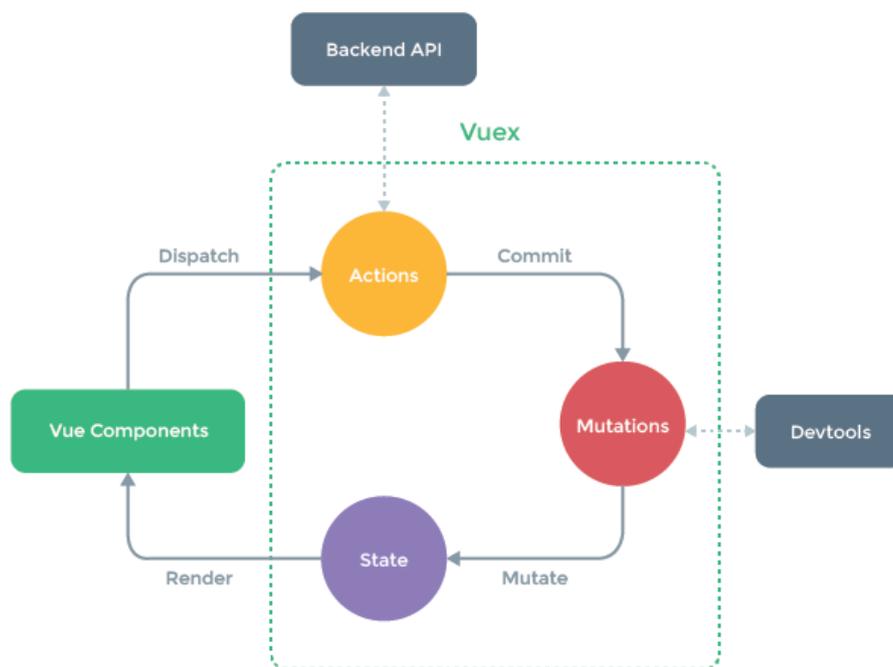


Figura 5. Arquitectura de Vuex - (Vuex, 2020).

Según Schmitz y Georgii (2016), una forma sencilla de entender Vuex dentro de un proyecto es a través de su concepto de Store. Vuex sirve como una "tienda" de datos para su aplicación, permitiendo que todos los componentes tengan acceso a un estado particular ya que todos los estados están centralizados dentro de esa tienda.

- **Store**, es donde se almacena el estado de la aplicación. Cuando decimos estado, estamos hablando de un grupo de docenas de variables para almacenar datos. Como sabemos, Vue tiene una capa de visualización reactiva que observa variables y cambia su contenido en consecuencia. Store en Vuex es donde se almacenan las variables, donde se almacenan los estados y donde la Vista observará los cambios.

- **Actions**, definido como la acción que puede cambiar un estado, es lo único que puede cambiar la información de un estado, significa que en lugar de cambiar store directamente, llamará a una Acción para cambiarla.
- **Mutations**, se describe como el "evento que cambia store". La acción cambiará store a través de "mutaciones". Solo las mutaciones deben acceder directamente a las variables de store.
- **Getters**, son métodos responsables de observar las variables en store y proporcionarle la aplicación. Asegura que los cambios en store se reflejarán en la aplicación.

C. Nuxt.js

Nuxt.js es un framework progresivo basado en Vue.js para crear aplicaciones web modernas. Se basa en las bibliotecas oficiales de Vue.js (vue, vue-router y vuex). El objetivo de Nuxt es hacer que el desarrollo web sea potente y eficaz con una gran experiencia de desarrollador en mente (Nuxt.js, 2020). Diseñado para darle una arquitectura sólida siguiendo las pautas oficiales de Vue. Adoptable de forma incremental, se puede utilizar para crear desde páginas de destino estáticas hasta complejas aplicaciones web listas para empresas.

Con Nuxt.js puede crear una (generada estática o universal, o de una sola página) aplicación web de forma rápida con la misma configuración y sin acumulación de configuración para escribir. Cuando un usuario visita una aplicación Nuxt.js o navega a una de sus páginas a través de <nuxt-link>, tiene un ciclo de funcionamiento muy específico (Nuxt.js, 2020). Veamos en la figura 6:

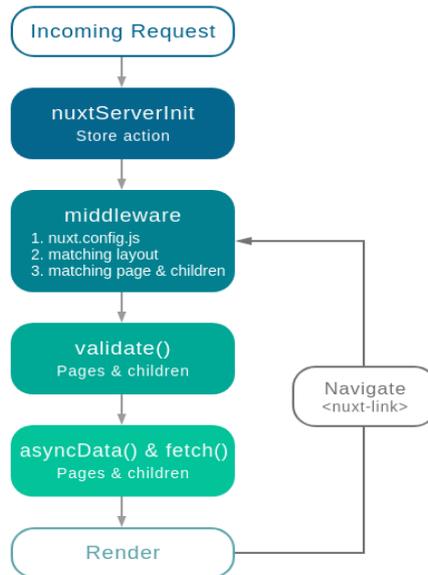


Figura 6. Ciclo de funcionamiento de Nuxt.js – (Nuxt.js, 2020).

1. El usuario visita inicialmente la aplicación, si `nuxtServerInit` se define en store, Nuxt.js la llamará y actualizará store.
2. Seguidamente, ejecuta cualquier middleware existente para la página que visita. Nuxt comprueba `nuxt.config.js` primero el archivo para el middleware global, luego comprueba el archivo `matching layout` correspondiente (para la página solicitada) y finalmente comprueba `matching page & children` para el middleware. El middleware tiene prioridad en ese orden.
3. Si la ruta que se visita es una ruta dinámica, y `validate()` existe un método para ello, la ruta se valida.
4. Luego, Nuxt.js llama a los métodos `asyncData()` y `fetch()` para cargar datos antes de mostrar la página. El método `asyncData()` se utiliza para obtener datos y representarlos en el lado del servidor, mientras que el método `fetch()` se utiliza para llenar store antes de representar la página.
5. Finalmente, se representa la página que contiene todos los datos adecuados.

Ventajas:

- Crear aplicaciones web universales más fácilmente, es una de las mayores ventajas y ¿Qué es una aplicación universal? Se usa para describir el código JS que puede ejecutarse tanto en el lado del servidor como en el lado del cliente.
- Configuración a través de la línea de comando con la plantilla de inicio.

- Obtenga división automática de código (páginas preprocesadas) y una excelente estructura de proyecto por defecto.
- Configuración de transiciones entre las rutas y escribir componentes de un solo archivo.
- Potente sistema de enrutamiento con datos asíncronos.
- Servicio de archivos estáticos.

2.2.5.3. *Experiencia de usuario (UI/UX).*

2.2.5.3.1. *Desarrollo basado en componentes*

Hay (2018) afirma que “no estamos diseñando páginas, estamos diseñando sistemas de componentes”. A medida que el desarrollo web y el diseño avanzan se tiene la necesidad de crear componentes que sean útiles para desarrollar sistemas de diseño, en lugar de crear colecciones simples de páginas web. El diseño atómico es una metodología que explica cómo aplicar un modelo mental para crear sistemas de diseño de forma eficiente, bajo un concepto de diseñar por componentes. El diseño atómico consiste en átomos, moléculas, organismos, plantillas y páginas que trabajan simultáneamente para crear sistemas de diseño (Frost, 2018).

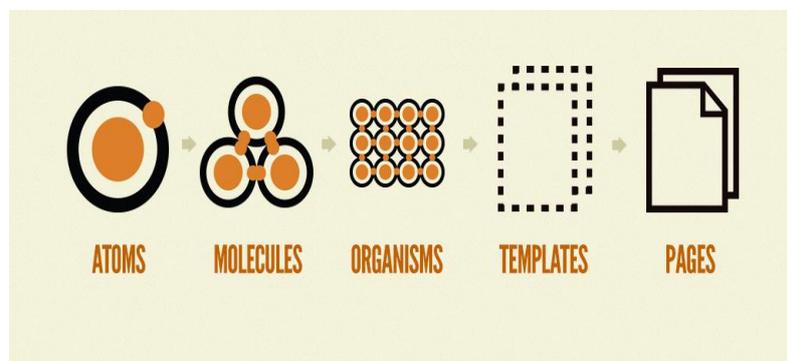


Figura 7. Diseño anatómico – (Frost, 2018).

- **Átomos** son los bloques de construcción básicos. Aplicados a las interfaces web, estos átomos incluyen elementos HTML básicos como etiquetas de formulario, entradas, botones y otros que no se pueden desglosar sin dejar de ser funcionales.
- **Moléculas** grupos de átomos unidos para ser las unidades fundamentales más pequeñas. Ahora una etiqueta de formulario, una entrada de búsqueda y un botón pueden unirse para crear una molécula de formulario de búsqueda, esta unión puede realizar una acción.

- **Organismos** son grupos de moléculas unidas entre sí desde una sección relativamente compleja y distinta de una interfaz, la molécula del formulario de búsqueda se combina con una molécula de navegación para formar un organismo de encabezado. Los organismos como los encabezados de sitios web consisten en moléculas más pequeñas como navegación primaria, formularios de búsqueda, navegación de utilidad y logotipos.
- **Plantillas** consisten principalmente en grupos de organismos unidos para formar páginas. Es aquí donde comenzamos a ver el diseño que se une y el diseño en acción.
- **Páginas** son instancias específicas de plantillas. Aquí, el contenido del marcador de posición se reemplaza con contenido representativo real para dar una descripción precisa de lo que un usuario finalmente verá e interactúa. Las páginas son esenciales ya que es donde se aprueba la eficiencia del diseño, y donde finalmente todos esos componentes se unen para formar una interfaz de usuario (UI) funcional.

El diseño atómico permite ver todo, desde lo abstracto hasta los resultados más concretos, y ver simultáneamente cómo esos elementos se unen para formar UI finales.

2.2.5.3.2. *Vuesax*.

Vuesax es un framework de componentes basado en Vue.js, diseñado desde cero para ser adoptable de forma incremental que facilita el desarrollo frontend y agiliza el trabajo con gran calidad visual. Se centra en facilitar el desarrollo de aplicaciones, mejorando el diseño sin eliminar la funcionalidad necesaria (Vuesax, 2020).

2.2.5.3.3. *Vuetify*.

Vuetify es un framework de componentes, su objetivo es proporcionar componentes limpios, semánticos y reutilizables que faciliten la creación de su aplicación. Vuetify utiliza el patrón de diseño Material Design de Google, tomando señales de otros marcos populares como Materialize.css, Material Design Lite, UI semántica y Bootstrap 4 (Vuetify, 2020).

2.2.6. **Ciclo de vida de desarrollo de software**

Según la ISO (International Organization for Standardization), norma 12207 define como “un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo,

la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso”.

El ciclo de vida de un software por lo general, incluye varias fases, desde el análisis de desarrollo preliminar hasta las pruebas y evaluaciones de software posteriores al desarrollo (Leau, Loo, & Tham, 2012). Todas estas actividades se llevan a cabo de diferentes maneras, según las necesidades. En cada una de las etapas de un modelo de ciclo de vida, se pueden establecer una serie de objetivos, tarea y actividades (Cantone, 2006).

2.2.6.1. Análisis.

En la primera fase del ciclo de vida del desarrollo de software es analizar y definir los requerimientos funcionales y no funciones del sistema, determinar las necesidades y características que deberá cubrir el sistema e identificar el valor que el sistema proporciona, el usuario describe minuciosamente y en detalle las funciones que desea que tenga dicho sistema. Esta primera fase es de gran importancia ya que es la base de todo el proyecto (Cantone, 2006) y (Chacón, Ruiz , & Mendoza, 2014).

2.2.6.2. Diseño.

En esta fase se utiliza la información recopilada en la primera fase para realizar el diseño, el diseño del software es de vital importancia, se especifica la estructura de todo el software; los datos, la arquitectura, interfaces y entre otros. Un buen diseño ayudara en la fase de desarrollo del producto final. Es de vital importancia verificar exhaustivamente cada uno de los requerimientos planteados ya que son la base para diseñar el software (Chacón et al., 2014). Según Pressman (2010) se pueden dividir en dos grandes procesos las actividades del diseño:

- **Diseño preliminar:** llamado también diseño de sistema, diseño arquitectónico o diseño de alto nivel. Está describe la arquitectura general, identifica sus componentes y su organización, y relaciones en el sistema. Se identifican los módulos en los que puede dividirse el sistema considerando la manera en la que se implementaran.
- **Diseño detallado:** Se centra en la lógica interna de los módulos, se refina la representación arquitectónica que lleva a una estructura detallada de datos, componentes e interfaces del software.

2.2.6.3. Desarrollo.

En la fase de desarrollo, es traducir el diseño del software escribiendo el código fuente del software requerido. Se realiza basándose en el análisis y diseño de los requerimientos, dicha programación se procede a desarrollar de acuerdo a la arquitectura elegida (Chacón et al., 2014).

2.2.6.4. Pruebas.

La etapa de prueba es para garantizar que el marco y sus segmentos individuales distintivos satisfagan las necesidades del arreglo particular realizado durante el arreglo del plan, a pesar de la mejora que contiene una medida mínima de errores concebibles. La idea es reconocer a tiempo los errores que podrían haberse cometido en la estructura y la organización de la mejora, es considerablemente menos costoso descubrir los problemas antes de que el marco se transmita a los clientes (Chacón et al., 2014).

El camino hacia la prueba y la identificación de errores se puede ajustar de varias maneras, dependiendo de la circunstancia única y el período del ciclo del producto en el que se encuentra la tarea (Chacón et al., 2014):

- **Pruebas de unidad:** se utilizan para confirmar el funcionamiento correcto de un segmento particular en el marco. Se sugiere que a medida que se agreguen nuevos aspectos destacados a las aplicaciones.
- **Pruebas de integración:** se ejecuta para controlar que el procedimiento consienta los requisitos previos del producto, son fundamentales para cuando necesita probar la conciliación de los diversos segmentos que conforman un marco, se utilizan para identificar errores en sus interfaces.
- **Pruebas de aceptación:** cuando se completa una prueba de reconocimiento, se realiza una prueba de reconocimiento, si la calidad se pasa efectivamente, el final del procedimiento de mejora y el inicio de la organización de soporte se demuestran con autoridad.

2.2.6.5. Implementación.

En esta fase se debe realizar la instalación y despliegue del mismo, para comenzar el uso formal del proyecto, realizar la conversión de datos y el cambio de sistema (la carga de la

información existente al nuevo sistema) y finalmente realizar una capacitación a los usuarios finales (Chacón et al., 2014).

2.2.6.6. *Mantenimiento.*

Esta fase, que tiene lugar después de la entrega del producto final, en donde se asegura que el sistema esté funcionando y adaptándose a nuevos requerimientos. Por lo general, consiste en introducir los ajustes necesarios para mejorar el rendimiento y corregir los problemas que puedan surgir (Chacón et al., 2014).

2.2.7. Metodología en el desarrollo del software

Según (Tinoco, Rosales, & Salas, 2010) Caracterizan una estrategia como una variedad de estrategias, procedimientos, dispositivos y archivos auxiliares que ayudan a los diseñadores de programación en sus esfuerzos por ejecutar nuevos marcos de datos. Las filosofías fuerzan un procedimiento restringido en la mejora de la programación para que sea cada vez más sorprendente y eficaz.

2.2.7.1. *Metodología ágil.*

Según (Navarro, Fernández, & Morales, 2013) define como “Las metodologías ágiles son flexibles, pueden ser modificadas para que se ajusten a la realidad de cada equipo y proyecto”. Se orientan a la obtención de satisfacción del cliente y rápida en resultados, se tiende a la simplicidad y la mejora continua del desarrollo del proyecto con entregas tempranas, continuas y con valor. Las principales metodologías ágiles son: Scrum, eXtreme Programming (XP) y Kanban

Las metodologías tradicionales son planteadas para el uso exhaustivo de la documentación durante todo el ciclo del desarrollo del proyecto, en cambio, las metodologías ágiles ponen una gran importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y mantener una buena relación con el cliente (Crespo, Peña, Pascual, & Fustiel, 2016).

2.2.7.1.1. Metodología XP

La programación extrema (denominada XP en términos ágiles) es una metodología ágil para el desarrollo de software, centrada en potenciar las relaciones interpersonales como

clave para el éxito del desarrollo de software, promueve el trabajo en equipo estableciendo un buen clima de trabajo. La filosofía de XP es satisfacer al completo las necesidades del cliente (Borja, s.f.). Desarrollado en 1996, por Kent Beck, Ward Cunningham y Ron Jeffries (Bahit, 2012).

Según Beck (2000) “XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas.” XP se apoya en cinco valores: comunicación, simplicidad, retroalimentación, valentía y respeto, los cuales enfatizan la esencia colaborativa del equipo (Borja, s.f.) y (Bahit, 2012).

A. Las historias de usuario.

Las historias de usuarios en XP son la técnica utilizada para especificar los requerimientos del software, las características que el sistema debe poseer, sean requerimientos funcionales o no funcionales. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla. Estos documentos no son estáticos, sino que pueden ser modificados, reemplazados o destruidos (Borja, s.f.).

B. Roles XP

En (Borja, s.f.) de acuerdo con la propuesta original de Beck los roles son:

- **Programador:** escribe las pruebas unitarias y produce el código del sistema.
- **Cliente:** escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- **Encargado de pruebas (Tester):** ayuda al cliente a escribir las pruebas funcionales, ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.
- **Encargado de seguimiento (Tracker)** Proporciona retroalimentación al equipo, verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado para mejorar futuras estimaciones, y realiza el seguimiento del proceso de cada iteración.

- **Entrenador (Coach):** es responsable del proceso global y proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el procedimiento correctamente.
- **Consultor:** es un miembro externo al equipo que posee el conocimiento técnico necesario para poder ayudar al equipo con determinados problemas
- **Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Se trata al cliente como parte del equipo de desarrolladores, cuando se utiliza la metodología, lo que evita la redundancia de información, y permite que se realicen pruebas constantes. Otro beneficio es que se efectúan cambios mientras avanza el desarrollo según los requerimientos solicitados evitando pérdidas de tiempo (Bustamante & Rodríguez, 2014).

C. Proceso XP

El ciclo de vida de los proyectos bajo XP puede verse como una sucesión de definiciones por parte del cliente, aceptación y continuo desarrollo de parte de los programadores. La diferencia con otras metodologías, es que estas sucesiones ocurren en muy corto tiempo (Beck, 2000).

Planeación

La planeación es la etapa inicial, empezando con la actividad de recabar requerimientos que permitan que los miembros del equipo comprendan el contexto del negocio. El cliente plantea las “historias”, historias de usuario que describe las características y funcionalidad del software que se va a desarrollar. Cada historia de usuario descrita tiene una prioridad con base en el valor de la funcionalidad del negocio.

Después los miembros del equipo evalúan cada historia, se estima que la historia requiere más de tres semanas de desarrollo, y que pueda descomponerse en historias más chicas y de nuevo se asigna un valor. Es importante observar que en cualquier momento es posible escribir nuevas historias. A medida que avanza el trabajo, el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas. Entonces, el equipo reconsidera todas las entregas faltantes y modifica sus planes en consecuencia (Bustamante & Rodríguez, 2014).

Las historias de usuarios en XP son la técnica utilizada para especificar los requerimientos ya sean requerimientos funcionales o no funcionales y las características que el sistema debe poseer. Cada historia de usuario debe ser suficientemente comprensible y delimitada para que los programadores puedan implementarla (Borja, s.f.).

Diseño

El proceso de diseño debe procurar diseños simples y sencillos para facilitar el desarrollo, siguiendo el principio de “utilizar el diseño más sencillo que consiga que todo funcione”. Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe. Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo del diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto (Bustamante & Rodríguez,

Codificación

En esta fase se lleva acabo la programación en pareja, la unidad de pruebas, y la integración (Bustamante & Rodríguez, 2014). Se desarrolla funcionalidades a partir de las historias de usuario y las iteraciones planeadas, así como mejoras del diseño. La codificación se realiza bajo estándares establecidos para mantener el código consistente facilitando su comprensión y escalabilidad. Para XP es necesario crear una comunicación fluida y directa entre el cliente y el equipo de trabajo en todas las fases, de este modo se reduce el tiempo de desarrollo.

Pruebas

Las pruebas son un elemento clave. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas, de modo que puedan ejecutarse en repetidas veces y con facilidad. A medida que se organizan las pruebas unitarias individuales en un “grupo de prueba universal”, las pruebas de la integración y validación del sistema pueden efectuarse a diario. Esto da al equipo XP una indicación continua del avance y también lanza señales de alerta si las cosas marchan mal. Wells dice: “Corregir pequeños problemas cada cierto número de horas toma menos tiempo que resolver problemas enormes justo antes del plazo final.” (Pressman, 2010, pág. 65). Las pruebas de aceptación, llamadas también pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios (Bustamante & Rodríguez, 2014).

CAPÍTULO III. Materiales y Métodos

3.1. Descripción del lugar de ejecución

El presente proyecto es ejecutado en el área de Dirección de Tecnologías de Información, en la oficina de Soporte y Mesa de Ayuda, cuya ubicación es en la Universidad Peruana Unión Campus Juliaca.

3.2. Materiales e insumos

Hardware

- ✓ Laptop HP Core i5
- ✓ Dispositivo Móvil Android

Software

Se describen todas las herramientas tecnológicas utilizadas para el cumplimiento del desarrollo del sistema web.

Tabla 1. *Herramientas utilizadas - Elaboración propia.*

Clasificación	Nombre de la herramienta	Descripción
Programas	Visual Studio Code	Editor de código fuente. Para la construcción del backend y frontend del proyecto.
	Postman	Entorno para realizar pruebas en las rutas que provee API REST.
	Enterprise Architect	Utilizado para el diseño y modelado de la base de datos.
	Adobe XD	Herramienta de diseño digital basada en vectores para sitios web y aplicaciones. Utilizada para diseñar el prototipo del sistema web.
Lenguaje de programación	JavaScript/Node.js	Lenguaje de programación usado para construir Backend y frontend del proyecto.
Base de datos	MongoDB	Motor de base de datos usado para el almacenamiento.

	LoopBack 4	Framework altamente escalable de Node.js, desarrollado sobre Express.js, facilita la creación de APIs REST para clientes (aplicaciones móviles, de escritorio, web, etc) así como microservicios. Se usó para toda la construcción de las APIs.
Frameworks	Vue.js	Se usó para la construcción de interfaz de usuario, el cual nos proporcionó toda la configuración necesaria para la transpiración de Javascript, el procesamiento de CSS.
	Nuxt.js	Framework progresivo basado en Vue.js para crear aplicaciones web modernas.
	Vuesax	Framework de componentes UI creado con Vue.js, para crear aplicaciones web. Se utilizó para la creación de la vista.
	Tailwind CSS	Un marco CSS para crear interfaces de usuario personalizadas, se utilizó para la maquetación del sistema web.
Herramienta para el control de versiones	GitHub	Herramienta colaborativa que ayudó a llevar un registro y control de cualquier cambio sobre el código del proyecto. Y poder trabajar sobre los últimos cambios realizados en el proyecto.
	Netlify	Plataforma todo en uno para automatizar proyectos web modernos, ofrece servicios de alojamiento y backend sin servidor para aplicaciones web y sitios web estáticos. Funcionará conectando su repositorio de GitHub para extraer el código fuente de su sitio web. Se utilizó para desplegar la aplicación de Frontend.
Plataformas	IBM Cloud	Plataforma de computación en la nube, completa y segura, engloba ambientes desde nube pública hasta multicloud híbrida de nueva generación, recursos avanzados de datos, inteligencia artificial. Plataforma para desplegar la aplicación de Backend
	MongoDB Atlas	Servicio MongoDB alojado en la nube en AWS, Azure y Google Cloud.
Herramienta para gestión de tareas	Asana	Herramienta de colaboración en equipo en línea que se especializa en la gestión del flujo de trabajo, flexible y elegante que puede adaptar a su voluntad.
	Slack	Herramienta de comunicación en equipo.

3.3. Tipo de investigación

Según (Tam, Vera, & Oliveros, 2008) la investigación aplicada “tiene como objetivo crear nueva tecnología a partir de los conocimientos adquiridos a través de la investigación

estratégica para determinar si estos pueden ser aplicados con o sin mayor refinamiento para los propósitos definidos. La información obtenida a través de este tipo de investigación debería ser también aplicable en cualquier lugar y por lo tanto ofrece oportunidades significativas para su difusión”. Por tanto, este proyecto es una investigación aplicada, busca resolver el problema de gestión y control de celulares corporativos que se realiza de manera manual apoyado en archivos de Excel y formatos lo que conlleva a la pérdida de la información, enfocado en la búsqueda y aplicación del conocimiento, con el Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS.

3.4. Diseño de investigación

Según (Bernal, 2010) “en la investigación descriptiva, se muestran narran, reseñan o identifican hechos, situaciones, rasgos, características de un objeto de estudio, o se diseñan productos, modelos prototipos, guías, etc., pero no se dan explicaciones o razones del porqué de las situaciones, hechos, fenómenos, etc., la investigación descriptiva se guía de las preguntas de investigación que se formula el investigador, se soporta en técnicas como la encuesta, entrevista, observación y revisión documental”. Se tomará el diseño de investigación descriptiva, ya que se acomoda a lo que se quiere investigar como la mejora en la gestión y control de celulares corporativos de la UPeU – campus Juliaca, a través del desarrollo de un sistema web.

3.5. Metodología de la investigación

Según lo conceptualizado se eligió la metodología XP donde se realiza la adaptación para resolver el problema, se hace el uso de las actividades necesarias para cada fase de la metodología, esta personalización se realizó de acuerdo a las necesidades para el desarrollo de la solución.

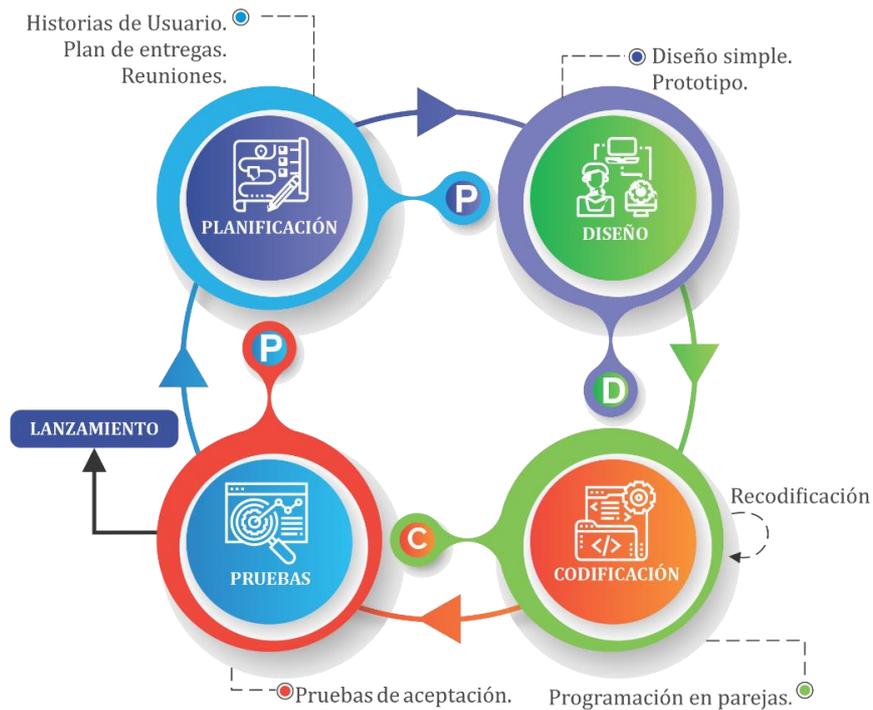


Figura 8. Metodología XP - Elaboración propia.

3.5.1. Planificación

En esta fase inicial se realizó el análisis de la situación actual, básicamente el levantamiento de la información, se identificaron los requerimientos funcionales y no funcionales del sistema, a partir de ello, se creó las historias de usuario que describen la salida necesaria, características y funcionalidad del sistema que se construyó.

3.5.2. Diseño

En esta fase se definió un diseño simple y sencillo para facilitar el proceso de la codificación, se realizó el diseño de la base de datos, el diseño, el diseño del prototipo y el diseño de la arquitectura tecnológica que se usará para el desarrollo, considerando que este diseño sea fácilmente entendible e implementable.

3.5.3. Codificación

En la siguiente fase consistió en la codificación del sistema, se procedió a transformar los requerimientos o historias de usuario definidos para cada iteración, especificados detalladamente la función que debe cumplir cada requerimiento solicitado. En el desarrollo se usaron prácticas de la metodología ágil Programación Extrema: se consideró que el cliente se

encuentre presente ya que es indispensable, programación en parejas, diseño simple, propiedad colectiva del código e integración continua, desarrollo bajo una herramienta colaborativa, lo que conlleva a un registro y control de cualquier tipo de cambio sobre el código.

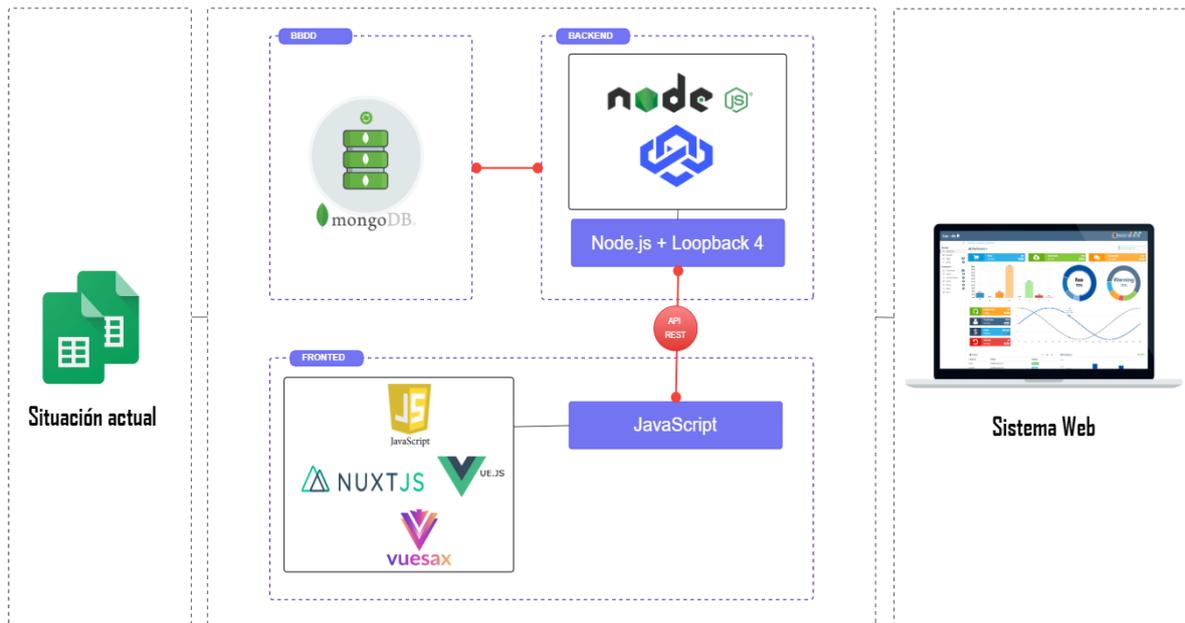


Figura 9. Arquitectura de solución de la investigación - Elaboración propia.

En la figura 9 se observa la arquitectura de solución del proyecto, consta de 3 partes. La primera parte representa la situación actual, en el cual todos los datos se encuentran almacenados en un archivo Excel. El segundo representa la interacción entre backend y frontend para el desarrollo del sistema web, así mismo muestra la tecnología utilizada y la última parte representa el sistema web con el cual cada usuario tendrá interacción.

Para el almacenamiento de los datos se eligió MongoDB, una base de datos NoSQL. El cual nos permitió centralizar los datos, hacer que se encuentre disponible oportunamente, esto a su vez constituye una importante herramienta para el tratamiento de datos, permitiendo ordenar los datos de manera que estén relacionados y estructurados acorde a las especificaciones que se planteó. Para la construcción de las APIs, se eligió el Framework LoopBack en su versión 4 porque facilita la creación de APIs dinámicas mediante la línea de comandos. Y finalmente para la construcción de la interfaz de usuario se escogió Nuxt.js que está basado en Vue.js que te provee una estructura de proyecto bien definida, permite la contratación de aplicaciones universales y SPA, flexible para incluir cualquier librería de

componentes para la interfaz y diseñado para construir aplicaciones para agilizar y automatizar algunos procesos en el desarrollo Frontend.

3.5.4. Pruebas

En esta fase se realizó las pruebas de aceptación, llamadas también pruebas del cliente, especificaciones del cliente que se centran en las características y funcionalidad del sistema que son visibles y revisables por parte del cliente. Fase en que se integró y validó el sistema que se desarrolló, las pruebas se validaron con la presencia de los clientes quienes definen si la prueba fue satisfactoria o deficiente.

3.6. Desarrollo del sistema

3.6.1. Planificación

3.6.1.1. Análisis de requerimientos.

En esta fase se realizó la recopilación de información necesaria, donde se encontró los siguientes documentos que registran las actividades realizadas para el control y gestión de equipos celulares.

- ✓ Política de uso
- ✓ Formato de asignación de equipo celular: En donde se detalla la línea, las características y accesorios del equipo asignado según el cargo correspondiente.
- ✓ Formato de entrega de equipos celulares: Devoluciones, en ella se detalla el estado del equipo a devolver y el motivo.
- ✓ Registro de líneas y equipos: Archivo Excel, en el cual se guarda la cantidad tanto de líneas y equipos, áreas y cargos, registro de asignación, devoluciones y planes de la línea.

A partir de la recopilación de la información y sistemas web explorados previamente, se determinaron los requerimientos funcionales y no funcionales del sistema web, donde los requerimientos funcionales fueron descritos a través de las historias de usuarios.

Tabla 2. *Requerimientos funcionales - Elaboración propia.*

Descripción	
RF01	Gestionar roles y permisos para dar acceso al usuario al sistema.
RF02	Gestionar personas
RF03	Gestionar líneas
RF04	Gestionar equipos
RF05	Asignación de líneas
RF06	Entrega de líneas y/o equipos
RF07	Gestionar devoluciones
RF08	Gestionar equipos perdidos y robados
RF09	Mostrar estadísticas generales sobre actividades y/o recursos del sistema.
RF10	Visualización de reportes.
RF11	Gestionar la venta de equipos

Tabla 3. *Requerimientos no funcionales - Elaboración propia.*

N°	Descripción
1	Diseño responsivo e intuitivo
2	Responder con un tiempo aceptable a las peticiones del usuario

3.6.1.2. Roles del sistema.

Tabla 4. Roles del sistema web – Elaboración propia.

N°	Rol	Frecuencia que usará la aplicación	Objetivo
1	Administrativos (Oficina de Gerencia y Tesorería)	Poca frecuencia	Visualización de reportes. Gestionar el sistema
2	DIGETI (Operador, Atención al cliente)	Uso frecuente	Brindar acceso Mantener actualizada la información

3.6.1.3. Historias de usuario.

Una vez realizado los requerimientos funcionales se realizó la elaboración de las historias de usuario. Se tiene niveles de prioridad y complejidad para la descripción de las historias de usuario, las cuales se especifica en cada historia de usuario requerida.

Tabla 5. Nivel de prioridad – *Elaboración propia.*

Valor	Prioridad
1	Baja
2	Media
3	Alta
4	Muy Alta

Tabla 5. Nivel de Complejidad – *Elaboración propia.*

Valor	Complejidad
1	Baja
2	Media
3	Alta
4	Muy Alta

Para el entendimiento de los requerimientos, estas son plasmadas en historias de usuario; técnica utilizada en la metodología XP. A continuación, se describen. Ver todas las historias en Anexo 1.

Tabla 6. Historia de Usuario N° 1: Gestionar personas – *Elaboración propia.*

Historia de usuario		
Requerimiento: RF02	ID: USHIS001	Usuario: DIGETI (Operador, Atención al cliente)
Nombre de historia: Gestionar personas		
Objetivo: Tener información del personal		
Prioridad: Alta	Complejidad: Media	
Descripción:		
Como DIGETI (Operador, Atención al cliente) deseo poder registrar y editar los datos de una persona, así mismo poder eliminar un registro siempre y cuando esta persona no esté vinculada con ninguna asignación; con el propósito de tener un registro de los datos y la cantidad de personas para poder realizar la entrega de línea y/o equipo según corresponda.		
Criterios de aceptación:		
1. Verificar que no haya duplicidad de una misma persona mediante el número de Documento de Identidad.		
2. Una persona vinculada a cualquier actividad del sistema no puede ser eliminada.		

Tabla 7. *Historia de Usuario N° 2: Gestionar roles y permisos – Elaboración propia.*

Historia de usuario	
Requerimiento: RF01	ID: USHIS002 Usuario: DIGETI (Operador, Atención al cliente)
Nombre de historia: Gestionar roles y permisos	
Objetivo: Controlar el acceso de los usuarios al sistema.	
Prioridad: Media	Complejidad: Alta
Descripción:	
Como DIGETI (Operador, Atención al cliente) deseo poder asignar rol y permisos para el acceso al sistema web; con el propósito de puedan tener acceso a los datos designados según el rol.	
Criterios de aceptación:	
1. Dar acceso al sistema a un usuario determinado asignado un rol.	
2. Permitir el ingreso al sistema según el rol brindado.	

Tabla 8. *Historia de Usuario N° 3: Asociar un plan a una línea – Elaboración propia.*

Historia de usuario	
Requerimiento: RF03	ID: USHIS003 Usuario: DIGETI (Operador, Atención al cliente)
Nombre de historia: Asociar un plan a una línea	
Objetivo: Asignar a una determinada equipo o persona.	
Prioridad: Alta	Complejidad: Alta
Descripción:	
Como DIGETI (Operador, Atención al cliente) deseo asignar un plan a una línea (número de celular), y un plan pueda pertenecer a un operador con la finalidad de asociar a un equipo o realizar la entrega en caso de que solo se solicite una línea.	
Criterios de aceptación:	
1. Un plan debe pertenecer a un operador.	
2. Una línea debe pertenecer a un plan y a un simcard.	
3. NO puede eliminarse un registro si se tiene al menos una asignación de equipo.	

Tabla 9. *Historia de Usuario N° 4: Gestionar equipos – Elaboración propia.*

Historia de usuario	
Requerimiento: RF04 ID: USHIS004 Usuario: DIGETI (Operador, Atención al cliente)	
Nombre de historia: Gestionar equipos	
Objetivo: Asignar a una línea o persona.	
Prioridad: Alta	Complejidad: Alta
Descripción:	
Como DIGETI (Operador, Atención al cliente) deseo realizar el registro de un equipo con sus respectivos detalles y accesorios con la finalidad de asociar con una línea o realizar la entrega en caso de que solo se solicite un equipo y tener un control.	
Criterios de aceptación:	
<ol style="list-style-type: none">1. Un equipo debe pertenecer a un modelo.2. Un modelo debe estar asociado con una marca.3. Un equipo puede tener más de un accesorio.4. Un modelo debe estar sujeto a políticas en caso de pérdida, robo o cualquier daño ocasionado sobre un equipo en una determinada campaña.5. NO puede eliminarse un registro si se tiene al menos una asignación.	

Tabla 10. *Historia de Usuario N° 5: Gestionar personas – Elaboración propia.*

Historia de usuario	
Requerimiento: RF05 ID: USHIS005 Usuario: DIGETI (Operador, Atención al cliente)	
Nombre de historia: Asignación de línea y/o equipo a un cargo	
Objetivo: Realizar la entregada	
Prioridad: Alta	Complejidad: Muy Alta
Descripción:	

Como DIGETI (Operador, Atención al cliente) deseo realizar una asignación de una línea y/o equipo a un cargo; con el propósito de realizar la entrega de estos según lo requiera.

Criterios de aceptación:

1. Permitir realizar la búsqueda mediante el número de celular al seleccionar una línea.
 2. Permitir realizar la búsqueda con IMEI, modelo y marca al seleccionar un equipo.
 3. Permitir realizar la búsqueda de un cargo mediante el nombre o nombre de área.
 4. Se puede hacer la asignación de solo equipo, línea o ambas.
 5. En un registro solo se puede EDITAR solo si esta no ha sido entregada.
 6. Una línea, un equipo y un cargo solo se pueden pertenecer a una asignación activa.
 7. Una vez seleccionado el equipo pueda mostrar la información de los accesorios
 8. NO puede eliminarse un registro si se tiene al menos una asignación.
-

Tabla 11. *Historia de Usuario N° 6: Asignación de línea y/o equipo–Elaboración propia.*

Historia de usuario

Requerimiento: RF06 **ID:** USHIS006 **Usuario:** DIGETI (Operador, Atención al cliente)

Nombre de historia: Entrega de línea y/ o equipos

Objetivo: Brindarle el equipo o línea asignada según el cargo a la persona

Prioridad: Muy Alta

Complejidad: Muy Alta

Descripción:

Como DIGETI (Operador, Atención al cliente) deseo entregar una línea y/o equipo a una persona según el cargo designado y tener el control de las entregas y la disponibilidad de estas.

Criterios de aceptación:

1. Al seleccionar la persona realizar la búsqueda mediante el Número de Documento de Identidad.
 2. Una vez seleccionado la persona pueda permitirme editar los datos de la persona sin tener que salir de la pantalla.
 3. En la lista de registro se puede realizar un filtro mediante área, equipo asignado (Modelo), Entregado, sin entregar.
 4. Generar un Formato de entrega de línea y/o equipo, según el formato utilizado.
 5. Imprimir el documento de la entrega de línea y/o equipo
 6. NO puede eliminarse un registro si está asignado a una persona.
 7. Permitir exportar en Excel el registro.
-

Tabla 12. *Historia de Usuario N° 7: Gestionar las devoluciones – Elaboración propia.*

Historia de usuario

Requerimiento: RF07 **ID:** USHIS007 **Usuario:** DIGETI (Operador, Atención al cliente)

Nombre de historia: Gestionar devoluciones

Objetivo: Tener una lista actualizada para saber la cantidad de línea y/o equipos disponibles.

Prioridad: Alta

Complejidad: Alta

Descripción:

Como DIGETI (Operador, Atención al cliente) deseo registrar la devolución de línea Y/o equipo entregado, según el motivo y el estado del equipo; con la finalidad de tener el control y la disponibilidad.

Criterios de aceptación:

1. Una devolución debe tener un motivo.
 2. Una devolución debe tener al menos una condición de equipo.
 3. Una devolución se realiza a partir de una entrega de línea y/o equipo.
 4. En caso de un equipo, verificar (Poder editar) los accesorios con los cuales se está realizando la devolución.
 5. Generar un formato de devolución.
 6. Imprimir el documento de devolución.
 7. NO puede eliminarse un registro.
-

Tabla 13. *Historia de Usuario N° 8: Gestionar pérdidas y robos - Elaboración propia.*

Historia de usuario

Requerimiento: RF08 **ID:** USHIS008 **Usuario:** DIGETI (Operador, Atención al cliente)

Nombre de historia: Gestionar pérdidas y robos

Objetivo: Tener un registro de equipos perdidos o robados para poder tener una cantidad actualizada de equipos disponibles.

Prioridad: Alta

Complejidad: Alta

Descripción:

Como DIGETI (Operador, Atención al cliente) deseo registrar la pérdida o robo según sea reportado la línea y/o equipo entregado, y poder realizar la gestión de esta para poder realizar la reposición según lo establecido en la política establecida en cada campaña.

Criterios de aceptación:

1. Una pérdida o robo se registra a partir de una entrega de línea y/o equipo.
 2. Al realizar un registro pueda realizarse la búsqueda mediante el número de documento de identidad de la persona entregada el equipo.
 3. El sistema pueda calcular los montos a pagar de cada usuario según la política.
-

Tabla 14. *Historia de Usuario N° 9: Mostrar estadísticas generales - Elaboración propia.*

Historia de usuario		
Requerimiento: RF09	ID: USHIS009	Usuario: DIGETI (Operador, Atención al cliente), Administrativos (Oficina de Gerencia y Tesorería)
Nombre de historia: Mostrar estadísticas generales		
Objetivo: Mostrar estadísticas generales		
Prioridad: Alta	Complejidad: Media	
Descripción:		
Como DIGETI (Operador, Atención al cliente) y Administrativos (Oficina de Gerencia y Tesorería) deseo ver la estadística general de los recursos del sistema y que esta a su vez pueda mostrarme un detalle.		
Criterios de aceptación:		
<ol style="list-style-type: none">1. Mostrar la cantidad total de líneas, perdida/robo, solo chip (líneas), líneas libres.2. Cantidad de usuarios por mes3. Cantidad de líneas por área4. Cantidad de equipos por modelo.5. Al realizar un clic sobre los datos pueda mostrarme un detalle del registro.		

Tabla 15. *Historia de Usuario N° 10: Gestionar venta de equipos – Elaboración propia.*

Historia de usuario		
Requerimiento: RF11	ID: USHIS010	Usuario: DIGETI (Operador, Atención al cliente)
Nombre de historia: Gestionar venta de equipos.		
Objetivo: Realizar la venta de un equipo		
Prioridad: Baja	Complejidad: Media	
Descripción:		

Como DIGETI (Operador, Atención al cliente) y Administrativos (Oficina de Gerencia y Tesorería) deseo configurar la venta de equipos por modelo.

Criterios de aceptación:

1. Asignar un precio por modelo.
 2. Generar un formato de venta.
 3. Imprimir el documento.
-

Tabla 16. *Historia de Usuario N° 11: Visualización de reportes – Elaboración propia.*

Historia de usuario		
Requerimiento: RF10	ID: USHIS011	Usuario: DIGETI (Operador, Atención al cliente), Administrativos (Oficina de Gerencia y Tesorería)
Nombre de historia: Visualización de reportes		
Objetivo: Verificar los datos y tomar decisiones.		
Prioridad: Baja	Complejidad: Alta	
Descripción:		
Como DIGETI (Operador, Atención al cliente) y Administrativos (Oficina de Gerencia y Tesorería) deseo contar con reportes mensuales de las líneas y/o equipos asignados con detalle, las líneas y/o equipos devueltos con detalle (fecha), la cantidad de líneas o equipos reportados por pérdida o robo (según las políticas de uso) (Resumen por fecha), la cantidad de equipos por cada modelo.		
Criterios de aceptación:		
1. Permitir descargar en formato de Excel y PDF, además poder visualizarse en línea		

3.6.1.4. Planificación de iteraciones.

A continuación, se detalla la planificación de las iteraciones realizadas, en la tabla 17 se detalla cada iteración.

Tabla 17. *Planificación de iteraciones – Elaboración propia.*

Iteración	Descripción	Elementos liberados	Fecha de Inicio	Fecha de Finalización
Iteración 1: Fase de planificación	Levantamiento de información. Realizar el análisis del proyecto.	Documentos que maneja la empresa. Documento de análisis.	14/06/2020	04/07/2020
Iteración 2: Fase de Diseño	Análisis y diseño del sistema web.	Diseño prototipo. Diseño del modelo de base de datos. Diseño de arquitectura del sistema web.	12/07/2020	01/08/2020
Iteración 3,4,5,6,7: Fase de Desarrollo Back-Front	Construcción de APIs. Desarrollo de interfaz de usuario.	Código del Backend y Frontend.	09/08/2020	05/09/2020
Iteración 6: Fase de Transición	Corrección de bugs. Pruebas de Aceptación. Documentación.	Sistema web desplegado. Documentación completa.	06/12/2020	27/02/2021

Cada una de las iteraciones definidas tiene una serie de tareas, a continuación, se detalla las tareas a realizar en cada una de las iteraciones.

Iteración 1: Fase de Planificación.

Tabla 18. *Tareas a desarrollar en la Iteración 1 – Elaboración propia.*

Tarea	Duración prevista	Inicio	Fin	Duración Real
Investigación previa y levantamiento de información.	18 horas	14/06/2020	19/06/2020	18 horas
Análisis de requerimientos funcionales y no funcionales.	8 horas	21/06/2020	25/06/2020	8 horas
Identificación de roles del sistema.	1 hora	26/06/2020	26/06/2020	1 hora
Realizar las historias de usuario.	10 horas	26/06/2020	01/07/2020	12 horas

Planificación de Iteraciones.	4 horas	01/07/2020	04/07/2020	4 horas
Documentación previa	4 horas	01/07/2020	04/07/2020	4 horas
Total	45 horas	14/06/2020	04/07/2020	47 horas

Como se aprecia en la tabla 19, en la iteración 1 la planificación ha sido mínimamente pesimista, provocando una divergencia de 2 horas en la duración total y la esperada, debido a que en la realización de las historias de usuario se tuvieron que realizar correcciones, reduciendo el número historias de usuario planteados inicialmente.

Iteración 2: Fase de Diseño.

Tabla 19. *Tareas a desarrollar en la Iteración 2 – Elaboración propia.*

Tarea	Duración prevista	Inicio	Fin	Duración Real
Diseñar el prototipo del sistema web.	18 horas	12/07/2020	17/07/2020	24 horas
Diseñar el modelo de la base de datos.	10 horas	19/07/2020	23/07/2020	12 horas
Diseño de la arquitectura del sistema web.	12 horas	24/07/2020	30/07/2020	12 horas
Documentación previa	4 horas	30/07/2020	01/08/2020	4 horas
Total	44 horas	12/07/2020	01/08/2020	52 horas

Como podemos ver la tabla 20, en la iteración 3 se ha producido una desviación entre la duración planificada y la real, debido a que en la realización del diseño del prototipo se tiene una divergencia de 6 horas, debido a que en un primer diseño se detectaron fallas según el proceso propuesto para la gestión de celulares, por ello también se tiene una divergencia de 2 horas para tomar las correcciones del modelado del diseño de la base datos.

Iteración 3,4,5,6,7: Fase de Desarrollo Back-Front.

Tabla 20. *Tareas a desarrollar en la Iteración 3,4,5,6,7- Elaboración propia.*

Iteración	Tarea	Duración prevista	Inicio	Fin	Duración Real
Iteración 3	Gestionar roles y permisos para dar acceso al usuario al sistema.	16 horas	09/08/2020	16/08/2020	16 horas
	Gestionar personas.	26 horas	17/08/2020	29/08/2020	26 horas
Iteración 4	Gestionar líneas.	21 horas	06/09/2020	19/09/2020	26 horas
	Gestionar equipos	21 horas	20/09/2020	03/10/2020	24 horas
Iteración 5	Asignación de líneas	22 horas	11/10/2020	21/10/2020	22 horas
	Entrega de líneas y/o equipos	33 horas	22/10/2020	31/10/2020	35 horas
	Gestionar devoluciones	14 horas	01/11/2020	14/11/2020	14 horas
Iteración 6	Gestionar equipos perdidos y robados	22 horas	22/11/2020	02/12/2020	22 horas
	Mostrar estadísticas generales sobre actividades y/o recursos del sistema.	20 horas	03/12/2020	12/12/2020	16 horas
	Visualización de reportes.	14 horas	20/12/2020	26/01/2021	14 horas
Iteración 7	Gestionar la venta de equipos	22 horas	27/12/2020	09/01/2021	

3.6.2. Diseño

3.6.2.1. Diseño del prototipo.

A continuación, se presentan los prototipos en las pantallas principales del sistema.

Login, en la figura 10 se muestra el sistema de acceso o login, el cual se encarga de la autenticación del usuario, comprobando que el usuario y contraseña sean correctos, y establece un entorno inicial para el usuario.

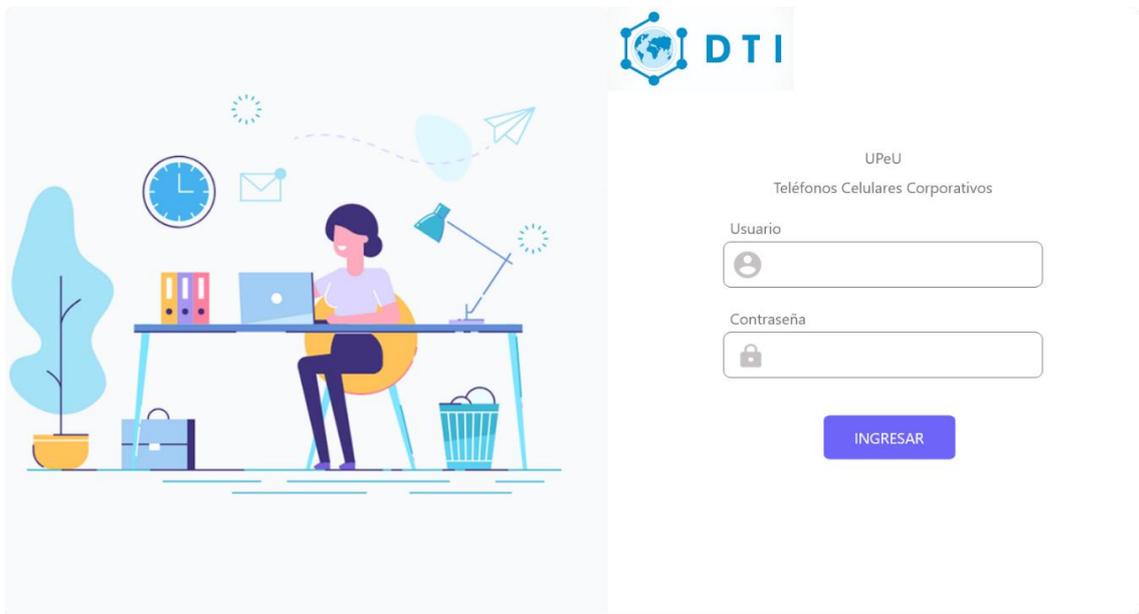


Figura 10. Login del sistema web - Elaboración propia.

Pantalla principal, en la figura 11 muestra la pantalla principal del sistema, donde se puede apreciar el menú lateral, el acceso a las pantallas principales y un resumen de estadísticas generales.

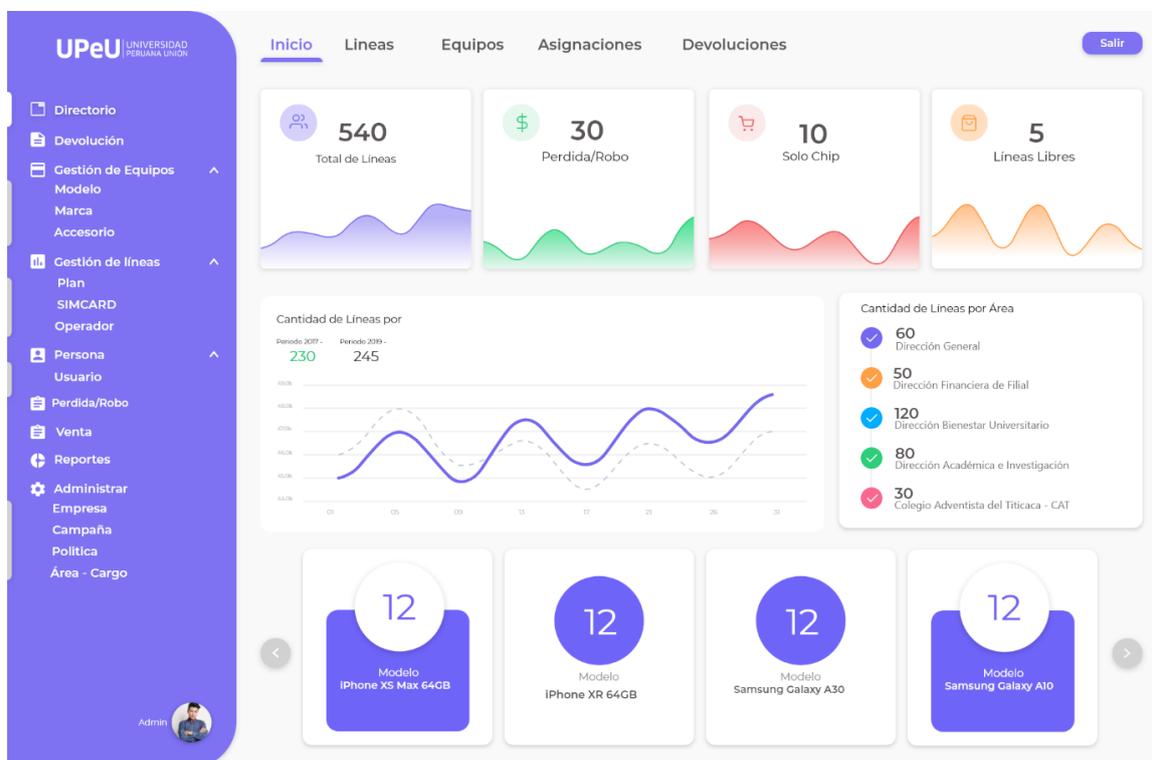


Figura 11. Pantalla principal del sistema - Elaboración propia.

Gestión de Equipos, en la figura 12 se muestra el prototipo de la gestión de equipos, en cual se realiza la gestión de equipos celulares y los accesorios.

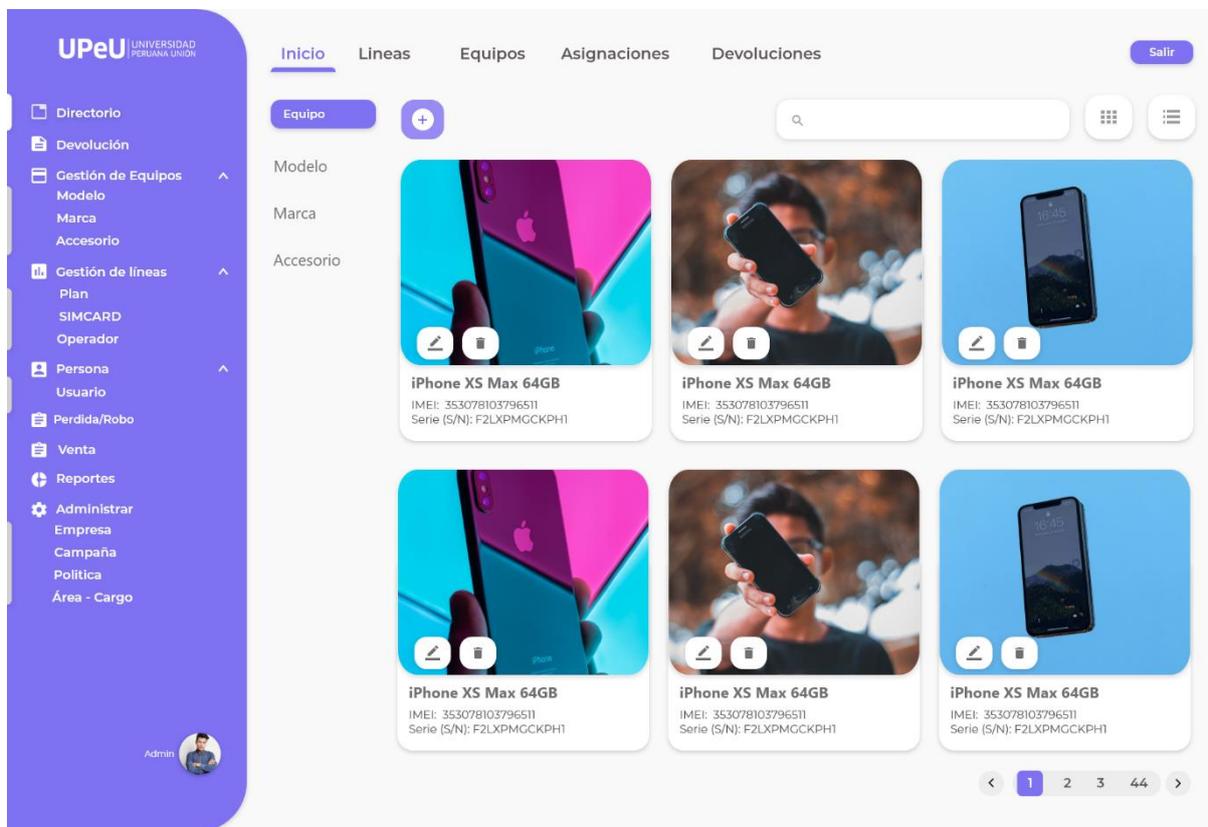


Figura 12. Gestión de equipos - Elaboración propia.

Entrega de línea y/o equipos, en la 13,14 y 15 se muestra el prototipo de asignación de línea y/o equipo, en cual se gestiona la entrega y devoluciones de la línea y/o equipo según el cargo asignado, finalizando con el formato de entrega o devolución.

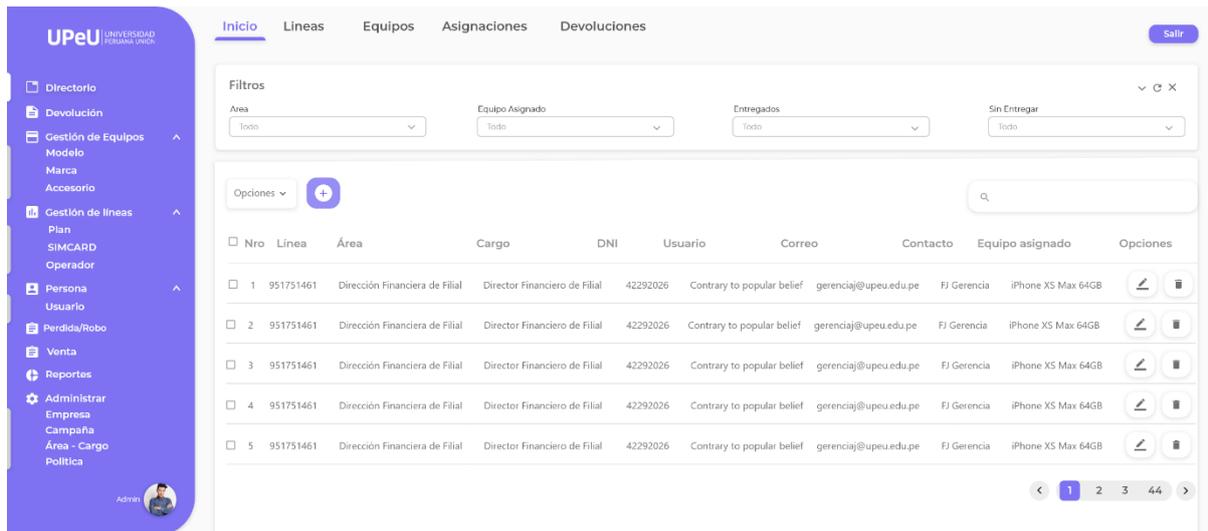


Figura 13. Vista de entrega de línea y/o equipo - Elaboración propia.

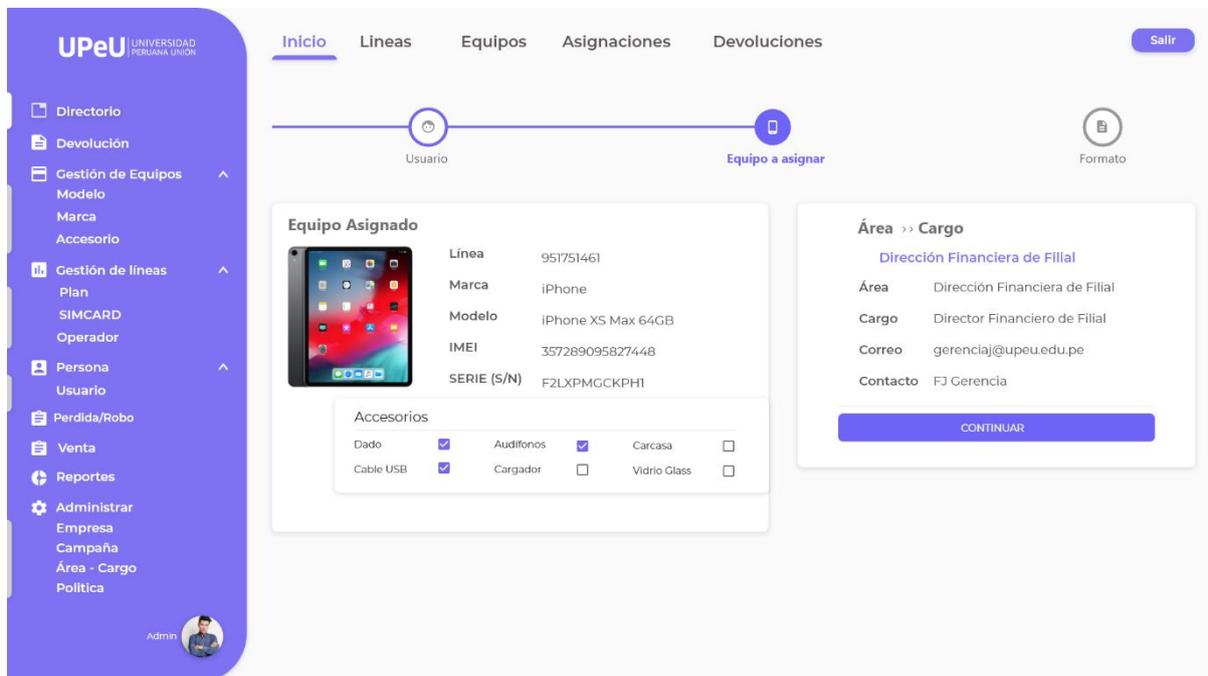


Figura 14. Asignación de línea y/o equipo - Elaboración propia.

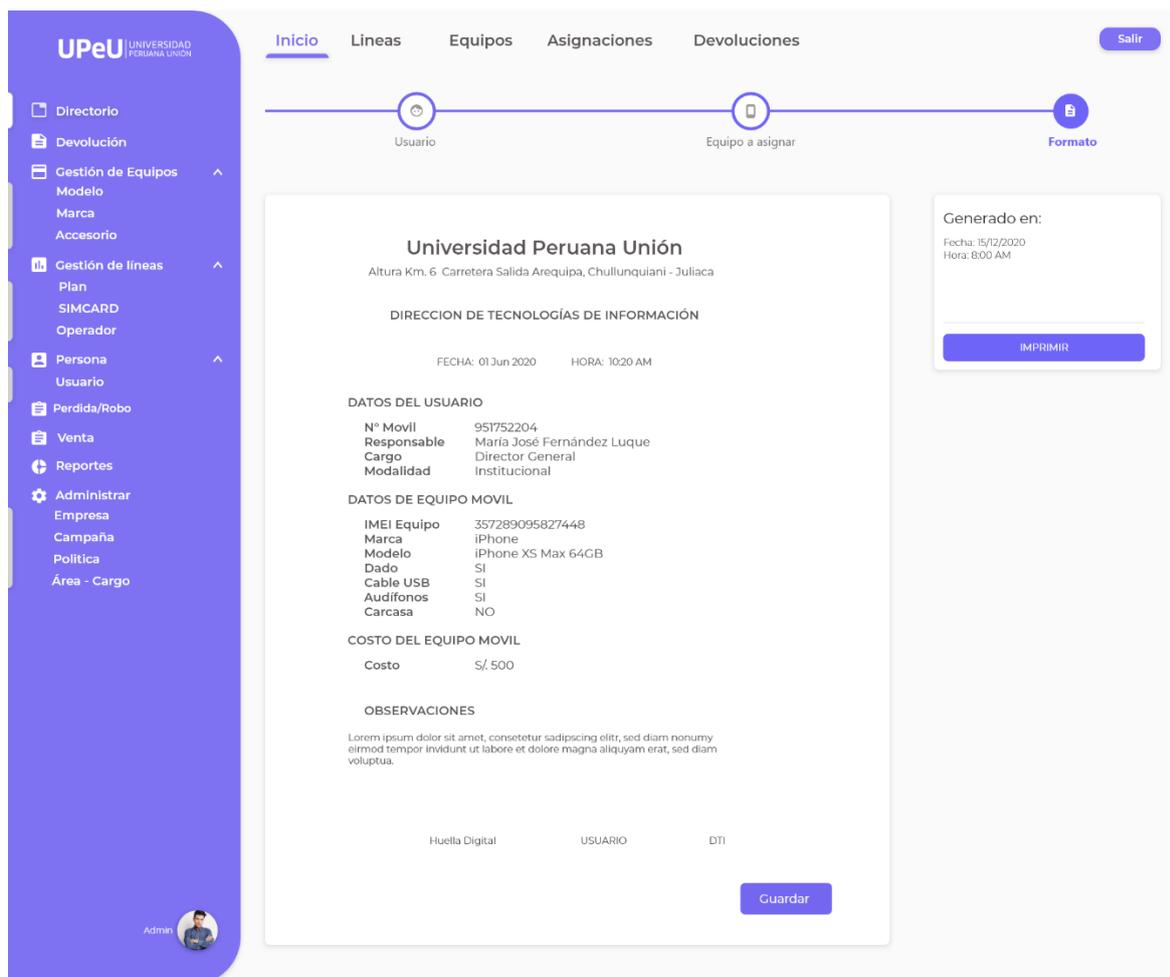


Figura 15. Formato generado de entrega de una línea y/o equipo - Elaboración propia.

3.6.2.2. Modelo de base de datos.

El análisis de requerimientos para el diseño del modelado de la base de datos incorpora las mismas que los requerimientos funcionales que se establecieron juntamente con el cliente. Se analizaron los requerimientos, para definir qué datos serán almacenados, además se definen los atributos de cada uno. El diseño de la base de datos nos permite abstraer los objetos que están involucrados y así poder crear nuestros modelos con LoopBack; un modelo describe objetos del dominio empresarial y define una lista de propiedades como nombre, tipo y otras restricciones. Como resultado de este análisis se obtuvo el modelo de la base de datos según se muestra en la figura 16.

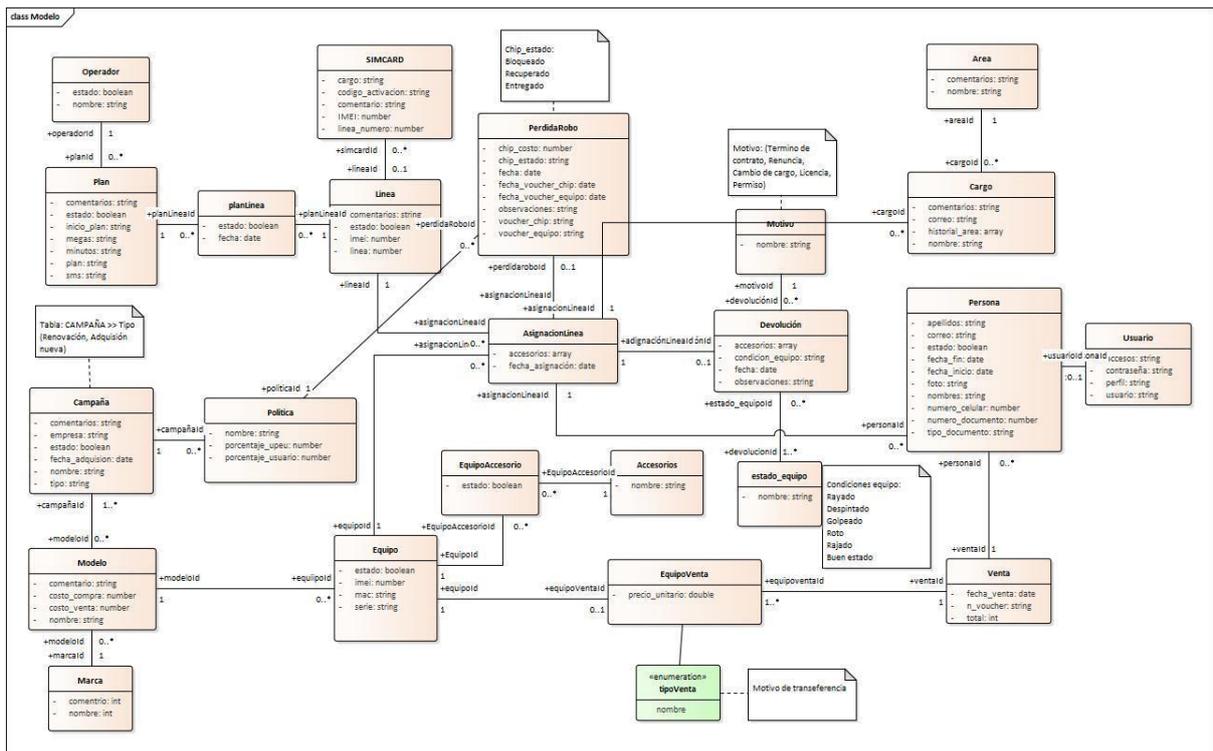


Figura 16. Diseño de modelo de la base de datos - Elaboración propia.

3.6.2.3. Diseño de la arquitectura.

En la figura 17. se muestra la arquitectura tecnología que se utilizó para el desarrollo del sistema web, es una arquitectura cliente servidor. Para la construcción del Backend se utilizó el Framework LoopBack 4, escrito sobre Node.js y apoyado sobre Express.js para la construcción de API REST. La característica de LoopBack es la creación de modelos, controladores, y todo lo que conlleva a la arquitectura mediante el uso de la terminal, o su herramienta de cliente, y eso facilita y agiliza en gran manera el desarrollo.

LoopBack utiliza el enrutamiento Express y middleware como conexión a una canalización de solicitud o respuesta para casos de API, más allá del procesamiento HTTP entrante. Otra de las características del Framework es su capacidad de conectarse a múltiples recursos de base de datos, y en sus dos tipos, NoSQL y SQL. También tiene la capacidad de conectarse a servicios externos.

El Frontend se realizó con Framework Nuxt.js, basado en Vue.js que te provee un marco de desarrollo completo, facilita la creación de aplicaciones universales (SSR) y aplicaciones de una página (SPA), es decir, que nos permite renderizar tanto en el cliente y el servidor. Con Nuxt.js podemos definir qué contenido es importante, por lo tanto, puede merecer ser

creado en el servidor o servido desde el servidor y que contenido no lo es, pudiendo ser construido en el cliente porque lo esencial se cargará al principio y lo resto se irá cargando asincrónicamente.

Nuxt.js le brinda el acceso al método fetch y asyncData dentro de sus componentes, con el cual puede obtener datos y representarlos en el lado del cliente. Nuxt.js al igual que Vue.js se basa en componentes, cada componente se divide en tres partes, en la estructura el código HTML, un CSS para dar estilos y un JavaScript para dotarlo de funcionalidad. Cada componente es reutilizable e independiente y poder utilizar en otros proyectos. Nuxt.js genera su propio enrutador basado en la estructura de la carpeta lo que hace más fácil y rápido crear uno, simplemente crea el directorio y los archivos, y Nuxt.js hace todo el trabajo. Por ejemplo, si creamos un directorio y un archivo "about / index.vue", Nuxt.js crea automáticamente la ruta "/ about" para esa página, no es necesario definir o configurar rutas en ningún otro lugar de la aplicación. Así mismo le brinda una excelente estructura del código para organizar su aplicación.

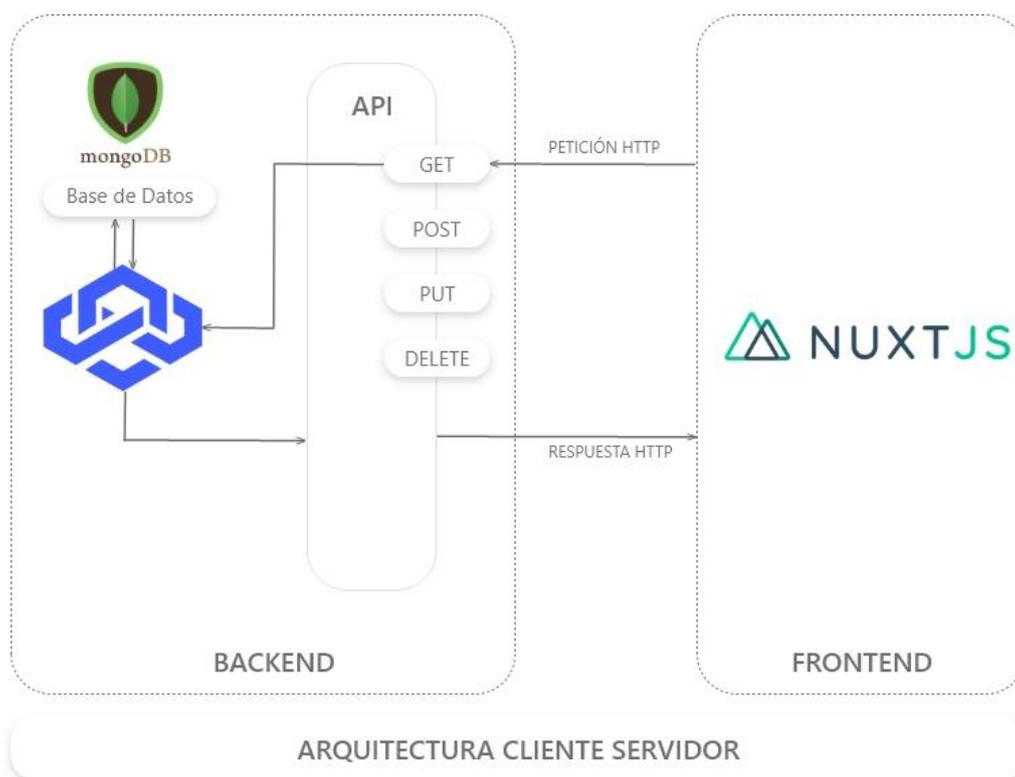


Figura 17. Diseño de la arquitectura del sistema web - Elaboración propia.

3.6.3. Desarrollo

3.6.3.1. Desarrollo de Backend (Construcción de las APIs).

Para almacenar los datos se optó por MongoDB, una base de datos NoSQL, la elección fue más por el Framework Loopback, ya que es la que mejor se adapta, además se tuvo en cuenta la facilidad de poder desplegar y administrar la base de datos desde una interfaz de usuario amigable e intuitivo en la plataforma MongoDB Atlas.

Lo primero que se hizo fue crear una cuenta en MongoDB Atlas, una vez se obtuvo la cuenta se procedió a crear la base de datos para nuestro proyecto, el proceso es seguir algunos pasos sin ninguna configuración avanzada; con un par de clicks y está listo para ser usada! pero sin antes agregar un usuario que tenga los permisos de lectura y escritura el cual nos permitirá hacer conexiones desde nuestra aplicación de backend o cualquier otra aplicación que da la posibilidad de hacer conexiones a MongoDB.

La siguiente figura 18 muestra la interfaz de MongoDB Atlas que te da el control completo de la base de datos para poder administrarlo mediante la interfaz de usuario, significa que todo el proceso de configuración lo hace de manera automática y lo deja lista para hacer la conexión. Las colecciones y documentos que aparecen en la figura son ya creadas desde Loopback, donde lo único que se tiene que hacer es colocar en un archivo JSON las credenciales o también es posible hacerlo mediante el CLI de Loopback.

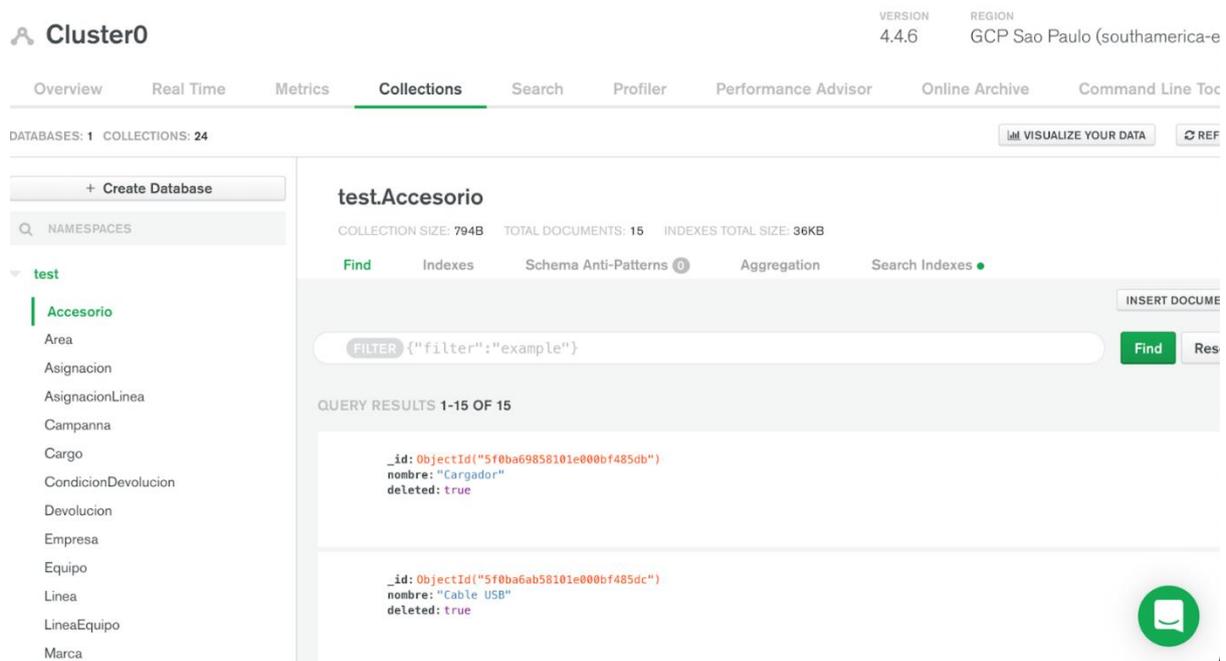


Figura 18. Interfaz de MongoDB Atlas – Elaboración propia.

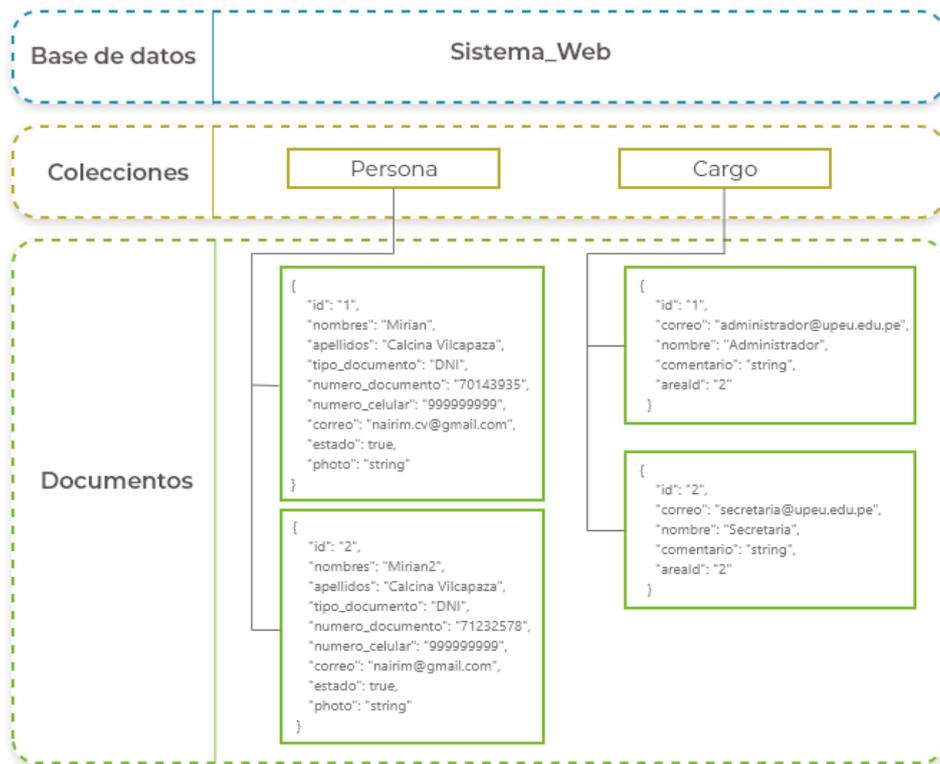


Figura 19. Estructura de MongoDB - Elaboración propia.

Para el desarrollo del Backend se utilizó LoopBack en su última versión que es la 4, esto por todas las ventajas descritas en el capítulo anterior. La herramienta principal de loopback es su CLI, que mediante comandos por consola o terminal se realizan absolutamente todas las operaciones empezando desde crear un proyecto hasta realizar todas las operaciones que se muestran en la siguiente figura 20.

```

→ ~ lb4 -help
Available commands:
lb4 app
lb4 extension
lb4 controller
lb4 datasource
lb4 import-lb3-models
lb4 model
lb4 repository
lb4 service
lb4 example
lb4 openapi
lb4 observer
lb4 interceptor
lb4 discover
lb4 relation
lb4 update
lb4 rest-crud
lb4 copyright
lb4 install-completion
lb4 uninstall-completion
→ ~

```

Figura 20. Comandos disponibles en LoopBack - Elaboración propia.

Para iniciar con un proyecto LoopBack primero se tiene que instalar la CLI, un paquete de node.js y lo podemos instalar mediante *npm* o *yarn* que son los manejadores de paquetes de node.js. Para crear un proyecto ejecutamos el comando *lb4 app* y esto generará toda la estructura del proyecto con TypeScript lista para empezar a crear las APIs.

En figura 21 vemos las partes que necesita LoopBack para lograr exponer una API, que básicamente son 4; Fuentes de datos, Modelos, Repositorios y Controladores. En la creación de una fuente de datos, se realiza la configuración de la conexión de la base de datos creada desde MongoDB Atlas. Luego podemos agregar los modelos en base al modelo de la base de datos diseñado para el proyecto. Para crear un repositorio es necesario tener ya creadas las dos cosas mencionadas anteriormente, ya que la función de un repositorio es crear una relación entre el modelo y la fuente de datos; recordar que LoopBack da la posibilidad de tener más de una fuente de datos que en este proyecto no es necesario, basta con tener una sola fuente de datos. Y por último tenemos la creación de los controladores que son creados a partir de un repositorio que te permiten realizar las operaciones CRUD sin ningún esfuerzo.

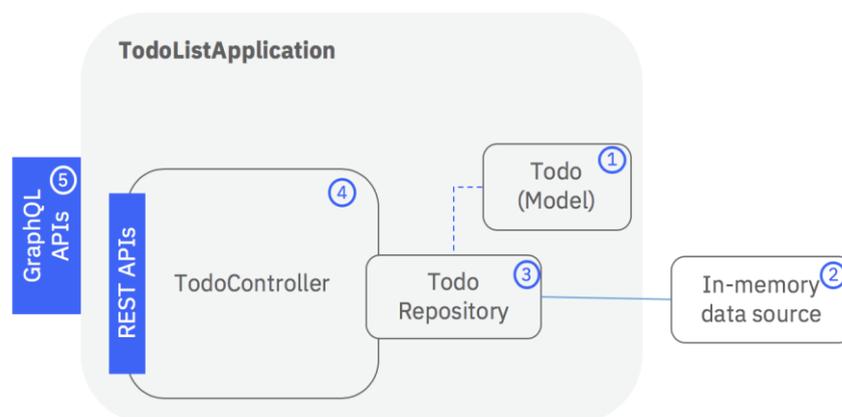


Figura 21. Creación de una API REST – (StrongLoop, 2021).

La instalación se realiza mediante el gestor de paquetes “npm”, que es de la siguiente forma:
npm i -g @loopback/cli.

Para la creación de los modelos ejecutamos el siguiente comando *lb4 model* según se muestra en la figura 22.

```
→ sistemaweb lb4 model
? Model class name: persona
? Please select the model base class Entity (A persisted model with an ID)
? Allow additional (free-form) properties? Yes
Model Persona will be created in src/models/persona.model.ts

Let's add a property to Persona
Enter an empty property name when done

? Enter the property name: nombres
? Property type: string
? Is nombres the ID property? No
? Is it required?: Yes
? Default value [leave blank for none]:

Let's add another property to Persona
Enter an empty property name when done

? Enter the property name: apellidos
? Property type: string
? Is apellidos the ID property? No
? Is it required?: Yes
? Default value [leave blank for none]:

Let's add another property to Persona
Enter an empty property name when done

? Enter the property name: tipo_documento
? Property type: string
? Is tipo_documento the ID property? No
? Is it required?: Yes
? Default value [leave blank for none]:

Let's add another property to Persona
Enter an empty property name when done

? Enter the property name: numero_documento
? Property type: string
? Is numero_documento the ID property? No
? Is it required?: Yes
? Default value [leave blank for none]:

Let's add another property to Persona
Enter an empty property name when done

? Enter the property name:
  create src/models/persona.model.ts
  update src/models/index.ts

Model Persona was created in src/models/

→ sistemaweb
```

Figura 22. Crear un modelo (lb4 model) - Elaboración propia.

Un modelo describe dominio de objetos de negocio, por ejemplo, persona, cargo, área, equipo, línea. Define una lista de propiedades con nombre, tipo y otras restricciones.

Para añadir el modelo a nuestra API y poder empezar a realizar operaciones CRUD sobre el mismo, añadimos tres componentes, estrictamente en el siguiente orden:

- ✓ **Datasource:** configuración para conectar con nuestra base datos (o cualquier otra si disponemos de sus datos de conexión), en este caso, MongoDB

- ✓ **Repositorio:** provee operaciones de acceso a datos (por ejemplo, CRUD)
- ✓ **Controlador:** implementa las operaciones definidas por la API.

Para añadir **datasource** ejecutamos *lb4 datasource* según se muestra en la figura 23 aquí se crea la fuente de datos, usando el comando de origen de datos de LoopBack podemos decirle a nuestra API donde almacenar y recuperar datos, en este caso seleccionamos MongoDB y le proporcionamos la conexión.

```
→ sistemaweb lb4 datasource
? Datasource name: mongo
? Select the connector for mongo: MongoDB (supported by StrongLoop)
? Connection String url to override other settings (eg: mongodb://username:password@hostname:port/database):
? host: localhost
? port: 27017
? user:
? password: [hidden]
? database: sistemaweb
? Feature supported by MongoDB v3.1.0 and above: Yes
  create src/datasources/mongo.datasource.config.json
  create src/datasources/mongo.datasource.ts
npm WARN sistemaweb@1.0.0 No license field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin",
+ loopback-connector-mongodb@4.2.0
added 33 packages from 21 contributors and audited 693 packages in 18.13s
found 0 vulnerabilities

  update src/datasources/index.ts
Datasource Mongo was created in src/datasources/
→ sistemaweb
```

Figura 23. Conexión a base de datos (lb4 datasource) - Elaboración propia.

Seguidamente creamos el repositorio, el cual nos proveerá operaciones CRUD, ejecutamos *lb4 repository*, es como un conector entre la fuente de datos y los modelos, uno de sus trabajos es actuar como inyector y extractor de base de datos, según se muestra en la figura 24, necesita usar *DefaultCrudRepository* clase, que vincula el modelo con la fuente de datos, esto hace posible manejar operaciones crudas básicas como crear una nueva persona. También se puede agregar métodos personalizados al repositorio para realizar más operaciones.

```
→ sistemaweb lb4 repository
? Please select the datasource MongoDatasource
? Select the model(s) you want to generate a repository Persona
? Please select the repository base class DefaultCrudRepository (Legacy juggler bridge)

create src/repositories/persona.repository.ts
update src/repositories/index.ts

Repository PersonaRepository was created in src/repositories/
→ sistemaweb
```

Figura 24. Crear un repositorio (lb4 repository) - Elaboración propia.

Por último, el controlador, ejecutamos **lb4 controller**, contiene el código para todas las funciones del proyecto, maneja la toda lógica del negocio, y actúa como un puente entre la API HTTP/REST y los modelos de dominio/base de datos, y estas son expuestas para realizar una conexión de los controladores a los modelos que están los repositorios, tiene un ciclo de vida solicitud-respuesta, ver figura 25.

```
→ sistemaweb lb4 controller
? Controller class name: persona
Controller Persona will be created in src/controllers/persona.controller.ts

? What kind of controller would you like to generate? REST Controller with CRUD functions
? What is the name of the model to use with this CRUD repository? Persona
? What is the name of your CRUD repository? PersonaRepository
? What is the name of ID property? id
? What is the type of your ID? string
? Is the id omitted when creating a new instance? Yes
? What is the base HTTP path name of the CRUD operations? /personas
create src/controllers/persona.controller.ts
update src/controllers/index.ts

Controller Persona was created in src/controllers/
→ sistemaweb
```

Figura 25. Crear un controlador (lb4 controller) - Elaboración propia.

En la siguiente figura 26 se tiene el controlador de **persona**, Este controlador necesita de un modelo, el cual se conectan mediante el repositorio: estas son definidas en el constructor. También se puede observar el método en el cual contiene el endpoint API REST (personas) en donde se crea una persona.

```
control-celulares > src > controllers > TS persona.controller.ts > PersonaController > constructor
19 import {Persona} from '../models';
20 import {PersonaRepository} from '../repositories';
21
22 export class PersonaController {
23   constructor(
24     @repository(PersonaRepository)
25     public personaRepository : PersonaRepository,
26   ) {}
27
28   @post('/personas', {
29     responses: {
30       '200': {
31         description: 'Persona model instance',
32         content: {'application/json': {schema: getModelSchemaRef(Persona)}}},
33     },
34   },
35   )
36   async create(
37     @requestBody({
38       content: {
39         'application/json': {
40           schema: getModelSchemaRef(Persona, {
41             title: 'NewPersona',
42             exclude: ['id'],
43           }),
44         },
45       },
46     })
47     persona: Omit<Persona, 'id'>,
48   ): Promise<Persona> {
49     return this.personaRepository.create(persona);
50   }
51
52   @get('/personas/count', {
53     responses: {
54       '200': {
55         description: 'Persona model count',
56         content: {'application/json': {schema: CountSchema}},
57     },
58   },
59   )
}
```

Figura 26. Controlador de persona - Elaboración propia.

Para probar nuestras APIs utilizamos el explorador de API de LoopBack, que brinda la capacidad de ver y probar, también se utilizó la herramienta de Postman.

<http://127.0.0.1:3000/explorer> / <https://tesisappcontrol.mybluemix.net/>. Ver figura 27.

PersonaAsignacionController		^
POST	/personas/{id}/asignacions	∨
PATCH	/personas/{id}/asignacions	∨
GET	/personas/{id}/asignacions	∨
DELETE	/personas/{id}/asignacions	∨
PersonaUserController		^
POST	/personas/{id}/user	∨
DELETE	/personas/{id}/user	∨
PlanController		^
GET	/planes/count	∨
PUT	/planes/{id}	∨
PATCH	/planes/{id}	∨
GET	/planes/{id}	∨
DELETE	/planes/{id}	∨

Figura 27. Explorador de APIs de Loopback - Elaboración propia.

3.6.3.2. Desarrollo de Frontend.

Para el desarrollo de la aplicación web se utilizó el framework NuxtJs, un framework muy potente para agilizar el desarrollo, para agilizar aún más se decidió usar Vuesax en su versión 4 y Tailwind css para las interfaces. A pesar de ser una de las ventajas de NuxtJs es el SSR, en este proyecto optamos por el modo renderizado de Single Page App (SPA), teniendo en cuenta que el SEO no es relevante en la aplicación, además no se necesita de un servidor de nodejs para ponerlo en funcionamiento, si no solo generarlo de manera estática y desplegarlo en cualquiera de las plataformas que te permiten desplegar proyectos de archivos estáticos.

Recordemos que Nuxt.js está sobre Vuejs, lo que significa que aprovechamos al 100% todas sus funcionalidades. En la figura 28 se muestra la estructura del proyecto, que mediante un comando del manejador de paquetes de Nodejs, ya sea *yarn* o *npm* siguiendo una serie de preguntas y eligiendo opciones al final tenemos un proyecto Nuxt.js lista para escribir código de la aplicación más no dedicarle tiempo a la configuración del entorno. A continuación, describimos el contenido de las carpetas más específicas según el proyecto realizado.

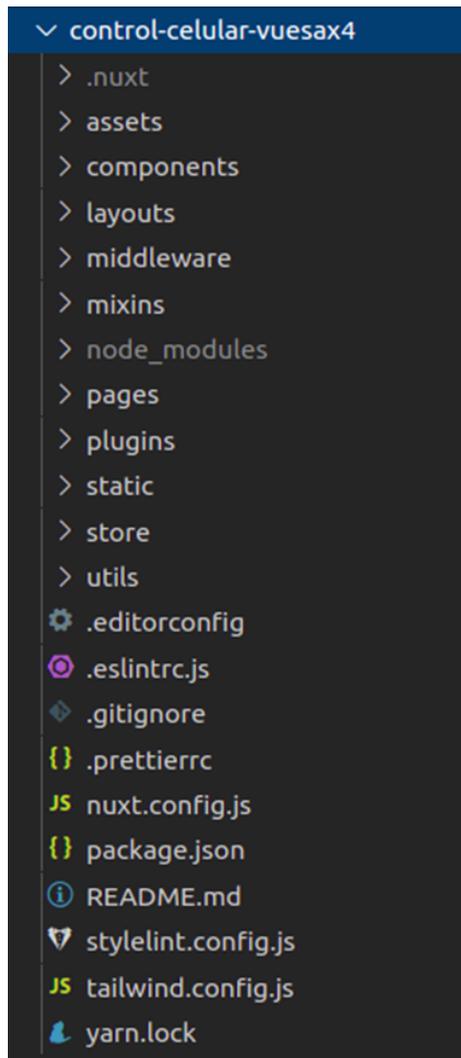


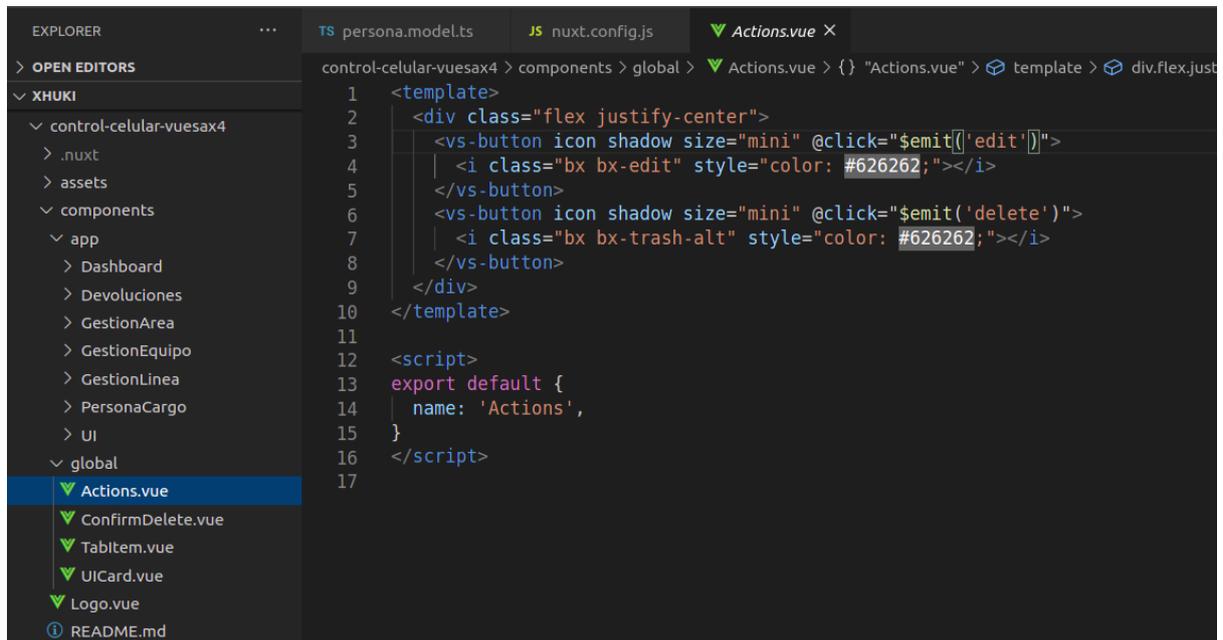
Figura 28. Estructura de carpetas Nuxt.js - Elaboración propia.

Assets: Contiene los recursos sin compilar de la app, como estilos base y configuración de Tailwind Css, también incluimos algunos logos para la app.

Components: Los “components”, son piezas de nuestra aplicación que pueden ser generalizadas y reutilizadas en múltiples partes del sistema. Cada componente es fácilmente accesible desde una página por medio de un *import* o exponerlo de manera global, así el componente está disponible en cualquier archivo *.vue* del proyecto sin necesidad de importarlo.

Se creó componentes locales y globales, la idea fundamental consiste en crear estructuras de código, cuyo principal objetivo es la posibilidad de reutilizarlo, tantas veces como sea posible. Se creó varios componentes según la necesidad del sistema web, por ejemplo, se creó el componente de las opciones de editar y eliminar *Actions.vue*, ver figura 28, este componente es utilizado en diferentes vistas del proyecto, es un componente global por lo que

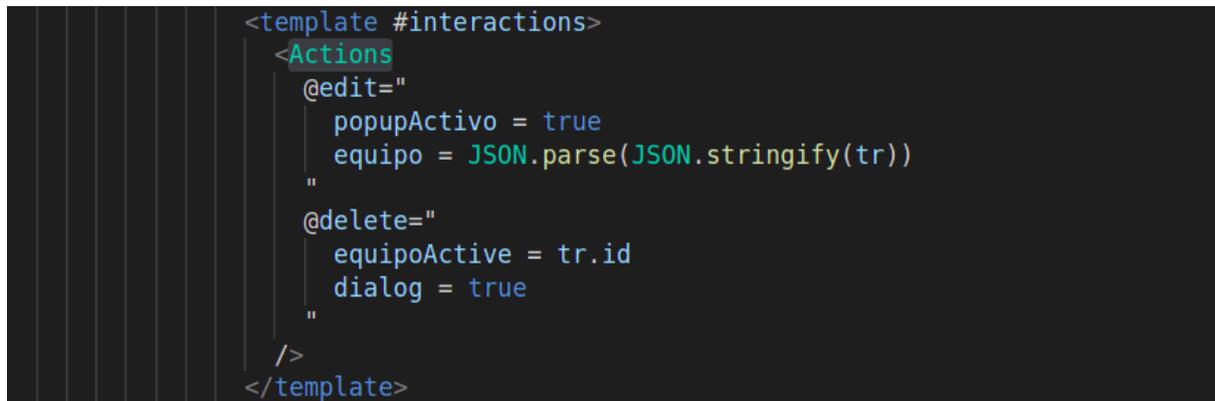
solo necesitamos añadir la línea de código correspondiente `<Actions/>`, como se ve en la figura 29.



The screenshot shows a code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like 'components', 'app', and 'global'. The 'global' folder is expanded, showing 'Actions.vue' selected. The code editor shows the following code:

```
1 <template>
2 <div class="flex justify-center">
3   <vs-button icon shadow size="mini" @click="$emit('edit')">
4     <i class="bx bx-edit" style="color: #626262;"></i>
5   </vs-button>
6   <vs-button icon shadow size="mini" @click="$emit('delete')">
7     <i class="bx bx-trash-alt" style="color: #626262;"></i>
8   </vs-button>
9 </div>
10 </template>
11
12 <script>
13 export default {
14   name: 'Actions',
15 }
16 </script>
17
```

Figura 29. Componente Actions - Elaboración propia.



The screenshot shows a code editor with the following code:

```
<template #interactions>
  <Actions
    @edit="
      popupActivo = true
      equipo = JSON.parse(JSON.stringify(tr))
    "
    @delete="
      equipoActive = tr.id
      dialog = true
    "
  />
</template>
```

Figura 30. Utilización de componente Actions - Elaboración propia.

Layouts: Contiene el andamio de la app, donde podemos incluir contenido que se repite en todas las páginas, como pueden ser los menús en el encabezado o el menú lateral. Para el proyecto solo tenemos dos, una para la página del login y el otro para el contenido de toda la app.

Middleware: Son funciones personalizadas que se ejecutan antes que se muestre alguna página o grupos de páginas (layout). Para este proyecto solo tenemos un middleware que se encarga

de verificar si el usuario está autenticado, esta es aplicada a la página de login, si el usuario está autenticado, está simplemente redirecciona al dashboard.

Los middlewares son comúnmente usados para verificar si un usuario está autenticado, o también para verificar si el usuario tiene acceso a ciertos módulos de la aplicación. Para controlar la autenticación de la app se utilizó el módulo *nuxt/auth*, uno de los más de 150 módulos disponibles que Nuxtjs posee. *Nuxt auth* tiene por efecto un middleware llamado *auth* y es la que aplicamos al layout de la app para restringir el acceso a usuarios no autenticados.

Mixins: es una forma flexible de distribuir funcionalidades reutilizables para los componentes, además es una de las características más avanzadas y potentes que tiene Vuejs. Un mixin puede contener cualquier opción de componente. Cuando un componente usa un mixin, todas las opciones en el mixins se “mezclan” en las propias opciones del componente. En el proyecto se agregó solo un mixin que básicamente es todo de la app, conexiones de api, CRUD completo escrito una sola vez para todas páginas, paginación y búsqueda incluida.

En la siguiente figura 31 se ve las partes del mixin, y verán nombre de las funciones y propiedades de manera global, por ejemplo, el objeto *item* es definido para capturar datos de los formularios, y la función *deleteItem* es para eliminar el elemento en la API provisionado desde el componente o página donde se hace el uso del mixin.

```
JS login.js • JS basic.js X
mixins > JS basic.js > [default] > data > filter
1 import { totalPages, totalSkip } from '../utils/paginate'
2
3 export default {
4 >   async fetch() {--
29   },
30   fetchOnServer: false,
31   data: () => ({
32     search: '',
33     items: [],
34     item: {},
35     loading: false,
36     dialog: false,
37     popup: false,
38     page: 1,
39 >   filter: { You, 2 months ago + update packages ...
43     },
44   }),
45
46   watch: {
47     '$route.query': '$fetch',
48
49 >   search(v) {--
62   },
63 >   size(v) {--
71   },
72 >   page(v) {--
79   },
80   },
81
82   methods: {
83 >   async getItems() {--
93   },
94 >   async saveItem() {--
113   },
114
115 >   async deleteItem() {--
124   },
125
126 >   clear() {--
132   },
133   },
134 }
---
```

Figura 31. Mixin - Elaboración propia.

En la figura 31. Utilizamos *fetch* para hacer llamadas a la API, agregamos métodos en cual se tiene cuatro funciones (getItems(), saveItem(), deleteItem() clear()) que son reutilizables en todos los componentes.

```

methods: {
  async getItems() {
    const { size, skip } = this.filter
    return await this.$axios
      .get(
        `${this.path}?filter[limit]=${size}&filter[skip]=${skip}&search=${this.search}`
      )
      .then((x) => ({
        ...x.data,
      }))
      .catch(() => ({}))
  },
  async saveItem() {
    this.loading = true
    if (this.item.id) {
      await this.$axios
        .put(`${this.path}/${this.item.id}`, this.item)
        .then(() => {
          this.clear()
        })
        .catch(() => {})
    } else {
      await this.$axios
        .post(`${this.path}`, this.item)
        .then(() => {
          this.clear()
        })
        .catch(() => {})
    }

    this.loading = false
  },
  async deleteItem() {
    this.loading = true
    await this.$axios
      .delete(`${this.path}/${this.item.id}`)
      .then(() => {
        this.clear()
      })
  }
}

```

Figura 32. Funciones (getItems(), saveItem(), deleteItem() clear() - Elaboración propia.

Podemos deshabilitar la recuperación del lado del backend si es necesario. *fetchOnServer*: false. Y de esta manera, el método de recuperación solo se llamará en el lado del cliente.

fetch no se llama en los cambios de query string por defecto. Para observar los de la query, agregamos un observador en *\$route.query* y llamar a *\$fetch*. Ver figura 32.

Para poder utilizar el *mixin*, importamos la ruta donde se encuentra nuestro archivo mixin, y definimos dentro de la instancia de vue de nuestro componente, través de la propiedad mixins. Ver figura 33.

```

<template #tbody>
  <vs-tr v-for="(tr, i) in items" :key="i">
    <vs-td>{{ i + 1 }}</vs-td>
    <vs-td>{{ tr.numero_documento }}</vs-td>
    <vs-td>{{ tr.nombres }}</vs-td>
    <vs-td> {{ tr.apellidos }}</vs-td>
    <vs-td>{{ tr.numero_celular }}</vs-td>
    <vs-td>{{ tr.correo }}</vs-td>
    <vs-td> ...
  </vs-tr>
</template>
<template #footer> ...
</template>
</vs-table>
</ui-card>

<vs-dialog v-model="popup" width="500px" prevent-close> ...
</vs-dialog>

<ConfirmDelete ...
/>
</div>
</template>
<script>
import basic from '@/mixins/basic'
export default {
  mixins: [basic],
  data: () => ({
    options: [
      { name: 'DNI', value: 'DNI' },
      { name: 'Pasaporte', value: 'PASAPORTE' },
    ],
  }),
  computed: {
    path() {
      return 'personas'
    }
  }
}

```

Figura 33. Uso de mixin en componentes - Elaboración propia.

Pages: En esta carpeta se definen las páginas del sistema web, que gracias a Nuxt nos olvidamos de hacer configuraciones de las rutas, ya que los genera automáticamente con la configuración de *vue-router* según la estructura de la carpeta que tenemos en *pages*, es decir transforma las carpetas o archivos en rutas que nuestro navegador podrá mostrarlo sin ningún problema, y con todo esto nos ahorramos el entrar a un archivo de configuración y agregar cada vez que queramos mostrar una página. Ver figura 34.

```
control-celular-vuesax4 > .nuxt > JS router.js > ...
24
25 export const routerOptions = {
26   mode: 'history',
27   base: '/',
28   linkActiveClass: 'nuxt-link-active',
29   linkExactActiveClass: 'nuxt-link-exact-active',
30   scrollBehavior,
31
32   routes: [{
33     path: "/asignaciones",
34     component: _5dd416ec,
35     name: "asignaciones"
36   }, {
37     path: "/campanna",
38     component: _2c008272,
39     name: "campanna"
40   }, {
41     path: "/campannapolitica",
42     component: _09b47441,
43     name: "campannapolitica"
44   }, {
45     path: "/devoluciones",
46     component: _6433b272,
47     name: "devoluciones"
48   }, {
49     path: "/empresa",
50     component: _c9a29a7a
```

Figura 34. routing en Nuxt.js - Elaboración propia.

Plugins: Esta carpeta está designada para agregar librerías externas de JavaScript al proyecto, el mismo Vuesax está configurada en la carpeta. Nuxt usa Axios para hacer solicitudes http, además trabaja en conjunto con el módulo de auth, por eso para implementar la autenticación con nuxt no se tiene que hacer nada, sólo hacer una simple configuración en el archivo principal de nuxt.

En esta carpeta agregamos un interceptor de axios, este interceptor emite eventos cuando una solicitud http responde con éxito o con un error. En esos eventos que son emitidos agregamos las notificaciones que el sistema muestra cuando se realiza una operación y así el usuario puede ver si su operación fue exitosa o si ocurrió algún error.

Store: Contiene archivos del Vuex Store. En el proyecto es usado para tres funciones principales, para manejar el estado del usuario autenticado, para manejar las notificaciones y para datos globales del sistema como el menú.

nuxt.config.js: Es el archivo de configuración principal de Nuxt, en donde se define como trabaja nuestro framework y extender funcionalidades, ver figura 35. A continuación se describen algunas configuraciones en el archivo.

```
bl-celular-vuesax4 > JS nuxt.config.js > default > head > meta

export default {
  /* ...
  ssr: false,
  /* ...
  target: 'server',
  /* ...
  head: {
    title: process.env.npm_package_name || '',
    meta: [
      { charset: 'utf-8' },
      { name: 'viewport', content: 'width=device-width, initial-scale=1' },
      {
        hid: 'description',
        name: 'description',
        content: process.env.npm_package_description || '',
      },
    ],
    link: [{ rel: 'icon', type: 'image/x-icon', href: '/favicon.ico' }],
  },
  /* ...
  css: [
    'vuesax/dist/vuesax.css',
    'boxicons/css/boxicons.min.css', // https://boxicons.com/
    '@assets/scss/base.scss',
  ],
  /* ...
  plugins: [
    '@plugins/vuesax',
    '@plugins/mask',
    '@plugins/axios',
    '@plugins/chart',
    '@plugins/components',
  ],
  /* ...
  components: false,
  /* ...
  buildModules: [
    // Doc: https://github.com/nuxt-community/eslint-module
    '@nuxtjs/eslint-module',
  ],
}
```

Figura 35. Archivo nuxt.config.js - Elaboración propia.

Con el módulo Nuxt Auth, agregamos la autenticación al sistema web, agregamos `@nuxtjs/auth` dependencia usando yarn, seguidamente agregamos `@nuxtjs/auth` y `@nuxtjs/axios` a la modules sección de `nuxt.config.js`, matriz para agregar el módulo. Agregamos `baseURL` a `axios` propiedad para establecer la URL base de nuestra API. La integración con `Axios` nos permite establecer automáticamente la URL base, un cliente HTTP basado en promesas. Ver figura 36.

```

auth: {
  redirect: {
    login: '/login',
    logout: '/',
    callback: '/login',
    home: '/',
  },
  strategies: {
    local: {
      token: {
        property: 'token',
        // global: true,
        required: true,
        type: 'Bearer',
      },
      user: {
        property: '',
        autoFetch: true,
      },
      endpoints: {
        login: { url: 'users/login', method: 'post' },
        // logout: { url: '/api/auth/logout', method: 'post' },
        user: { url: 'users/me', method: 'get' },
      },
    },
  },
},
axios: {
  baseURL: process.env.BASE_URL || 'http://localhost:3000/',
  // baseURL: 'https://tesisappcontrol.mybluemix.net/',
  // baseURL: 'https://tesisapp.cfapps.io/',
},

```

Figura 36. Archivo nuxt.config.js (Configuración de Nuxt Auth) - Elaboración propia.

Se utilizo SPA, renderizado del lado cliente, donde presentamos el contenido en el navegador mediante JavaScript. En lugar de obtener todo el contenido del HTML, solo obtenemos un documento HTML básico con un archivo JavaScript que renderizará el resto del sitio usando el navegador. Para desplegar una SPA utilizamos **ssr** como **false** y el comando **build** para construir ver figura 34.

En seguida, pasamos a mostrar las interfaces del sistema web agrupadas por módulos.

En la figura 37. Se muestra la vista principal del sistema web, en donde se tiene una estadística general de total de líneas, cantidad de Pérdida/Robo, cantidad de solo chip, líneas libres, cantidad de usuarios y la cantidad de líneas por área.

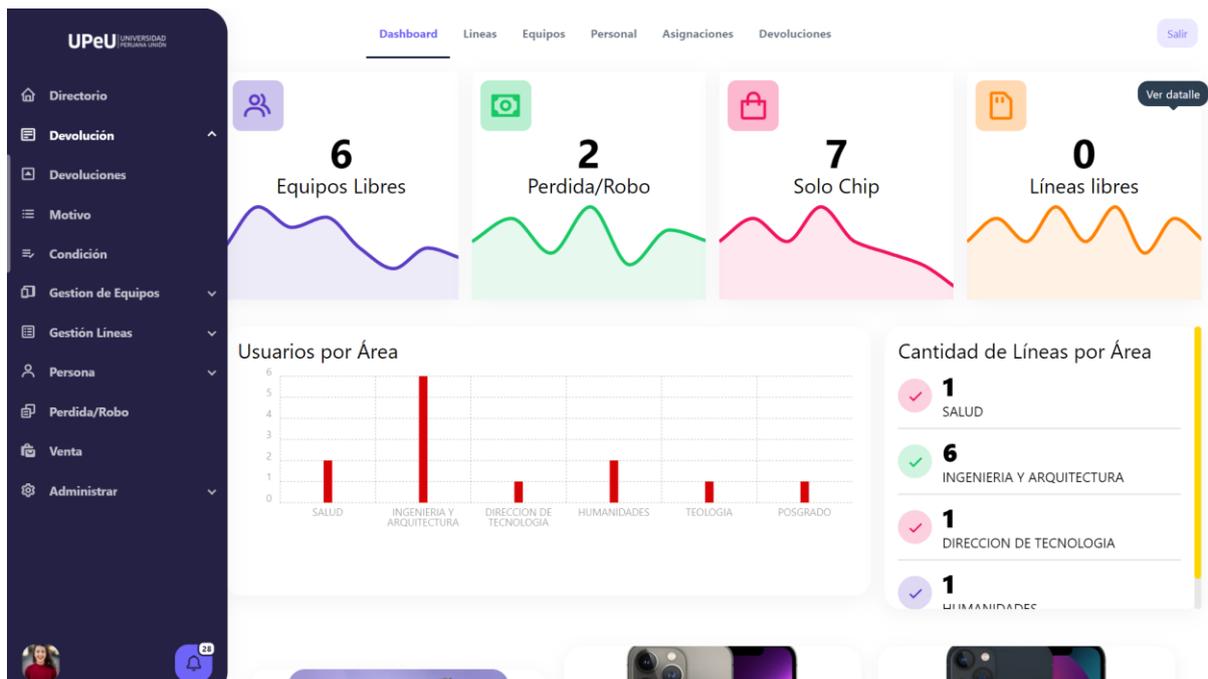


Figura 37. Pantalla principal del sistema web – Elaboración propia

Módulo Administrar

En este módulo se configura Empresa, Campaña a la que pertenecerán los equipos, Política que se maneja para cada campaña, a continuación, se muestra cada la vista del módulo administrar, ver figura 38.

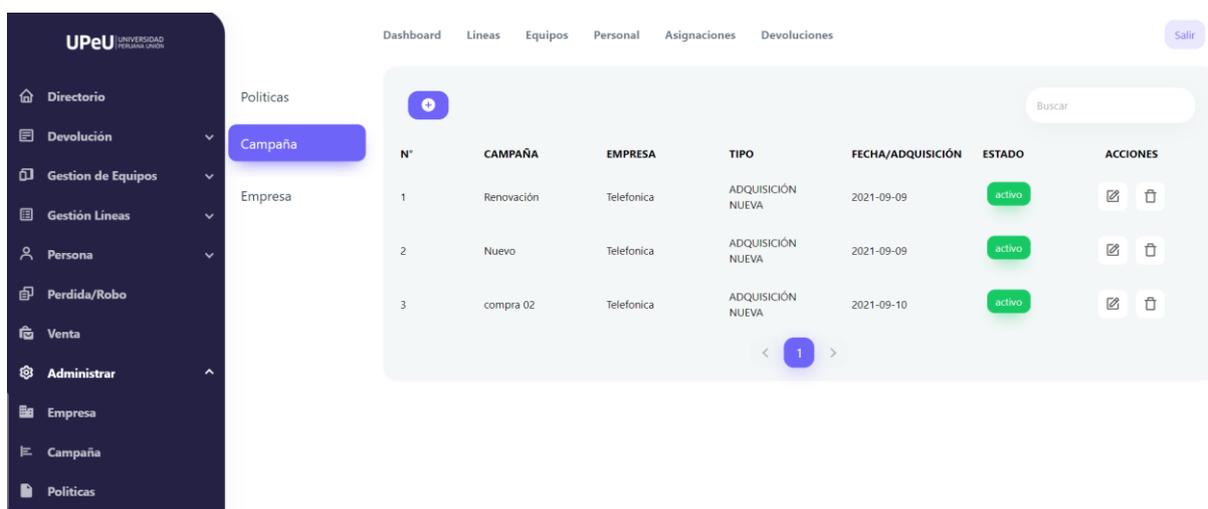


Figura 38. Administrar campaña - Elaboración propia.

Módulo Gestión de Personas

En este módulo es donde se gestiona las personas y los accesos al sistema web, así mismo la se configura Área y Cargo, al que pertenecerá cada persona, ver figura 39.

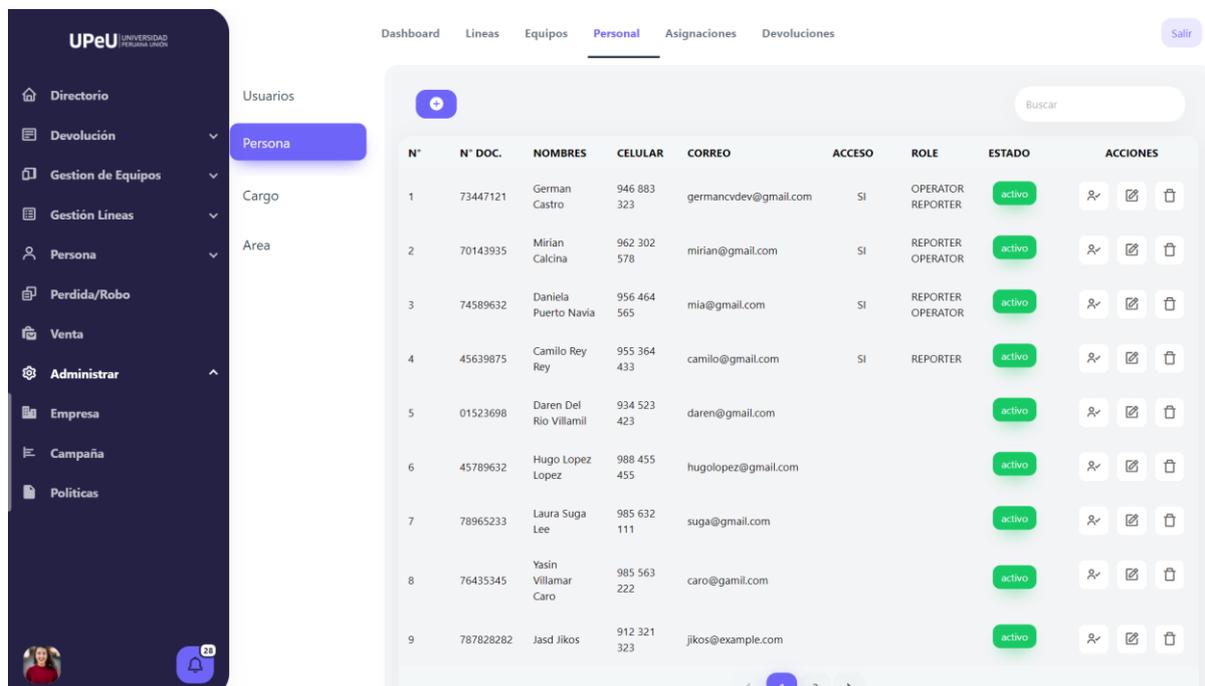


Figura 39. Módulo de gestión de persona - Elaboración propia.

Módulo Gestión de Líneas

En este módulo se gestiona el Operador al que pertenece el Plan, SIMCARD, Plan y Líneas. A continuación, se muestra las vistas del módulo.

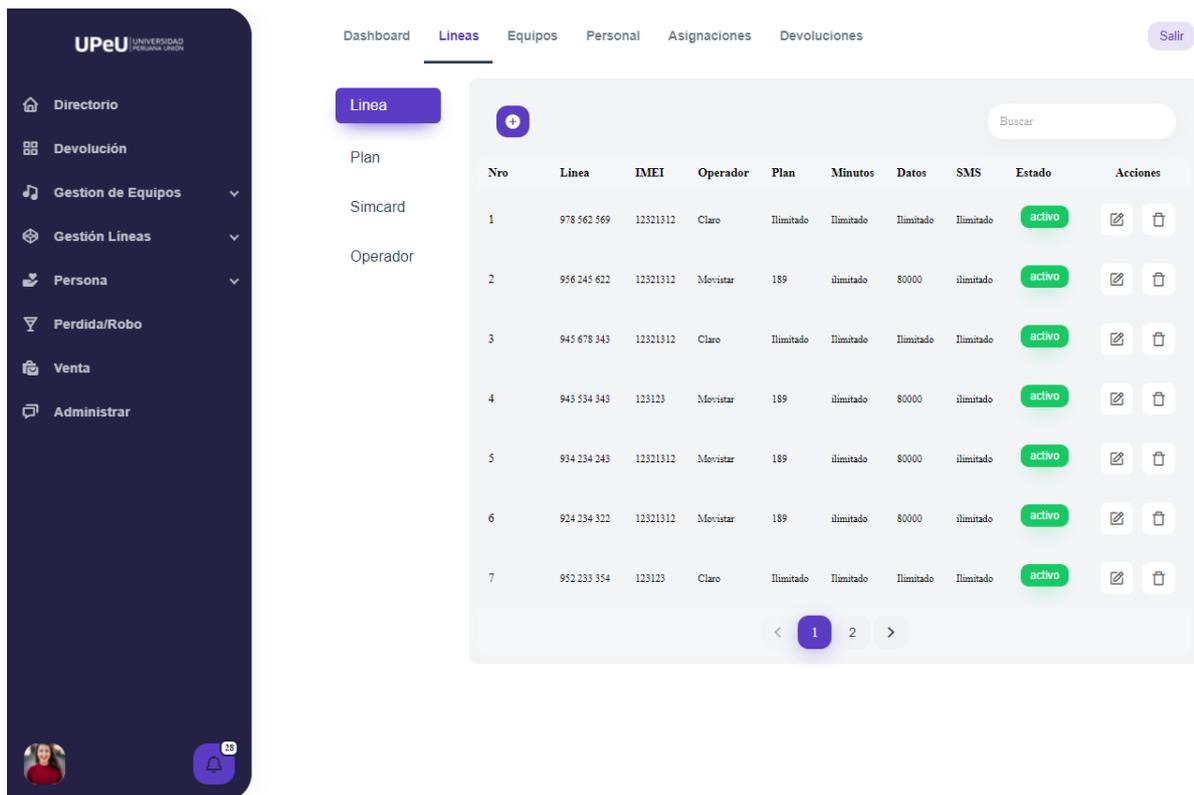


Figura 40. Gestión de líneas - Elaboración propia.

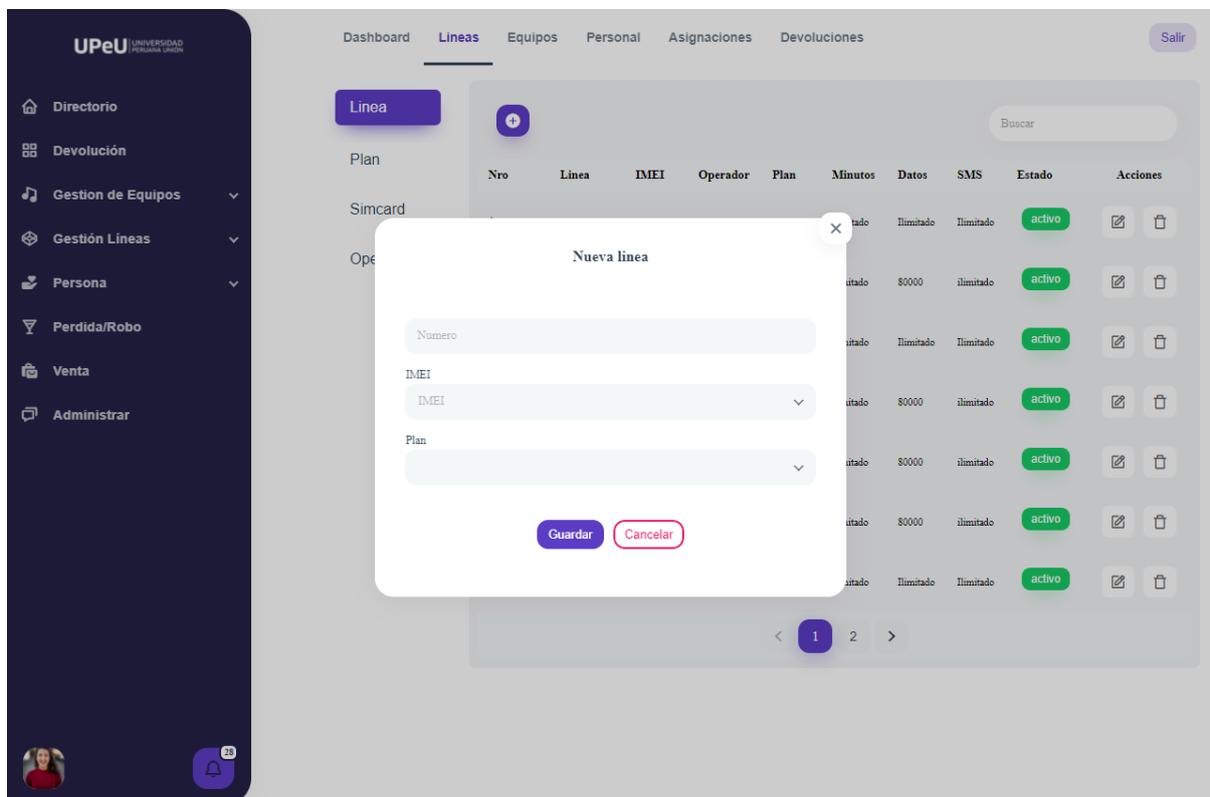
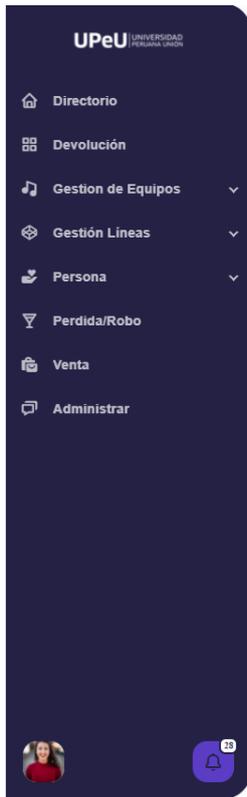


Figura 41. Agregar una línea - Elaboración propia.



Dashboard **Lineas** Equipos Personal Asignaciones Devoluciones Salir

Linea +

Plan

Simcard

Operador

Nro	Plan	Operador	Megas	Minutos	SMS	Inicio de ciclo	Estado	Acciones
1	189	Movistar	80000	ilimitado	ilimitado	2020-10-20	activo	
2	ilimitado	Claro	ilimitado	ilimitado	ilimitado	2020-10-26	activo	
3	160	Entel	12312	321312	321312	2021-06-20	activo	

< 1 >

Figura 42. Agregar planes - Elaboración propia.

Módulo Gestión de Equipos

En este módulo se gestiona el equipo, el cual tiene, un Modelo, Marca y sus respectivos Accesorios. A continuación, se muestra las vistas del módulo.

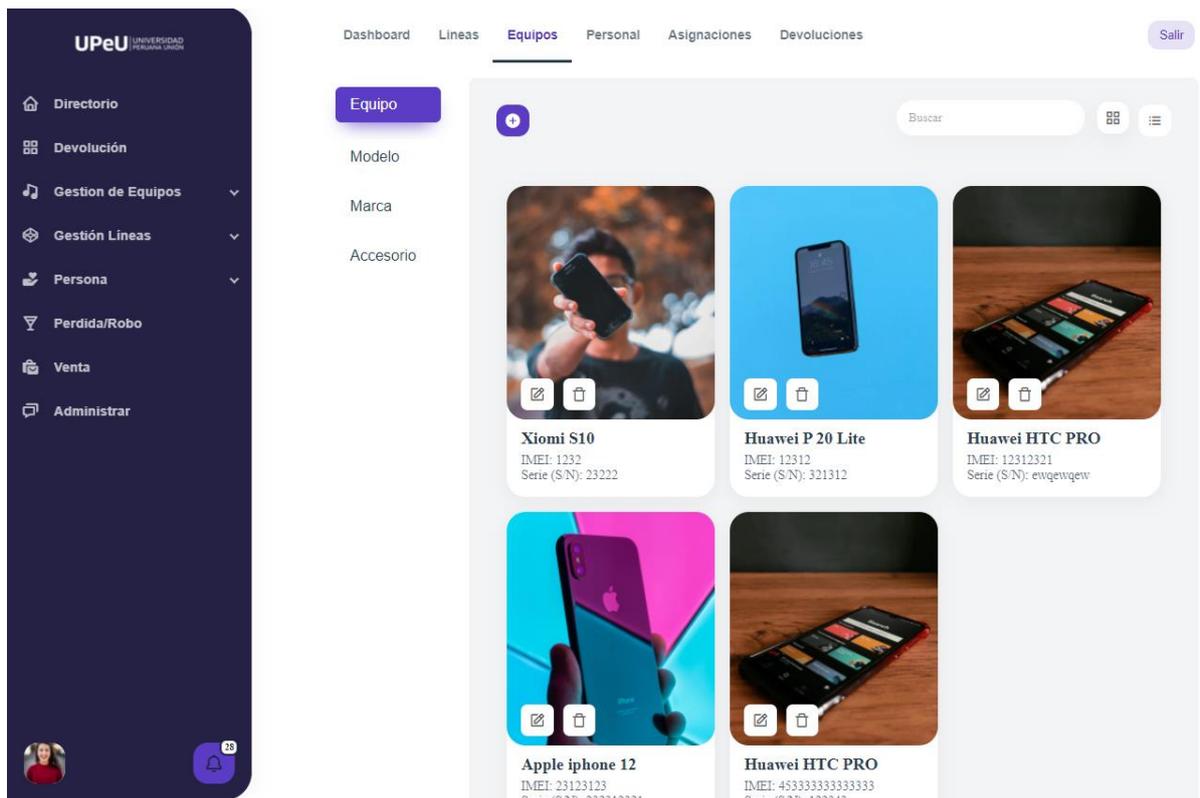


Figura 43. Gestión de equipos - Elaboración propia.

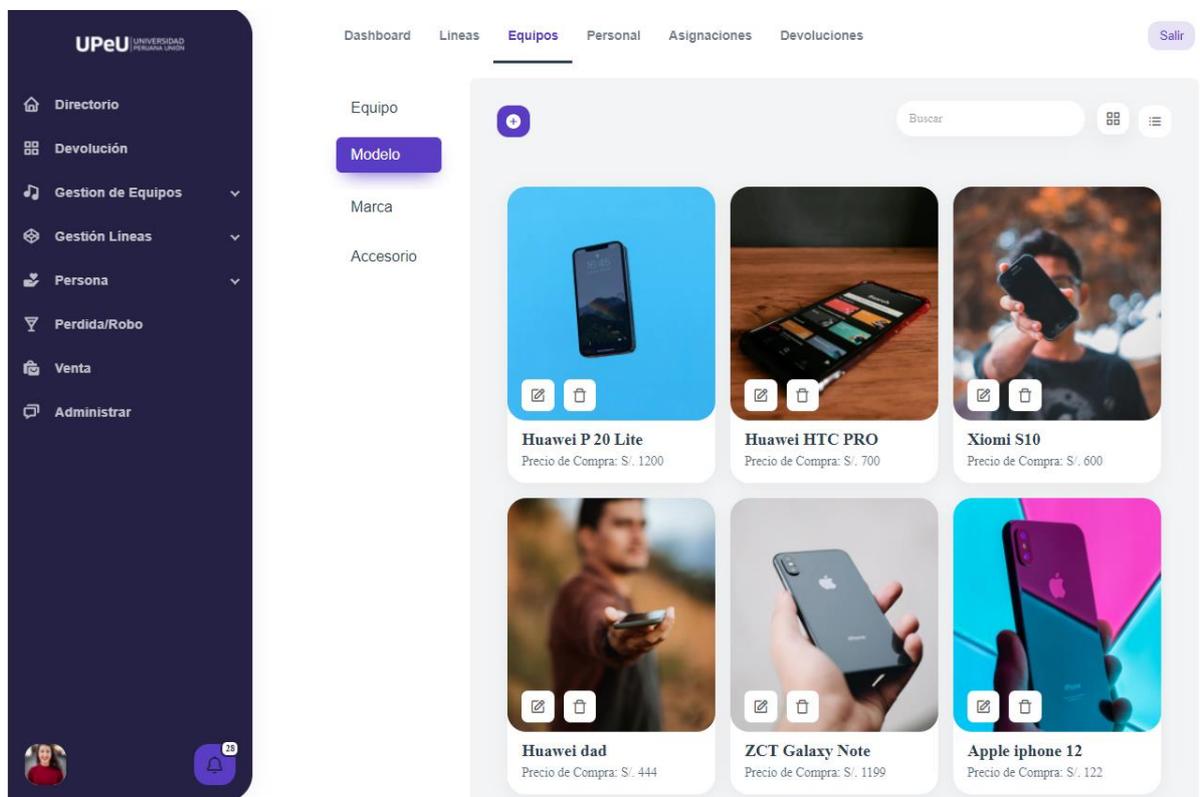


Figura 44. Agregar modelo - Elaboración propia.

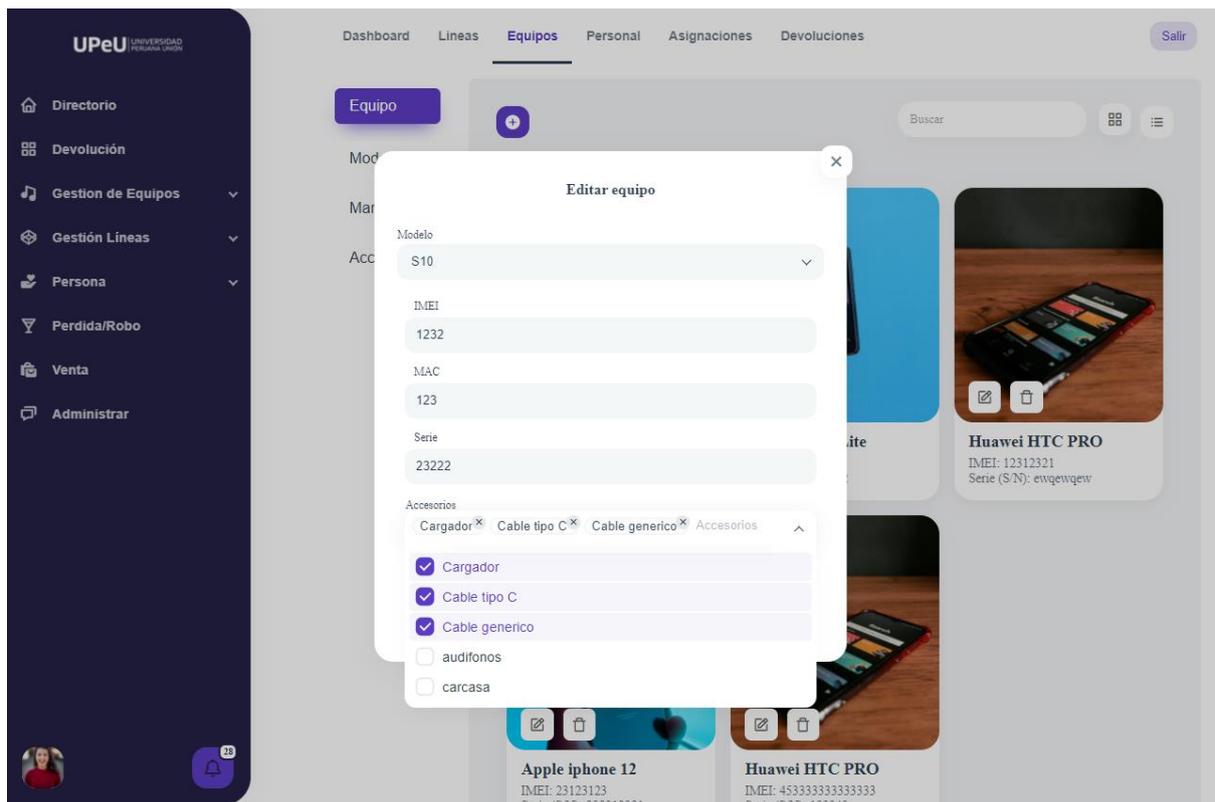


Figura 45. Nuevo equipo - Elaboración propia

Asignaciones

En esta vista se gestiona las asignaciones de Línea y/o Equipo según corresponda, a un personal según el cargo. Generando un formato de entrega de asignación. Ver figura 45.

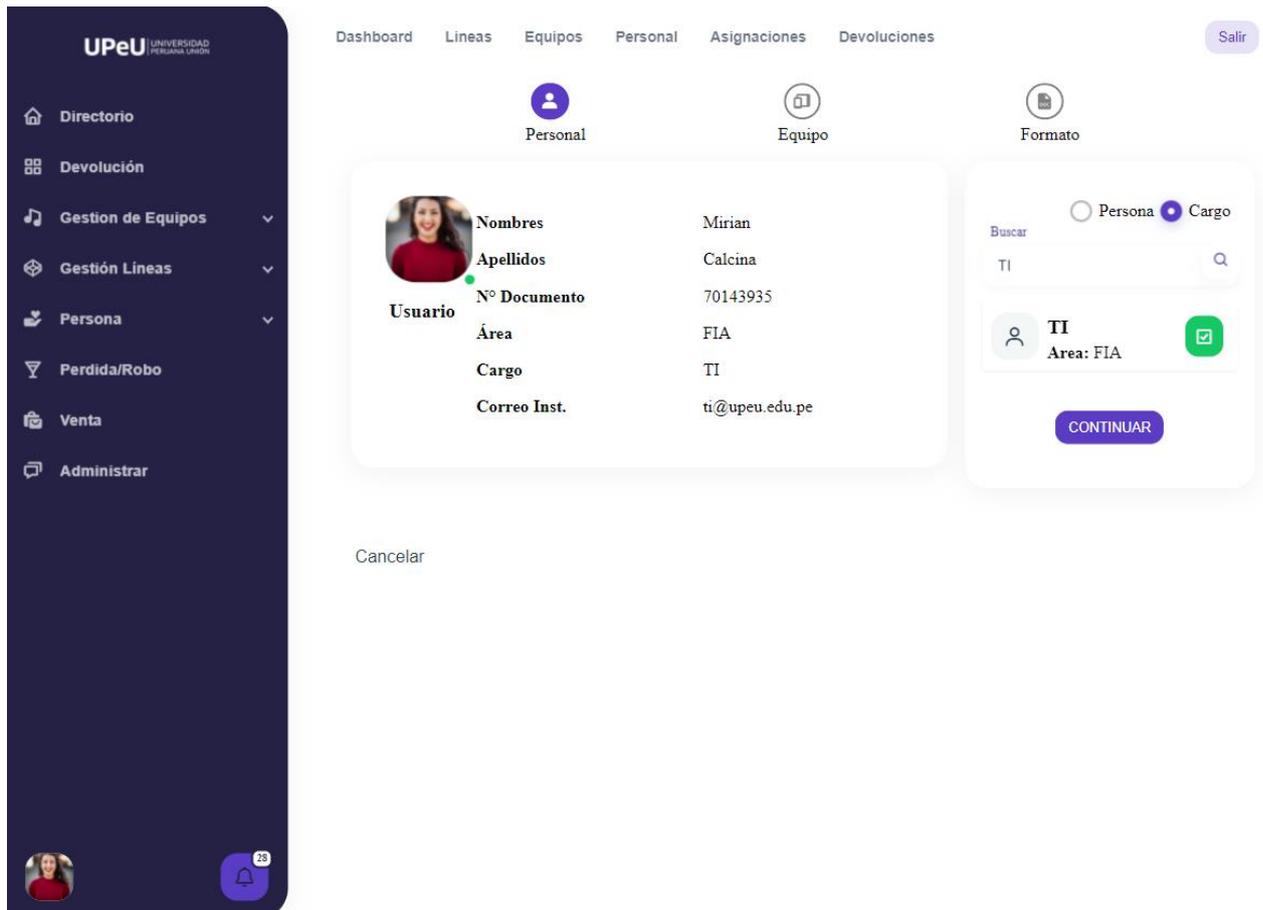


Figura 46. Asignación de línea y/o equipo - Elaboración Propia.

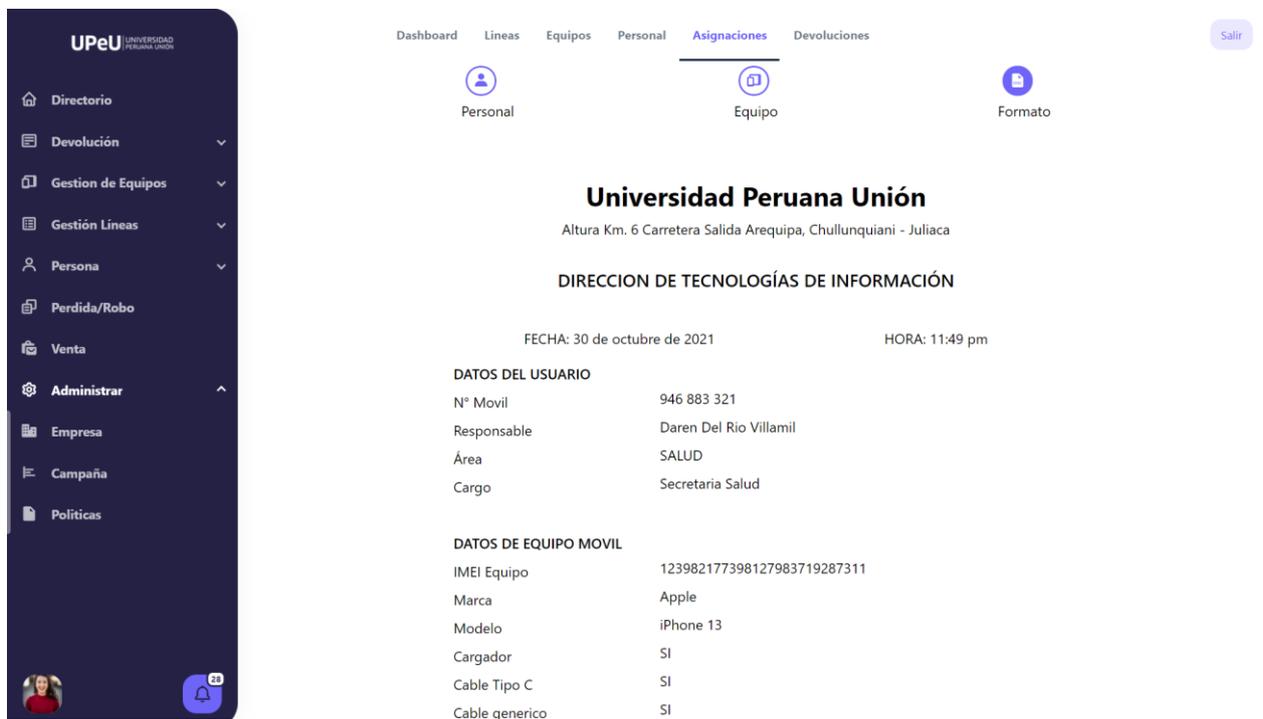


Figura 47. Formato de asignación de línea y/o equipo - Elaboración Propia.

3.6.4. Pruebas

3.6.4.1. Pruebas de aceptación.

Las pruebas de aceptación son derivadas a partir de las historias de usuarios, en estas se definieron criterios de aceptación para completar los objetivos del proyecto.

Tabla 21. Prueba de aceptación de la historia número 1 – Elaboración propia.

Gestionar personas			
ID Prueba:	PA01	Versión de SW:	1.0
Objetivo de la prueba:	Se realizaron las siguientes pruebas: <ul style="list-style-type: none">• Una persona vinculada a cualquier actividad del sistema no puede ser eliminada.• Verificar que no haya duplicidad de una misma persona mediante el número de Documento de Identidad.		
Requerimientos relacionados:	RF02	Historia de usuario relacionado:	USHIS001
Desarrollo de la prueba			
Pasos	Una vez autenticado en el sistema web, debe ir al menú Gestionar Persona, opción Persona, se mostrará una lista de personas que estén registradas en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar una nueva persona, llenar los campos y clic en Guardar, teniendo en cuenta que la persona no exista en el sistema, pero si el DNI que desea registrar ya se encuentra en el sistema, mostrará una notificación, indicando que ya se encuentra registrada. Para eliminar una persona se debe seleccionar el icono Eliminar de acciones en la tabla que lista todo el registro, pero si está persona se encuentra vinculado con registro de actividad pueda mostrar una notificación.		
Comportamiento esperado	El sistema pueda realizar exitosamente la función de validar los registrados y no registrados, listar, agregar, editar y eliminar, si se le está permitido.		
Resultados de la prueba	Si se agrega una persona y los campos son correctos actualiza la base de datos y se muestra la nueva persona en la lista, en caso de que ocurra un error en la validación de los campos se muestra una notificación indicando de que existe un registro o que campo es incorrecto. Si se desea eliminar una persona, muestra un mensaje para confirmar la acción, se actualiza la base de datos, los cambios se muestran en la lista, teniendo en cuenta de que, si este registro a eliminar tiene una vinculación relacionada a una actividad, mostrará una notificación.		
Evaluación de la prueba: Prueba satisfactoria			

Observaciones:	
-----------------------	--

Tabla 22. Prueba de aceptación de la historia número 2 – *Elaboración propia.*

Gestionar roles y permisos para dar acceso al usuario al sistema.			
ID Prueba:	PA02	Versión de SW:	1.0
Objetivo de la prueba:	Se realizaron las siguientes pruebas: <ul style="list-style-type: none"> • Dar acceso al sistema a un usuario determinado asignado un rol. • Permitir el ingreso al sistema según el rol brindado. 		
Requerimientos relacionados:	RF01, RF02	Historia de usuario relacionado:	USHIS002
Desarrollo de la prueba			
Pasos	Una vez autenticado en el sistema web, debe ir al menú Gestionar Persona, opción Persona, se mostrará una lista de personas que estén registradas en el sistema. Para brindar un acceso debe ubicarse en el icono Brindar acceso de acciones, donde mostrará la opción de un Nuevo usuario, tomando el correo como usuario y poder crear una contraseña y seleccionar el rol a asignar según el usuario. Validando el acceso: Colocar su usuario y contraseña, hacer clic en el botón Ingresar.		
Comportamiento esperado	El sistema pueda realizar exitosamente la función de dar un acceso al sistema según el rol asignado.		
Resultados de la prueba	Si se agrega un acceso y los campos son correctos actualiza la base de datos y se muestra el nuevo usuario en la lista, opción Usuarios. En caso de que ocurra un error en la validación de los campos se muestra una notificación indicando de que existe un registro o que campo es incorrecto. Ingreso correctamente al sistema.		
Evaluación de la prueba: Prueba satisfactoria			
Observaciones:			

Tabla 23. Prueba de aceptación de la historia número 3 – *Elaboración propia.*

Gestionar líneas			
ID Prueba:	PA03	Versión de SW:	1.0
Objetivo de la prueba:	Se realizaron las siguientes pruebas: <ul style="list-style-type: none"> • Un plan debe pertenecer a un operador. • Una línea debe pertenecer a un plan y a un simcard. • NO puede eliminarse un registro si se tiene al menos una asignación de equipo. 		

Requerimientos relacionados:	RF03	Historia de usuario relacionado:	USHIS003
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Gestionar Líneas, opción Plan, se mostrará una lista de planes que estén registrados en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar una nuevo Plan, llenar los campos en donde se tenga un campo para seleccionar el Operador (Muestre todos los Operadores registrados) y clic en Guardar.</p> <p>Una vez autenticado en el sistema web, debe ir al menú Gestionar Líneas, opción Línea, se mostrará una lista de líneas que estén registradas en el sistema. Clic en + para agregar una nueva, se mostrará una vista con los campos requeridos para agregar una nueva Línea, llenar los campos en donde se tenga una un campo para seleccionar el Plan (Muestre todos los Planes registrados) y clic en Guardar.</p> <p>En caso de pérdida o robo de la Línea, Clic en Editar, en la opción de Recuperar, donde se muestran todos los IMEIs registrados.</p> <p>Para eliminar una Línea se debe seleccionar el icono Eliminar de acciones en la tabla que lista todo el registro, pero si está Línea se encuentra vinculado con registro de actividad se mostrará una notificación.</p>		
Comportamiento esperado	El sistema pueda realizar un registro exitoso, teniendo en cuenta los campos requeridos. Validando si un registro se encuentra vinculado.		
Resultados de la prueba	<p>Si se Agrega una Línea o Plan y los campos son correctos actualiza la base de datos y se muestra la nueva línea o plan en la lista.</p> <p>Si se desea eliminar un registro, muestra un mensaje para confirmar la acción, se actualiza la base de datos, los cambios se muestran en la lista, teniendo en cuenta de que, si este registro a eliminar tiene una vinculación relacionada a una actividad, mostrará una notificación.</p>		
Evaluación de la prueba: Prueba satisfactoria			
Observaciones:			

Tabla 24. Prueba de aceptación de la historia número 4 – Elaboración propia.

Gestionar equipo			
ID Prueba:	PA04	Versión de SW:	1.0
Objetivo de la prueba:	<p>Se realizaron las siguientes pruebas:</p> <ul style="list-style-type: none"> • Un equipo debe pertenecer a un modelo. • Un modelo debe estar asociado con una marca. • Un equipo puede tener más de un accesorio. 		

	<ul style="list-style-type: none"> • Un modelo debe estar sujeto a políticas en caso de pérdida, robo o cualquier daño ocasionado sobre un equipo en una determinada campaña. • NO puede eliminarse un registro si se tiene al menos una asignación. 		
Requerimientos relacionados:	RF04	Historia de usuario relacionado:	USHIS004
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Gestionar equipo, opción Equipo, se mostrará una lista de equipos que estén registrados en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar un nuevo equipo, llenar los campos en donde se tenga un campo para seleccionar el Modelo (Muestre todos los Modelos registrados según la Marca), un campo Accesorios (Muestre todos los accesorios registrados y que por defecto esté seleccionado tres accesorios) y clic en Guardar.</p> <p>Una vez autenticado en el sistema web, debe ir al menú Gestionar Equipo, opción Modelo, se mostrará una lista de modelos que estén registrados en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar una nuevo Modelo, llenar los campos en donde se tenga una un campo para seleccionar el Marca (Muestre todas las Marcas registradas), un campo Campaña (Muestre todas las campañas activas registradas que están asociadas directamente a una política) y proceder a guardar correctamente.</p> <p>Para eliminar un registro se debe seleccionar el icono Eliminar de acciones en la tabla que lista todo el registro, pero si el registro a eliminar se encuentra vinculado con registro de actividad pueda mostrar una notificación.</p>		
Comportamiento esperado	El sistema pueda realizar un registro exitoso, teniendo en cuenta los campos requeridos. Validando si un registro se encuentra vinculado para no eliminar un registro.		
Resultados de la prueba	<p>Si se agrega un nuevo registro y los campos son correctos actualiza la base de datos y se muestra el nuevo registro en la lista.</p> <p>Si se desea eliminar un registro, muestra un mensaje para confirmar la acción, se actualiza la base de datos, los cambios se muestran en la lista, teniendo en cuenta de que, si este registro a eliminar tiene una vinculación relacionada a una actividad, mostrará una notificación.</p>		
Evaluación de la prueba: Prueba satisfactoria			
Observaciones:			

Tabla 25. Prueba de aceptación de la historia número 5 – Elaboración propia.

Asignación de línea y/o equipo a un cargo			
ID Prueba:	PA05	Versión de SW:	1.0

Objetivo de la prueba:	Asignación de línea y/o equipo a un cargo.		
Requerimientos relacionados:	RF03, RF04	Historia de usuario relacionado:	USHIS003, USHIS004
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Asignaciones, se mostrará una lista de asignaciones que estén registrados en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar uno nuevo:</p> <ol style="list-style-type: none"> 1. Buscar el cargo y seleccionar. Clic en CONTINUAR 2. Buscar el equipo por IMEI o modelo y seleccionar. Mostrará la información del equipo y los accesorios a asignar. 3. Buscar la línea según el número y seleccionar. Clic en CONTINUAR 4. La siguiente vista es el Formato generado con todos los datos, y un campo OBSERVACIONES donde puede agregar alguna OBS. <p>El campo equipo o Línea dependerá de la entrega a realizar.</p>		
Comportamiento esperado	El sistema pueda realizar una asignación exitosamente con todo lo requerido para una asignación de equipo y/o línea a un cargo.		
Resultados de la prueba	Si se agrega un nuevo registro y los campos son correctos actualiza la base de datos y se muestra el nuevo registro en la lista.		
Evaluación de la prueba: Prueba satisfactoria			
Observaciones:			

Tabla 26. Prueba de aceptación de la historia número 6 – Elaboración propia.

Entrega de línea y/ o equipos			
ID Prueba:	PA06	Versión de SW:	1.0
Objetivo de la prueba:	Entrega de línea y/ o equipos		
Requerimientos relacionados:	RF02, RF03, RF04	Historia de usuario relacionado:	USHIS001, USHIS003, USHIS004, USHIS005
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Asignaciones, se mostrará una lista de asignaciones que estén registrados en el sistema. Clic en + para agregar uno nuevo, se mostrará una vista con los campos requeridos para agregar uno nuevo:</p> <ol style="list-style-type: none"> 1. Buscar a la persona por DNI, nombres, apellidos y seleccionar. 2. Buscar el cargo y seleccionar. Clic en CONTINUAR 		

	<ol style="list-style-type: none"> 3. Buscar el equipo por IMEI o modelo y seleccionar. Mostrará la información del equipo y los accesorios a asignar. 4. Buscar la línea según el número y seleccionar. Clic en CONTINUAR 5. La siguiente vista es el Formato generado con todos los datos, y un campo OBSERVACIONES donde puede agregar alguna OBS, clic en Imprimir. <p>El campo equipo o Línea dependerá de la entrega a realizar.</p> <p>Si ya se encuentra un registro de la línea o equipo a asignar: Clic en acciones, Editar:</p> <ol style="list-style-type: none"> 1. Buscar a la persona por DNI, nombres, apellidos y seleccionar. 2. Clic en Formato, agregar si es necesario una OBS e Imprimir.
Comportamiento esperado	El sistema pueda realizar una asignación exitosamente con todo lo requerido para una asignación y finalmente generar un formato para imprimir.
Resultados de la prueba	Si se agrega un nuevo registro y los campos son correctos actualiza la base de datos y se muestra el nuevo registro en la lista.
Evaluación de la prueba: Prueba satisfactoria	
Observaciones:	

Tabla 27. Prueba de aceptación de la historia número 7 – Elaboración propia.

Gestionar las devoluciones			
ID Prueba:	PA07	Versión de SW:	1.0
Objetivo de la prueba:	Realizar una devolución de un equipo y/o línea.		
Requerimientos relacionados:	RF05, RF06	Historia de usuario relacionado:	USHIS004, USHIS005
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Devolución/Devoluciones, se mostrará una lista de devoluciones que estén registrados en el sistema. Para realizar una devolución:</p> <ol style="list-style-type: none"> 1. Buscamos (Búsqueda por número de documento de identidad, número de línea) o nos ubicamos en registro a realizar la devolución 2. Clic en acciones, en el botón hacer devolución o dar de baja. 3. Seleccionamos hacer devolución. 4. Editar los accesorios de según la devolución, 5. Seleccionamos la condición y el motivo. 6. Colocamos una observación si fuera necesario. 7. Clic en Guardar. 		

	Una vez guardado se mostrará en la lista de registros de devoluciones, donde puede visualizar a detalle en acciones, ver detalle.
Comportamiento esperado	El sistema pueda realizar una devolución exitosamente.
Resultados de la prueba	Si se agrega un nuevo registro y los campos son correctos actualiza la base de datos y se muestra el nuevo registro en la lista de devoluciones.
Evaluación de la prueba: Prueba satisfactoria	
Observaciones:	

Tabla 28. Prueba de aceptación de la historia número 8 – Elaboración propia.

Gestionar pérdidas y robos			
ID Prueba:	PA08	Versión de SW:	1.0
Objetivo de la prueba:	Registro de pérdida y/o robos		
Requerimientos relacionados:	RF06	Historia de usuario relacionado:	USHIS006
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, debe ir al menú Devolución/Devoluciones, se mostrará una lista de devoluciones que estén registrados en el sistema. Para realizar un registro de pérdida o robo:</p> <ol style="list-style-type: none"> 1. Buscamos (Búsqueda por número de documento de identidad, número de línea) o nos ubicamos en registro. 2. Clic en acciones, en el botón hacer devolución o dar de baja. 3. Seleccionamos Dar de baja. 4. Seleccionamos según sea el caso PÉRDIDA O ROBO 5. Colocamos una observación si fuera necesario. 6. Clic en Guardar. <p>Una vez guardado se mostrará en la lista de registros de devoluciones, donde puede visualizar el registro con estado PENDIENTE. Para aplicar la política clic en acciones, ver detalle. En esta vista nos mostrará la información detallada del equipo:</p> <ol style="list-style-type: none"> 1. Clic en Aplicar, según sea el caso por pérdida o robo. 2. Clic en Elegir una imagen (Para poder adjuntar el voucher de pago) 3. Clic en CERRAR DEVOLUCIÓN para concluir con el registro. 		
Comportamiento esperado	El sistema pueda realizar un registro por pérdida o robo exitosamente.		
Resultados de la prueba	Si se agrega un nuevo registro y los campos son correctos actualiza la base de datos y se muestra el nuevo registro en la lista de devoluciones.		
Evaluación de la prueba: Prueba satisfactoria			

Observaciones:	
-----------------------	--

Tabla 29. Prueba de aceptación de la historia número 9, 10 – Elaboración propia.

Mostrar estadísticas generales y visualización de reportes			
ID Prueba:	PA09, PA10	Versión de SW:	1.0
Objetivo de la prueba:	Mostrar un Dashboard en la vista principal y al realizar un clic sobre los datos pueda mostrarme reporte a detalle.		
Requerimientos relacionados:	RF01, RF03, RF04, RF05, RF06, RF07, RF08	Historia de usuario relacionado:	USHIS001, USHIS003, USHIS004, USHIS005, USHIS006, USHIS007, USHIS008
Desarrollo de la prueba			
Pasos	<p>Una vez autenticado en el sistema web, mostrará en la vista principal una estadística general de cantidades.</p> <ul style="list-style-type: none"> • Mostrar la cantidad total de líneas, perdida/robo, solo chip (líneas), líneas libres. • Cantidad de usuarios por área • Cantidad de líneas por área • Cantidad de equipos por modelo. • Al realizar un clic sobre los datos pueda mostrarme un reporte a detalle. 		
Comportamiento esperado	El sistema pueda mostrar la estadística general de cantidades y un reporte a detalle.		
Resultados de la prueba	Visualización de cantidad y el reporte a detalle.		
Evaluación de la prueba: Prueba satisfactoria			
Observaciones:			

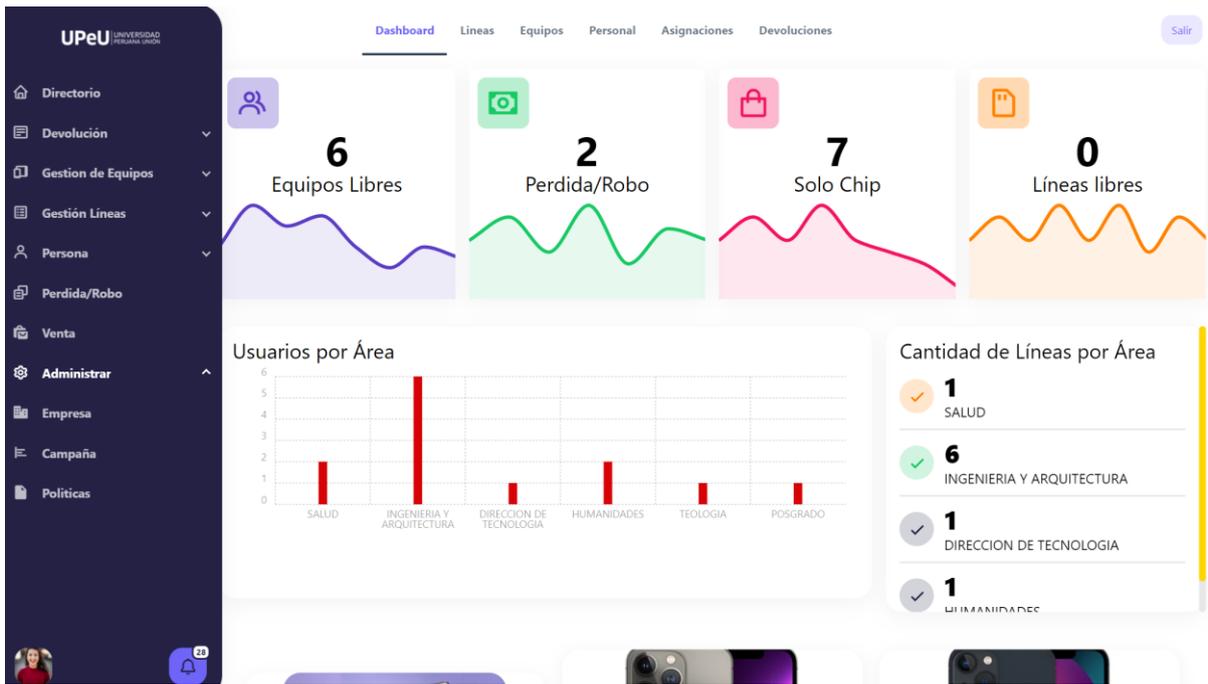


Figura 48. Dashboard - Pantalla principal del sistema web - Elaboración propia.

PERSONA	CARGO	EQUIPO	LINEA
SALUD			
Daren Del Rio Villamil daren@gmail.com	Secretaria Salud salud@ss.com	iPhone 13 Apple	946 883 321 Plan: 109
INGENIERIA Y ARQUITECTURA			
Camilo Rey Rey camilo@gmail.com	Secretaría Fia fia@example.com	A70 Samsung	988 987 123 Plan: 109
Yasin Villamar Caro caro@gamil.com	Secretaría Fia fia@example.com	MI 9 LITE Xiami	980 798 817 Plan: 189
Mirian Calcina mirian@gmail.com	Secretaría Fia fia@example.com	S6 Samsung	967 777 777 Plan: 109
Laura Suga Lee suga@gmail.com	Secretaría Fia fia@example.com	MI 9 LITE Xiami	912 526 766 Plan: 109
Mirian Calcina mirian@gmail.com	Secretaría Fia fia@example.com	s20 Samsung	978 766 722 Plan: 189
Mirian Calcina mirian@gmail.com	Secretaría Fia fia@example.com	s20 Samsung	978 766 722 Plan: 189
DIRECCION DE TECNOLOGIA			
German Castro germancvdev@gmail.com	Secretaria de DTI dti@example.com	A80 Samsung	971 662 735 Plan: 189
HUMANIDADES			

Figura 49. Visualización de reportes a detalle - Elaboración propia.

3.6.5. Despliegue

Para realizar el despliegue del sistema web se utilizó Netlily para el frontend y IBM cloud para la aplicación backend. Netlily permite vincular nuestro sistema web con Git, con el

objetivo de poder actualizar automáticamente tras cada actualización que se lleva a cabo en el repositorio, ver figura 50.

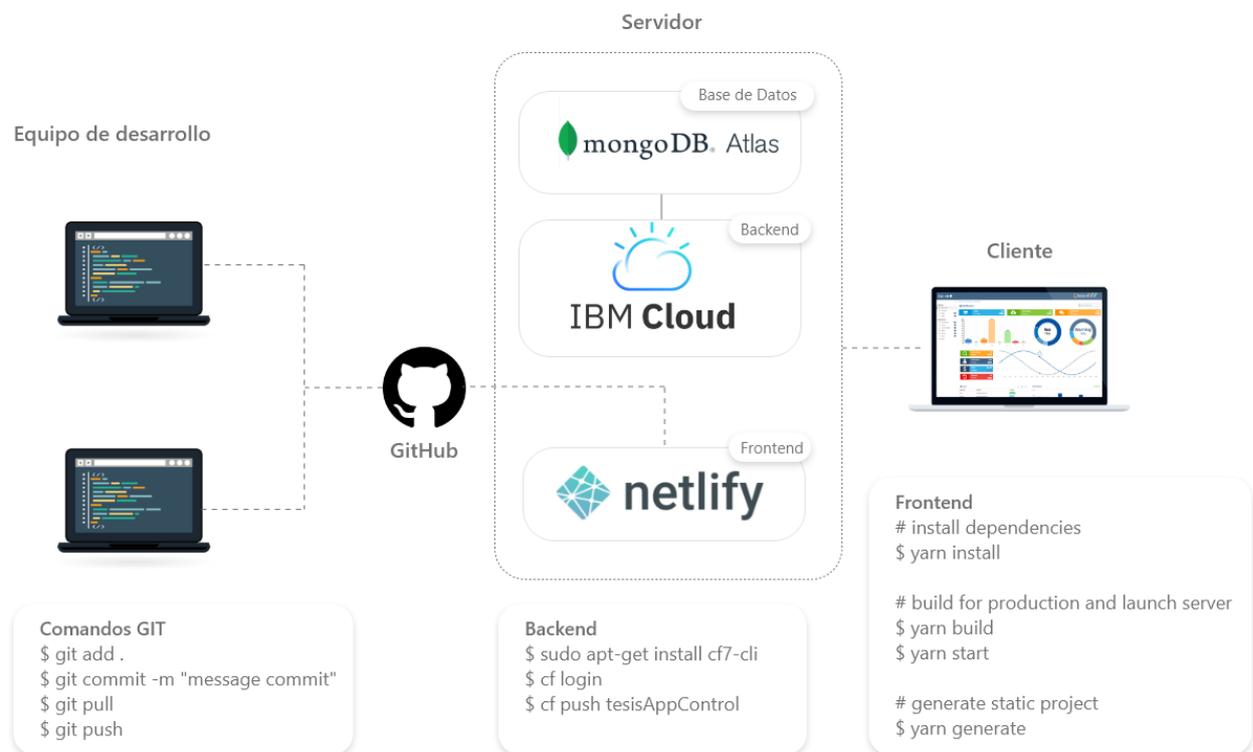


Figura 50. Diagrama de despliegue - Elaboración propia.

CAPÍTULO IV. RESULTADOS Y DISCUSIÓN

4.1. Resultado del objetivo específico 1

Para el cumplimiento de este primer objetivo de analizar los requerimientos, se aplicó la fase de planificación de XP, teniendo como resultado los requerimientos funcionales (gestión de roles y permisos, gestionar personas, gestionar líneas, asignación de líneas, entrega de líneas y/o equipos, gestionar devoluciones, gestionar equipos perdidos y robados, y mostrar estadísticas generales), no funcionales (diseño responsivo e intuitivo y responder con un tiempo aceptable a las peticiones del usuario) y las historias de usuario más sus especificaciones. Cada requerimiento e historia de usuario se obtuvo a través de la recopilación y análisis de los documentos utilizados y el archivo Excel que se tiene para la gestión y control de equipos celulares. Para ver a más detalle en el capítulo III, desarrollo del sistema y en el Anexo A.

4.1.1. Discusión

Es de esencial importancia realizar un buen análisis de requerimientos como indica en su investigación (Cenas, 2018) en el cual uno de sus objetivos planteados fue recopilar la información de los procesos para poder determinar los requerimientos. Por otro, De Liz (2020) realizó el análisis competitivo y las encuestas, para poder definir los requerimientos funcionales y no funcionales, correspondientes a las acciones fundamentales de la aplicación. Tomando en cuenta los antecedentes mencionados, en esta investigación se usó la recolección y levantamiento de la información para la definición de los requerimientos funcionales y no funcionales y para mayor entendimiento de estas se realizó las historias de usuario, permitiéndonos identificar lo esencial para poder realizar el desarrollo del sistema.

4.2. Resultado del objetivo específico 2

Con respecto al segundo objetivo específico de la investigación para el diseño del sistema web, se aplicó la fase de diseño de la metodología Programación Extrema (XP), para lo cual se elaboró el diseño del prototipo, diseño del modelado de la base de datos y arquitectura del sistema web.

XP sugiere que el diseño debe ser sencillo. Para una mejor comprensión se realizó el diseño del prototipo, como se muestra en la tabla 32, y Anexo B, ya que esta que se enfoca en el aspecto visual y la estructura de sus diferentes secciones. En base a las historias de usuario definidas, se analizó cuáles son las funcionalidades del sistema web, se realizó el diseño del modelado de la base datos, ver figura 16. Teniendo el análisis de las historias de usuario, modelado de la base de datos y prototipos, se diseñó la arquitectura del sistema, cliente-servidor, ver figura 51.

Tabla 30. *Prototipos realizados - Elaboración propia.*

Prototipo	Diseñando en base a los requerimientos	Validado por el cliente
Login del sistema	✓	✓
Pantalla principal	✓	✓
Gestión de equipos	✓	✓
Gestión de líneas	✓	✓
Gestión de personas	✓	✓
Entrega de línea y/ o equipos	✓	✓
Pérdida y/o robo	✓	✓
Gestión de ventas	✓	✓
Gestión de devoluciones	✓	✓
Configuración (Empresa, Campaña, política, área y cargo)	✓	✓

En la siguiente figura, se aprecia la arquitectura del sistema, cliente-servidor, con LoopBack, podemos conectarnos con una fuente de datos como, por ejemplo, una base de datos o un API REST para su consumo y luego exponer la aplicación como un API REST creando una definición de modelo. Un modelo de LoopBack define los datos de aplicación, las reglas de validación, las prestaciones de acceso a datos y la lógica de negocio de un API, y proporciona un API REST de forma predeterminada. Una vez expuestas se consume con el framework Nuxtjs mediante la solicitud/respuesta de HTTP.

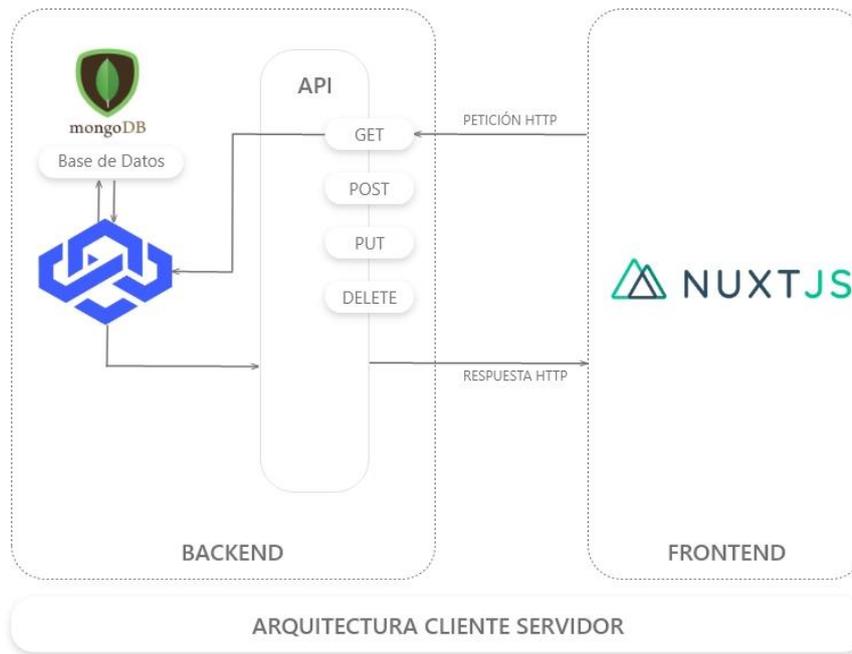


Figura 51. Arquitectura de sistema web – Elaboración propia.

4.2.1. Discusión

Otro de los propósitos de la presente investigación fue realizar el diseño De Liz (2020) realizó el diseño de la arquitectura de sistema, una arquitectura Baas (Backend as a Service), el cual consistió en implementar el backend de la aplicación utilizando un proveedor externo, a través de la utilización de su API, que hace de puente entre el frontend, y para la implementación del frontend utilizó el framework Nuxt.js (basado en Vue.js y Node.js). Encontrando coincidencia con esta investigación, con la utilización del framework Nuxt.js del lado del frontend el cual consume las APIs expuestas por loopBack4 según la definición de la arquitectura del sistema, arquitectura cliente-servidor.

4.3. Resultado del objetivo específico 3

Una vez realizado el análisis de los requerimientos y el diseño, comienza la fase de desarrollo, siendo el objetivo principal de esta investigación. Para la construcción del sistema web por el lado del backend se utilizó MongoDB como base de datos y el framework LoopBack en su versión 4 para la construcción de las APIs, el cual proporciona el acceso a la base de datos a través de las peticiones HTTP. Para la construcción de las APIs LoopBack básicamente necesita de: Modelos, Fuentes de datos, Repositorios y Controladores. En la creación de una fuente de datos, se realiza la configuración de la conexión de la base de

datos creada desde MongoDB Atlas. Seguidamente agregamos los modelos en base al modelo de la base de datos diseñado para el proyecto. Para crear un repositorio es necesario tener creado ya un modelo y la configuración de la base de datos, ya que la función de un repositorio es crear una relación entre el modelo y la fuente de datos; recordar que LoopBack da la posibilidad de tener más de una fuente de datos que en este proyecto no es necesario, basta con tener una sola fuente de datos. Y por último tenemos la creación de los controladores que son creados a partir de un repositorio que te permiten realizar las operaciones CRUD sin ningún esfuerzo. Para probar nuestras APIs utilizamos el explorador de API de LoopBack, que brinda la capacidad de ver y probar, también se utilizó la herramienta de Postman. <https://tesisappcontrol.mybluemix.net/>.

Por el lado del frontend se utilizó el framework Nuxt.js, el cual, nos permitió trabajar de forma limpia y ordenada, de manera que cualquier otro desarrollador podrá entender e incorporar los módulos necesarios, además puede incluir las librerías si son necesarias. Nuxt.js hace uso de la APIs para llevar a cabo todas las funcionalidades y vuelax para el diseño de las interfaces. Como resultado se obtuvo el sistema web bajo la metodología de Programación Extrema (XP), ver Anexo C. Se agregó un mixin que básicamente es todo de la app, conexiones de api, CRUD completo escrito una sola vez para todas páginas, paginación y búsqueda incluida, nos permitió la reutilización del código, de manera que no se duplique innecesariamente, lo que hace posible la distribución de funcionalidades reutilizables.

4.3.1. Discusión

Se utilizó la metodología XP para el desarrollo del sistema web, encontrándose coincidencia con De Liz (2020) utilizó el framework Nuxt.js la el desarrollo del lado de frontend, la implantación del modelo de datos con Vuex y la interfaz de usuario con Vue y Vuetify. Por otro lado, Tacilla (2018) desarrolló un sistema informático web de gestión de incidencias usando el framework AngularJS y NodeJS, llegando a la conclusión de que el proyecto es factible y mejor el proceso de gestión de incidencia, reduciendo el tiempo en dar solución a las incidencias reportadas, y logrando aumentar el porcentaje de atención aumentando el nivel de satisfacción del cliente.

Otro de los aspectos obtenidos, coincide con Falcón (2017), quien desarrolló un sistema web que le permitió reducir el tiempo de trabajar y generar valor para la empresa. Coincidiendo en usar herramientas de software libre, como JavaScript, que facilita el desarrollo por que abarata

los costos de los recursos de software necesario para el desarrollo del sistema web y además se puede encontrar abundante documentación.

En esta investigación se está de acuerdo con esta aseveración con el párrafo anterior porque el desarrollo del sistema web tuvo como resultado esta característica, los datos que se almacenaban en archivos de Excel ahora se centraliza en una base de datos, MongoDB, consistente que contiene la información centralizada y actualizada y de esta manera se aborda la problemática de esta investigación y se demuestra la necesidad de contar con sistema web.

4.4. Resultado del objetivo específico 4

4.4.1. Análisis de tiempo de atención

En esta sección se detalla el análisis del tiempo de atención con la situación actual y con el uso del sistema web de acuerdo a la experiencia de los participantes encargados. Se muestra a manera detallada en Anexo G.

4.4.1.1. Atención Actual.

Tabla 31. Atención actual en la entrega de equipo - Elaboración propia.

Atención en la Entrega de Equipo Actividades	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Acercarse a DTI (mesa de ayuda) para recojo del equipo	0,0		X
Revisar correo de orden de asignación de equipo	1,0	X	
Buscar y confirmar los datos del personal junto a su cargo (Archivo Excel)	1,3	X	
Buscar el registro del equipo (Archivo Excel)	1,8	X	
Buscar el equipo en el almacén	3,8	X	
Buscar el registro del estado del equipo (Revisar si es necesario las fichas anteriores de devolución)	3,7	X	
Imprimir 2 copias de la ficha de entrega	1,0	X	
Firmar y colocar la huella digital en las 2 fichas de entrega	1,2		X
Recepcionar el equipo asignado	1,3		X
Verificar los datos consignados en la ficha de entrega y colocar sello de conformidad	1,0	X	
Tiempo Total:	16,2		

Tabla 32. *Atención actual de devolución de equipo – Elaboración propia.*

Atención en la devolución de Equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Actividades			
Acercarse a DTI (mesa de ayuda) para realizar la devolución de equipo	0,0		X
Revisar correo de orden de devolución de equipo	1,2	X	
Buscar y confirmar los datos del personal junto a su cargo (Archivo Excel)	1,2	X	
Buscar el registro del equipo (Archivo Excel)	1,5	X	
Recepcionar y verificar el estado del equipo a devolver (Revisar si es necesario las fichas anteriores de devolución)	4,5	X	
Aceptar la conformidad del estado del equipo	1,3	X	
Imprimir el formato de devolución del equipo	1,0	X	
Llenar la ficha de devolución con los datos solicitados, firma y huella digital	3,2		X
Verificar los datos consignados en la ficha de devolución y colocar sello de conformidad	3,2	X	
Tiempo Total:	17,0		

4.4.1.2. Atención con el sistema web.**Tabla 33.** *Atención con el sistema web en la entrega de equipos-Elaboración propia.*

Atención en la entrega de equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Actividades			
Acercarse a DTI (mesa de ayuda) para recojo de equipo	0,0		X
Revisar correo de orden de asignación de equipo	1,0	X	
Buscar y confirmar los datos del personal y su cargo (Sistema)	1,0	X	
Buscar el registro del equipo en el Sistema	1,2	X	
Buscar el equipo en el almacén	3,2	X	
Imprimir 2 copias de la ficha de entrega	1,0	X	
Firmar y colocar la huella digital en las 2 fichas de entrega	1,2		X
Recepcionar el equipo asignado	1,5		X
Verificar los datos consignados en la ficha de entrega y colocar sello de conformidad	1,2	X	
Tiempo Total:	11,2		

Tabla 34. *Atención con el sistema web - Elaboración propia.*

Atención en la devolución de Equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Actividades			
Acercarse a DTI (mesa de ayuda) para entrega del equipo móvil	0,0		X
Revisar correo de orden de devolución de equipo	1,0	X	
Buscar y confirmar los datos del personal y su cargo (Sistema)	1,2	X	
Verificar los datos del equipo asignado en el sistema	2,3	X	
Recepcionar y verificar el estado del equipo (En caso se tenga una observación colocar una nota)	3,8	X	
Aceptar la conformidad del estado del equipo	1,0	X	
Confirmar la devolución del equipo en el sistema	0,5	X	
Tiempo Total:	9,8		

Tabla 35. *Resultados obtenidos con el sistema web - Elaboración propia.*

Actividades	Atención Actual (minutos)	Atención con el sistema web (minutos)	Diferencia (minutos)	Porcentaje
Entrega de equipos móviles	16.2	11.2	5	30,72 %
Devolución de equipos móviles	17	9.8	7.2	42.35%

4.4.2. Análisis de satisfacción de uso del sistema web

Se detalla la percepción de los usuarios finales (Coordinador de Servicios TI, Operador de Soporte Técnico y Mesa de Ayuda) respecto al uso del sistema web, se realizó mediante una encuesta, anexo F.

Tabla 36. *Pregunta N° 1 - La interfaz es amigable/intuitiva - Elaboración propia.*

Calificación	Cantidad
Nada de acuerdo	0
Poco de acuerdo	0
De acuerdo	0
Muy de acuerdo	4

La interfaz es amigable/intuitiva.

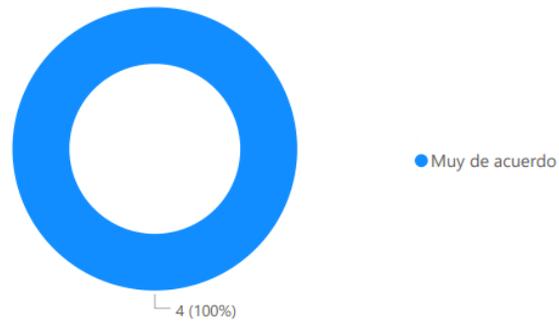


Gráfico 1. Análisis de satisfacción, Pregunta 1 - Elaboración propia.

INTERPRETACIÓN: Del total usuarios finales encuestados, el 100% califican como muy de acuerdo a un interfaz amigable/intuitiva del sistema web.

Tabla 37. Pregunta N° 2 - El tiempo de respuesta es óptimo - Elaboración propia.

Calificación	Cantidad
Nada de acuerdo	0
Poco de acuerdo	0
De acuerdo	2
Muy de acuerdo	2

El tiempo de respuesta es optimo

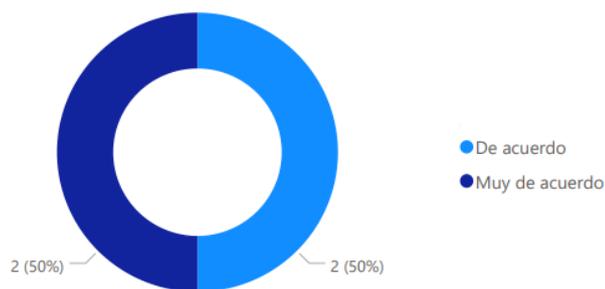


Gráfico 2. Análisis de satisfacción, Pregunta 2 - Elaboración propia.

INTERPRETACIÓN: Del total usuarios finales encuestados, el 50% califican como acuerdo y el 50% restante como muy de acuerdo al tiempo de respuesta del sistema web.

Tabla 38. *Pregunta N° 3 - Me brinda información actualizada - Elaboración propia.*

Calificación	Cantidad
Nada de acuerdo	0
Poco de acuerdo	0
De acuerdo	0
Muy de acuerdo	4

Me brinda información actualizada en cualquier momento que requiera.

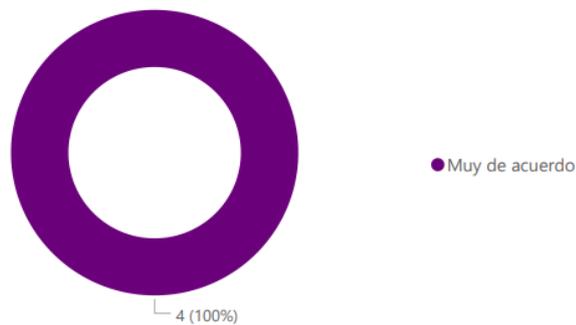


Gráfico 3. Análisis de satisfacción, Pregunta 3 - Elaboración propia.

INTERPRETACIÓN: Del total usuarios finales encuestados, el 100% califican como muy de acuerdo respecto a la información actualizada que brinda el sistema web.

Tabla 39. *Pregunta N° 4 - Me ayuda a reducir el material físico - Elaboración propia.*

Calificación	Cantidad
Nada de acuerdo	0
Poco de acuerdo	0
De acuerdo	1
Muy de acuerdo	3

Me ayuda a reducir el material físico.

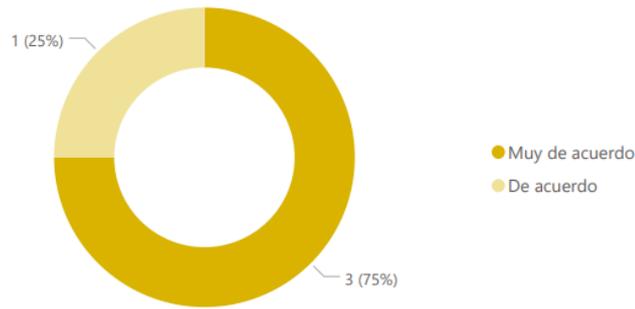


Gráfico 4. Análisis de satisfacción, Pregunta 4 - Elaboración propia.

INTERPRETACIÓN: Del total usuarios finales encuestados, el 75% califican como muy acuerdo y el 25% restante como de acuerdo con respecto a la reducción de material físico al utilizar del sistema web.

Tabla 40. Pregunta N° 5 - Me ayuda a gestionar con facilidad - Elaboración propia.

Calificación	Cantidad
Nada de acuerdo	0
Poco de acuerdo	0
De acuerdo	1
Muy de acuerdo	3

Me ayuda a gestionar los celulares corporativos con facilidad

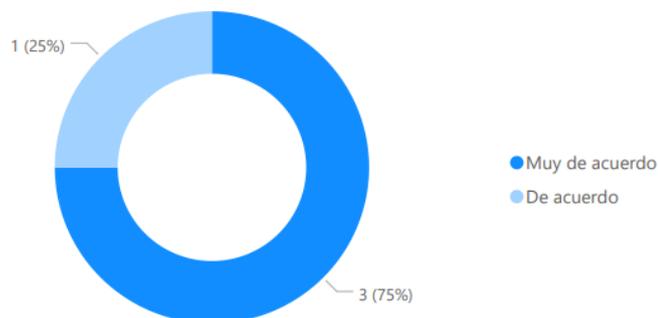


Gráfico 5. Análisis de satisfacción, Pregunta 5 - Elaboración propia.

INTERPRETACIÓN: Del total usuarios finales encuestados, el 75% califican como muy acuerdo y el 25% restante como de acuerdo a la facilidad de gestionar los celulares corporativos con la ayuda del sistema web.

4.4.3. Discusión

En el análisis del tiempo de atención se representaron en tablas del proceso actual y con el uso del sistema web, en la cual participaron de la evaluación los encargados directos que poseen experiencia en la atención de la gestión de equipos móviles corporativos de la universidad, los cuales registraron el tiempo de atención que les llevó en la entrega y devolución de los equipos, teniendo como resultado lo siguiente: en la parte de la entrega de un equipo móvil se tiene en el proceso actual según la tabla 33 una demora de tiempo promedio de 16.2 minutos y el proceso con el uso del sistema web según la tabla 35 se tiene un tiempo promedio de atención de 11.2, lo cual se tiene una mejora de tiempo de 5 minutos en la entrega de un equipo móvil y en la devolución de equipos con el proceso actual según la tabla 34, se tiene un tiempo promedio de 17 minutos y con el proceso del sistema web según la tabla 36 se tendría un tiempo promedio de 9,8 minutos en la atención, evidenciando de esta manera una mejora en la atención al usuario de 7,2 minutos, mostrándonos de esta manera una notable mejora en la atención al usuario en ambos procesos. También en la parte del análisis de satisfacción del cliente se pudo hacer una encuesta a los encargados directos en la atención al usuario, ya que estos tuvieron interacción con el sistema propuesto y de acuerdo a su percepción respondieron a la encuesta teniendo como resultado, cifras favorables en el uso y en la eficiencia del proceso. El sistema ayuda a reducir el tiempo de trabajo empleado coincidiendo con la investigación de Falcón (2017), quien desarrollo un sistema web que le permitió reducir el tiempo de trabajo y generar valor para la empresa, el sistema ayudo a reducir en gran manera el tiempo, y en la investigación realizada de Cenas (2018) ayudo a reducir un 54.5% del tiempo empleado en el trabajo sobre todo en los reportes y consultas de la información, permitiendo un control más ordenado y eficaz.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se concluye con desarrollo del sistema web para la gestión y control de celulares corporativos, validado con las pruebas de aceptación y encuesta por parte de los usuarios finales (Coordinador de Servicios TI, Operador de Soporte Técnico y Mesa de Ayuda) las cuales tuvieron resultados satisfactorios.

Se aplicó la metodología XP el cual permitió realizar todos los entregables del sistema. XP es una buena alternativa para desarrollar sistemas web, es una metodología flexible, ya que brinda la posibilidad de definir o actualizar los requerimientos a medida que se avanza en el proyecto en base a las historias de usuario y las pruebas dentro de cada iteración. Para el cumplimiento del primer objetivo, se logró recopilar la información que se utiliza para la gestión y control de celulares, una vez realizado se logró realizar el análisis definiendo los requerimientos funcionales y no funcionales, y estas fueron plasmadas en historias de usuario con los criterios de aceptación, este punto, para ver el cumplimiento de cada historia.

Con respecto al segundo objetivo se logró realizar el diseño del prototipo, de la arquitectura y el modelado de la base de datos, en base a los requerimientos y las historias de usuario para poder tener una mejor comprensión de lo se requiere en la fase de desarrollo.

En cuanto al tercer objetivo específico se logró desarrollar el sistema web, utilizando herramientas de software libre permite que cuente con una herramienta a la medida que automatiza la gestión y control de celulares corporativos, así tener una data actualizada cuando se requiera. Por el lado del backend, se desarrolló las APIs con LoopBack, el cual nos facilitó la creación de APIs mediante la línea de comandos, esto hace que crear modelos, repositorio, controladores y conectar a la base de datos, MongoDB, sea increíblemente sencillo. Y por el lado de frontend, con Nuxt.js, brindando un desarrollo más rápido con enrutador genérico automático, funciones de uso compartido, división automática de código con páginas renderizadas previamente, lo más importante el uso de Mixins, el cual, nos facilitó en gran manera la reutilización de funcionalidades, conexiones de API, CRUD completo escrito una sola vez para ya no estar escribiendo la misma funcionalidad para cada módulo.

Con respecto al cuarto objetivo se pudo mostrar la mejora que se obtiene con el sistema web de gestión de equipos móviles corporativos, puesto que según la tabla 37, nos muestran de la siguiente manera: en la atención de la entrega de equipos hay una mejora del 30,72% que equivale a una reducción de atención de 5 minutos y en la atención en la devolución de equipos hay una mejora del 42,35% lo cual equivale una disminución de atención de 9,8 minutos, evidenciando de esta manera el aumento de la eficiencia en la atención al usuario. Asimismo, se realizó una encuesta de satisfacción con el sistema web propuesto realizado a los que harán uso del sistema en este caso al personal encargado de la atención al usuario y los resultados se pueden apreciar en las tablas 38, 39, 40, 41 y 42, lo cual se tiene resultados favorables en la aceptación del sistema, determinando de esta manera que el sistema mejora en el tiempo de atención al usuario con un sistema que tiene la aceptación de los operadores.

5.2. Recomendaciones

Utilizar la metodología de programación extrema (XP) es una buena alternativa para desarrollar sistemas web, ofrece un desarrollo más ágil, brinda la posibilidad de definir o actualizar los requerimientos a medida que se avanza en el proyecto.

Validar constantemente con la ayuda del cliente cada requerimiento y funcionalidad, para evitar crear funcionalidades no solicitadas y evitar la pérdida de tiempo.

Utilizar full stack JavaScript, es una de las mejores opciones para desarrollar una aplicación. No hubo necesidad de cambiar entre diferentes lenguajes de programación. LoopBack, le permite: crear API REST con poca o ninguna codificación, acceder a datos SQL, NOSQL, y otras API REST, incorporar relaciones de modelos y controles de acceso para API complejas. Su CLI proporciona una amplia gama de comandos que puede hacer, desde crear modelos, repositorios y controladores, hasta configurar una fuente de datos para la aplicación. Por último, es sencillo empezar a utilizar LoopBack porque la curva de aprendizaje es baja.

Nuxt es una excelente opción que ofrece todo lo necesario para desarrollar del lado de frontend, cuenta con diferentes módulos y paquetes que ayudan a que el desarrollo sea mucho más sencillo. En última instancia, la elección dependerá de lo que intente construir. Cada opción tiene sus ventajas. En general, elija Nuxt si necesita SSR / SPA o si el SEO es importante para su desarrollo.

REFERENCIAS

- Aiquipa, A. A. (2018). *Sistema web para el control de la gestión de mantenimiento de equipos informáticos de la compañía Naviera Natalia SAC*. Universidad César Vallejo, Lima.
- Amao, J. (2016). *Sistema de inventario vía web para mejorar el control de los equipos informáticos en la empresa J&C Soluciones S.A.C*. Universidad César Vallejo, Trujillo.
- Bahit, E. (2012). *Scrum y eXtreme Programming para Programadores*.
- Barrios Arce, J. I. (24 de diciembre de 2019). Sistemas de Información. El Business Intelligence. *BIG DATA*. Obtenido de <https://www.juanbarrios.com/sistemas-de-informacion-el-business-intelligence/>
- Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice*.
- Beck, K. (2000). *Praise for Extreme Programming Explained, Second Edition*.
- Bernal, C. (2010). *Metodología de la Investigación*. Colombia: Pearson Educación de Colombia Ltda.
- Borja, Y. (s.f.). *Metodología Ágil de Desarrollo de Software-XP*.
- Brett, N. (2018). *Getting to Know Vue.js*. Apress. Obtenido de <https://www.oreilly.com/library/view/getting-to-know/9781484237816/>
- Brown, E. (2019). *Web development with Node and Express*. O'Reilly Media, Inc. Obtenido de <https://www.oreilly.com/library/view/web-development-with/9781492053507/>
- Bustamante, D., & Rodríguez, J. (Marzo de 2014). Obtenido de La web del programador: http://95.120.179.179/files/1502127241_Metodologia-XP.pdf
- Cantone, D. (2006). *Ciclo de vida del software*. Creative Andina Corp.
- Cenas, J. (2018). *Sistema informático web para el control del proceso de alquiler de GPS en la empresa HENPHONE S.R.L. – Chimbote, 2018*. Universidad San Pedro Facultad de Ingeniería, Chimbote.

- Chacón, D., Ruiz, J., & Mendoza, K. (2014). *Ciclo de Vida del Software*. Universidad Estatal a Distancia.
- Chiarelli, A. (2018). *Beginning React*. Obtenido de https://subscription.packtpub.com/book/application_development/9781789530520
- Copes, F. (2021). *Vue Router*. Obtenido de <https://flaviocopes.com/vue-router/>
- Crespo, J., Peña, E., Pascual, V., & Fustiel, Y. (2016). *Elección entre una metodología ágil y tradicional basado en técnicas de soft computing*.
- Cronbach, L. (1951). *Coefficient alpha and the internal structure of tests*. Psychometrika.
- De Liz, J. (2020). *Aplicación web progresiva para una red social de aficionados a los videojuegos*. Universidad Autónoma de Madrid, Madrid. Obtenido de https://repositorio.uam.es/bitstream/handle/10486/692833/Lis_Aguirregomezcorta_juande_tfg.pdf?sequence=1
- Facebook Inc. (2020). *React*. Obtenido de <https://reactjs.org/>
- Falcón, H. (2017). *Desarrollo de un aplicativo Web para la Gestión de Equipos Informáticos en el Área de Soporte Técnico de Informática - INEI Huánuco - 2017*. Universidad Nacional Hermilio Valdizán Facultad de Ingeniería Industrial y de Sistemas, Huánuco.
- Fielding, R. (2000). *Network-based Architectural Styles*.
- Franklin, J., Wanyoike, M., & Bouchefra, A. (2019). *Working with Vue.js*. Obtenido de <https://www.oreilly.com/library/view/working-with-vuejs/9781492071426/>
- Frijlink, D. (03 de Diciembre de 2020). *Nuxt.js Single Page Application Boilerplate*. Obtenido de DEV: <https://dev.to/dennisfrijlink/nuxt-js-single-page-application-boilerplate-3215>
- Frost, B. (2018). *Designing Systems*. Obtenido de <https://blog.prototypr.io/atomic-design-dise%C3%B1o-at%C3%B3mico-113411bc7816>
- Garlan, D., & Shaw, M. (1996). *An Introduction to Software Architecture*. World Scientific Publishing Co.

- Giamas, A. (2017). *Mastering MongoDB 3.x*. Obtenido de http://proquest.safaribooksonline.com/book/databases/9781783982608/fault-tolerance-and-high-availability/cccf65b3_10b4_4619_a22d_980ff51a917f_xhtml#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODE3ODM5ODI2MDglMkZkMGUzNzRjOV84NTQ2XzQ1MWZfYjM1Y18xOTkyN2Y0N2Q1ZWZfeGh
- Hay, S. (2018). *Diseño Atómico*. Obtenido de <https://blog.prototypr.io/atomic-design-dise%C3%B1o-at%C3%B3mico-113411bc7816>
- Holmes, S., & Harber, C. (2019). *Getting MEAN with Mongo, Express, Angular, and Node*. Manning Publications. Obtenido de <https://www.oreilly.com/library/view/getting-mean-with/9781617294754/>
- Holque, S. (2018). *Full-Stack React Projects Modern web development using React 16, Node, Express, and MongoDB*. Obtenido de <https://learning.oreilly.com/library/view/full-stack-react-projects/9781788835534/1c9ea6b9-f700-4199-9c0b-040d1ce5dd18.xhtml>
- Jadhav, M., Balkrishna, S., & Anushree, D. (2015). *Single Page Application using AngularJS*.
- Jiménez, E. (2018). *¿Qué son las bases de datos?* Obtenido de <https://medium.com/@emrajipower/qu%C3%A9-es-una-base-de-datos-58683e6d4304>
- Keeling, M. (2017). *Design It!*
- Kumar, D. (2018). Best Practices For Building RESTFUL Web Services. *Infosys*, 12.
- Leau, Y., Loo, W., & Tham, W. (2012). *Software Development Life Cycle AGILE vs Traditional Approaches*.
- Lima, L., Petrucelli, E., & Espíritu Santo, F. (2019). OVERVIEW ABOUT STATE MANAGEMENT IN VUE.JS WITH THE VUEX LIBRARY. 55-56.
- Maida, E., & Pacienza, J. (2015). *Metodologías de desarrollo de software*. Universidad Católica Argentina.

- Márquez, B., & Zulaica, J. (2004). *Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español*. Universidad de las Américas Puebla, México.
- Mead, A. (2018). *Learning Node.js Development*. Packt. Obtenido de <https://www.packtpub.com/product/learning-node-js-development/9781788395540>
- Microsoft. (2020). Obtenido de Conceptos básicos del sistema de gestión de inventarios: <https://dynamics.microsoft.com/es-es/field-service/inventory-management-system/>
- Mora, P. (2015). *UF1879 - Equipos de interconexión y servicios de red*. Obtenido de <https://books.google.com.pe/books?id=YbhWDwAAQBAJ&pg=PA156&lpg=PA156&dq=HTTP+se+basa+en+sencillas+operaciones+de+solicitud/respuesta.+Un+cliente+establece+una+conexi%C3%B3n+con+un+servidor+y+env%C3%ADa+un+mensaje+con+los+datos+de+la+solicitud.+El+servidor>
- Mundo API. (5 de Agosto de 2019). *API WORLD*. Obtenido de <https://apiworld.co/2019-api-award-winners-announced/>
- Navarro, A., Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *redalyc.org*, 11.
- Node.js. (2020). Obtenido de <https://nodejs.org/es/>
- Nuxt.js. (2020). Obtenido de <https://nuxtjs.org/>
- Pressman, R. (2010). *Ingeniería del Software. Un Enfoque Practico*.
- Prusty, N. (2016). *Modern JavaScript Applications*. Obtenido de https://subscription.packtpub.com/book/web_development/9781785881442/5/ch051v11sec27/introducing-websocket
- Schmitz, D., & Georgii, D. (2016). *Vuex Concepts: The Flux Application Architecture for Vue.js*.
- Scott, A. (2020). *JavaScript Everywhere*. O'Reilly Media, Inc. Obtenido de <https://www.oreilly.com/library/view/javascript-everywhere/9781492046974/>

- Sharma, A. (2018). *Full-Stack Web Development with Vue.js and Node*. Pack. Obtenido de <https://www.packtpub.com/product/full-stack-web-development-with-vue-js-and-node/9781788831147>
- Shipton, L. (20 de Abril de 2021). *Last Call*. Obtenido de The RapidAPI Blog: <https://rapidapi.com/blog/best-nodejs-frameworks/>
- Shute, Z. (2019). *Advanced JavaScript : speed up web development with the powerful features and benefits of JavaScript*. Packt Publishing. Obtenido de <https://www.oreilly.com/library/view/advanced-javascript/9781789800104/>
- Singh, H., & Tanna, M. (2018). *Serverless Web Applications with React and Firebase*. Obtenido de https://subscription.packtpub.com/book/web_development/9781788477413/1/ch011v11sec10/react
- State of JS. (2020). *State of JS*. Obtenido de State of JS: <https://2020.stateofjs.com/en-US/technologies/front-end-frameworks/>
- Street, M., Passaglia, A., & Halliday, P. (2018). *Vue.js 2 Web Development*. Pack Publishing Ltd.
- StrongLoop. (2020). Obtenido de LoopBack 4: <https://loopback.io/>
- StrongLoop. (2020). *Express*. Obtenido de <https://expressjs.com/>
- Subramanian, H., & Raj, P. (2019). *Hands-on RESTful web API design patterns and best practices : design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs*. Obtenido de <https://www.oreilly.com/library/view/hands-on-restful-api/9781788992664/>
- Tacilla, J. (2018). *Sistema Informático Web de Gestión de Incidencias Usando el Framework AngularJS y NodeJS para la Empresa Redteam Software LLC*. Universidad Privada Antenor Orrego, Trujillo.
- Tam, J., Vera, G., & Oliveros, R. (2008). Tipos, Métodos y Estrategias de Investigación Científica. 145-154.

Tinoco, O., Rosales, P., & Salas, J. (2010). *Criterios de selección de metodologías de desarrollo de software*.

Vuesax. (2020). *Vuesax*. Obtenido de <https://vuesax.com/>

Vuetify. (2020). *Vuetify*. Obtenido de <https://vuetifyjs.com/en/>

Vuex. (2020). *Vuex*. Obtenido de <https://vuex.vuejs.org/#what-is-a-state-management-pattern>

Whiteley, D. (8 de Agosto de 2019). *Congratulations to LoopBack, Winner of 2019 API Award for API Middleware*. Obtenido de <https://strongloop.com/strongblog/loopback-2019-api-award-api-middleware/>

Ylquimiche, S. (2018). *Propuesta de un sistema de control de inventarios de los equipos de cómputo de los laboratorios de la Facultad de Ciencias Empresariales de la Universidad Peruana Unión, 2018*. Universidad Peruana Unión, Lima.

You, E. (2020). *Vue.js*. Obtenido de <https://vuejs.org/v2/guide/>

ANEXOS

Anexo A. Fase de Planificación: Historia de usuario y criterios de aceptación.

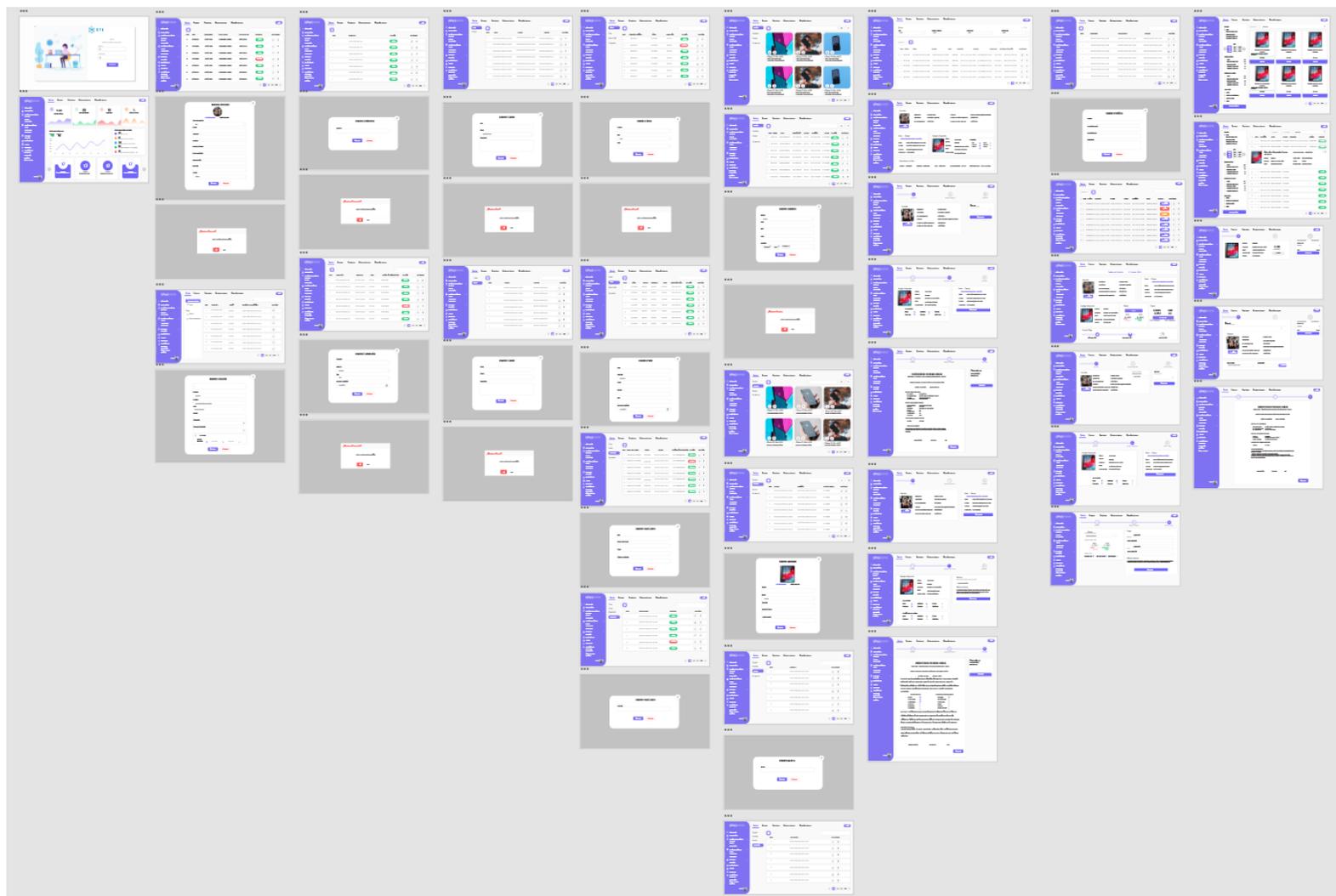
		ENUNCIADO DE LA HISTORIA				CRITERIOS DE ACEPTACION	
ID	ROL	CARACTERISTICA / FUNCIONALIDAD	OBJETIVO	PRIORIDAD	COMPLEJIDAD	ESCENARIOS	COMPORTAMIENTO ESPERADO
USHIS001	DIGETI (Operador, Atención al cliente)	Gestionar personas	Tener información del personal	Alta	Media	1	Una persona vinculada a cualquier actividad del sistema no puede ser eliminada.
						2	Verificar que no haya duplicidad de una misma persona mediante el número de Documento de Identidad.
USHIS002	DIGETI (Operador, Atención al cliente)	Gestionar roles y permisos	Controlar el acceso de los usuarios al sistema.	Media	Alta	1	Dar acceso al sistema a un usuario determinado asignado un rol.
						2	Permitir el ingreso al sistema según el rol brindado.
USHIS003	DIGETI (Operador, Atención al cliente)	Asociar un plan a una línea	Asignarle a una determinada equipo o persona.	Alta	Alta	1	Un plan debe pertenecer a un operador.
						2	Una línea debe pertenecer a un plan y a un simcard.
						3	NO puede eliminarse un registro si se tiene al menos una asignación de equipo.
USHIS004	DIGETI (Operador, Atención al cliente)	Gestionar equipos	Asignar a una línea o persona.	Alta	Alta	1	Un equipo debe pertenecer a un modelo.
						2	Un modelo debe estar asociado con una marca.
						3	Un equipo puede tener más de un accesorio.
						4	Un modelo debe estar sujeto a políticas en caso de pérdida, robo o cualquier daño ocasionado sobre un equipo en una determinada campaña.
						5	NO puede eliminarse un registro si se tiene al menos una asignación.
USHIS005	DIGETI (Operador, Atención al cliente)	Asignación de línea y/o equipo a un cargo	Realizar la entrega	Alta	Muy alta	1	Permitir realizar la búsqueda mediante el número de celular al seleccionar una línea.
						2	Permitir realizar la búsqueda con IMEI, modelo y marca al seleccionar un equipo.
						3	Permitir realizar la búsqueda de un cargo mediante el nombre o nombre de área.

						4	Se puede hacer la asignación de solo equipo, línea o ambas.
						5	En un registro solo se puede EDITAR solo si esta no ha sido entregada.
						6	Una línea, un equipo y un cargo solo se pueden pertenecer a una asignación activa.
						7	Una vez seleccionado el equipo pueda mostrar la información de los accesorios
						8	NO puede eliminarse un registro si se tiene al menos una asignación.
USHIS006	DIGETI (Operador, Atención al cliente)	Entrega de línea y/ o equipos	Brindarle el equipo o línea asignada según el cargo a la persona	Muy alta	Muy alta	1	Al seleccionar la persona realizar la búsqueda mediante el Numero de Documento de Identidad.
						2	Una vez seleccionado la persona pueda permitirme editar los datos de la persona sin tener que salir de la pantalla.
						3	En la lista de registro se pueda realizar un filtro mediante área, equipo asignado (Modelo), Entregado, sin entregar.
						4	Generar un Formato de entrega de línea y/o equipo, según el formato utilizado.
						5	Imprimir el documento de la entrega de línea y/o equipo
						6	NO puede eliminarse un registro si está asignado a una persona.
						7	Permitir exportar en Excel el registro.
USHIS007	DIGETI (Operador, Atención al cliente)	Gestionar las devoluciones	Tener una lista actualizada para saber la cantidad de línea y/o equipos disponibles.	Alta	Alta	1	Una devolución debe tener un motivo.
						2	Una devolución debe tener al menos una condición de equipo.
						3	Una devolución se realiza a partir de una entrega de línea y/o equipo.
						4	En caso de un equipo, verificar (Poder editar) los accesorios con los cuales se está realizando la devolución.
						5	Generar un formato de devolución.
						6	Imprimir el documento de devolución.
						7	NO puede eliminarse un registro.
USHIS008	DIGETI (Operador, Atención al cliente)	Gestionar perdidas y robos	Tener un registro de equipos perdidos o robados para	Alta	Alta	1	Una pérdida o robo se registra a partir de una entrega de línea y/o equipo.
						2	Al realizar un registro pueda realizarse la búsqueda mediante el número de documento de identidad de la persona entregada el equipo.
						3	El sistema pueda calcular los montos a pagar de cada usuario según la política.

			poder tener una cantidad actualizado de equipos disponibles.				
USHIS009	DIGETI (Operador, atención al cliente) Administrativos (Oficina de Gerencia y Tesorería)	Mostrar estadísticas generales	Tener un dato actualizado sobre las actividades y/o recursos del sistema.	Alta	Media	1	Mostrar la cantidad total de líneas, perdida/robo, solo chip (líneas), líneas libres.
						2	Cantidad de usuarios por mes
						3	Cantidad de líneas por área
						4	Cantidad de equipos por modelo.
						5	Al realizar un clic sobre los datos pueda mostrarme un detalle del registro.
USHIS010	DIGETI (Operador, atención al cliente) Administrativos (Oficina de Gerencia y Tesorería)	Visualización de reportes	Verificar los datos y tomar decisiones.	Baja	Alta	1	Visualizar las líneas y/o equipos asignados con detalle.
						2	Visualizar las líneas y/o equipos devueltos con detalle (fecha).
						3	Visualizar la cantidad de líneas o equipos reportados por pérdida o robo (según las políticas de uso) (Resumen por fecha)
						4	Visualizar un reporte de la cantidad de equipos por cada modelo
						5	Permitir descargar en formato de Excel y PDF, además poder visualizarse en línea
USHIS011	DIGETI (Operador, Atención al cliente)	Gestionar venta de equipos.	Realizar la venta de un equipo	Baja	Media	1	Asignar un precio por modelo.
						2	Generar un formato de venta.
						3	Imprimir el documento.

Para ver a más detalle en: <https://n9.cl/jp6up>

Anexo B. Diseño de prototipo del sistema web.



Para ver a más detalle en: <https://n9.cl/adobe-xd>

Anexo C. Código de Mixin – Nuxtjs

```
1 import { totalPages, totalSkip } from '../utils/paginate'
2
3 export default {
4   async fetch() {
5     if (this.query) {
6       const query = this.$route.query
7       this.page = query.page ? parseInt(query.page) : 1
8       this.search = query.search ?? ''
9       this.size = query.size ?? '7'
10    }
11    const loading = this.$vs.loading({
12      type: 'points',
13    })
14
15    this.filter.skip = totalSkip(this.page, this.filter.size)
16    const { results, count } = await this.getItems()
17    this.items = results
18    this.filter.total = totalPages(count, this.filter.size)
19
20    if (this.filter.total !== 0 && this.page > this.filter.total) {
21      this.page = this.page > 1 ? this.page - 1 : 1
22      if (!this.query) {
23        setTimeout(() => {
24          this.$fetch()
25        }, 10)
26      }
27    }
28    loading.close()
29  },
30  fetchOnServer: false,
31  data: () => ({
32    search: '',
33    items: [],
34    item: {},
35    loading: false,
36    dialog: false,
37    popup: false,
38    page: 1,
39    filter: {
40      size: 7,
41      total: 0,
```

```

42     skip: 0,
43   },
44   }),
45
46   watch: {
47     '$route.query': '$fetch',
48
49     search(v) {
50       if (this.query) {
51         if (v.length) {
52           this.$router.push({
53             query: { search: v, page: 1, size: this.size },
54           })
55         } else {
56           this.$router.push({ query: { page: 1 } })
57         }
58       } else {
59         this.page = 1
60         this.$fetch()
61       }
62     },
63     size(v) {
64       if (this.query) {
65         this.$router.push({
66           query: { page: this.page ?? 1, size: v, search: this.search },
67         })
68       } else {
69         this.$fetch()
70       }
71     },
72     page(v) {
73       if (this.query) {
74         this.filter.skip = totalSkip(this.page, this.filter.size)
75         this.$router.push({ query: { page: v } })
76       } else {
77         this.$fetch()
78       }
79     },
80   },
81

```

```

82   methods: {}
83   async getItems() {
84     const { size, skip } = this.filter
85     return await this.$axios
86       .get(
87         `${this.path}/?filter[limit]=${size}&filter[skip]=${skip}&search=${this.search}`
88       )
89     .then((x) => ({
90       ...x.data,
91     }))
92     .catch(() => ({}))
93   },
94   async saveItem() {
95     this.loading = true
96     if (this.item.id) {
97       await this.$axios
98         .put(`${this.path}/${this.item.id}`, this.item)
99         .then(() => {
100           this.clear()
101         })
102         .catch(() => {})
103     } else {
104       await this.$axios
105         .post(`${this.path}`, this.item)
106         .then(() => {
107           this.clear()
108         })
109         .catch(() => {})
110     }
111     this.loading = false
112   },
113 },
114
115   async deleteItem() {
116     this.loading = true
117     await this.$axios
118       .delete(`${this.path}/${this.item.id}`)
119       .then(() => {
120         this.clear()
121       })

```

```

125
126   clear() {
127     this.$fetch()
128     this.popup = false
129     this.item = {}
130
131     this.dialog = false
132   },
133 },
134 }
135

```

Backend (APIs): <https://tesisappcontrol.myblucemix.net/>

Sistema web: <https://upbeat-fermi-056ba2.netlify.app/login/>

Anexo D. Archivo Excel, formatos utilizados para la gestión de celulares corporativos.

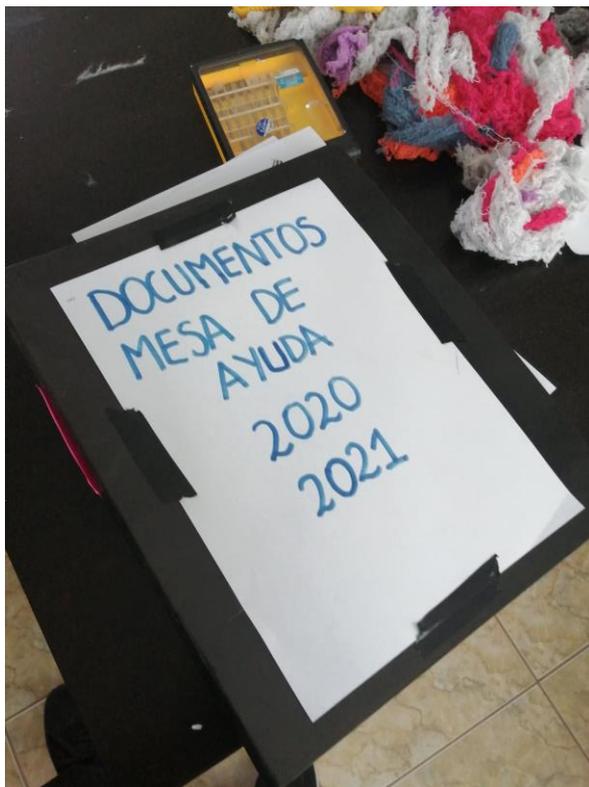
Lineas 2021

Archivo Editar Ver Insertar Formato Datos Herramientas Extensiones Ayuda Última modificación hace 6 horas por DTI Juliaca

100% Solo lectura

Nº	LINEA	IMEI EQUIPO	MAC	SERIE (SIN)	MODELO	MARCA	MODELO (MARCA)	DADO	TABLE USE	UIIDIFONO	ARCASA	COSTO	OPERADOR
1	951751463	35729095716989		F2LXPES1KPH1	A2101	iPhone	iPhone XS Max 64GB	SI	SI	SI	NO	S/ 6.299,00	MOVISTAR
2	951751461	35728995827448		F2LXPMPGCKPH1	A2101	iPhone	iPhone XS Max 64GB	SI	SI	SI	NO	S/ 6.299,00	MOVISTAR
3	951752055	352827113422019		F17D6MSDN6Y2	A2215	iPhone	iPhone 11 Pro	SI	SI	SI	NO	S/ 6.299,00	MOVISTAR
4	951752057	353079103990864	B8.B2.F8.79.6D.11	F71YT121K0K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
5	951752058	353078103990376	B8.B2.F8.77.A1.24	F71YR11DKKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
6	951752197	353078103845888	B8.B2.F8.8A.23.7D	F71YR11WDKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
7	951751594	3530781039495786	B8.B2.F8.75.00.EA	F71YR11HKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
8	951752054	353078103896293		F71YR329YKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
9	951292518	353078104007124	B8.B2.F8.77.2D.9D	F71YR3N2KXK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
10	951769876	353079103871734		F71YR1L70K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
11	951769878	353078103969474		F71YT1BNKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
12	951752053	353078103979810		F71YR1Y1K0K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
13	951301938	353078103796511	B8.B2.F8.85.00.46	F71YT15RKK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
14	951752056	353078103792551	B8.B2.F8.76.1B.F9	C8QXN5STK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
15	951769890	353079103790678	B8.B2.F8.7C.FD.7D	F71YR2F60K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
16	951769889	353078103932454	B8.B2.F8.78.D5.A8	F71YR140K0K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
17	951292558	353078103933478	B8.B2.F8.88.49.47	F71YR34SKXK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
18	951751320	353079103944300	B8.B2.F8.82.09.56	F71YR22EKXK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
19	951752051	353078103833637	B8.B2.F8.77.9A.D2	F71YR10EHKXK1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
20	950302620	353079103889207	B8.B2.F8.83.46.20	F71YR1N1K0K1	A2105	iPhone	iPhone XR 64GB	SI	SI	SI	NO	S/ 4.699,00	MOVISTAR
21	951292549	358392102911069	8C.E5.C0.2C.93.97	R28M7139NVE	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
22	951752205	358392102930376	8C.E5.C0.2C.93.0D	R28M7139LRZ	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
23	951292650	358392102927778	8C.E5.C0.2C.91.05	R28M7139CWB	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
24	951751595	358392102929261	8C.E5.C0.2C.92.F2	R28M7139HDE	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
25	943868598	358392102927737	8C.E5.C0.2C.90.FD	R28M7139CRY	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
26	951301896	358392102927885	8C.E5.C0.2C.91.1B	R28M7139D7M	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
27	951301928	358392102927521	8C.E5.C0.2C.90.D3	R28M7139CAM	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
28	951301925	358392102927943	8C.E5.C0.2C.91.27	R28M7139DDB	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
29	951300004	357971102661221	88.29.9C.DF.90.0B	R58M85HYCKN	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
30	951751596	358392102927604	8C.E5.C0.2C.90.E3	R28M7139CCN	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
31	951769887	358392103548615	8C.E5.C0.38.64.2D	R28M8079ZTL	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
32	951751593	358392103550561	8C.E5.C0.38.65.B3	R28M807A5QX	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
33	951292529	358392103550256	8C.E5.C0.38.65.75	R28M807A4SB	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR
34	951292528	358392103549886	8C.F5.C0.38.65.7B	R28M807A3NF	SM-A305G	Samsung	Samsung Galaxy A30	SI	SI	SI	SI	S/ 1.099,00	MOVISTAR

DIRECTORIO INVENTARIO EQUIPO FORMATO DESCUENTOS 2021 DIRECTORIO ZTE PERSONA INVENTARIO ACTUAL Copia de FORMATO 1 Copia de FOI



 **ENTREGA EQUIPO CELULAR - UPeUJ** 

FECHA _____

YO: _____
IDENTIFICADO CON DNI: _____ ESTANDO TRABAJANDO EN EL AREA
_____ CON EL CARGO: _____
_____ Y HABIENDO RECIBIDO EN CALIDAD DE ASIGNACION EL EQUIPO
CELULAR MODELO: _____
CON CHIP RPM NRO: _____ Y CON LOS SIGUIENTES ACCESORIOS:
AUDIFONO () CARGADOR () MEMORIA EXTERNA _____ () CARCASA
ADICIONAL () CABLE DATOS ()
Y QUE POR MOTIVOS DE _____
ESTOY DEVOLVIENDO EL EQUIPO CELULAR EN LAS SIGUIENTES CONDICIONES:

CONDICIONES ACCESORIOS	CONDICIONES EQUIPO
BATERIA	RAYADO
CARGADOR	DESPINTADO
AUDIFONO	GOLPEADO
CABLE USB	ROTO
MEMORIA	RAJADO
CARCASA AD	BUEN ESTADO

Y HABIENDO FIRMADO MI COMPROMISO RESPECTO A LAS POLITICAS DE USO DE TELEFONOS CELULARES, ME HAGO RESPONSABLE SI ES NECESARIO ASUMIR EL COSTO POR DAÑOS Y AVERIAS DEL EQUIPO SI LAS PRESENTA AL MOMENTO DE HACER LA ENTREGA.

_____ DTI _____ USUARIO _____

OBSERVACIONES: _____

Anexo E. Formato encuestas.

Universidad Peruana Unión

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPEU - Campus Juliaca

Encuesta sobre la gestión y control de celulares corporativos antes del desarrollo del Sistema Web.

Agradezco de antemano su apoyo para el llenado de esta encuesta que se realiza con fines de investigación.

1. Tiempo de atención en la entrega de equipo (Atención actual)

Atención en la Entrega de Equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Actividades			
Acercarse a DTI (mesa de ayuda) para recojo del equipo			X
Revisar correo de orden de asignación de equipo		X	
Buscar y confirmar los datos del personal junto a su cargo (Archivo Excel)		X	
Buscar el registro del equipo (Archivo Excel)		X	
Buscar el equipo en el almacén		X	
Buscar el registro del estado del equipo (Revisar si es necesario las fichas anteriores de devolución)		X	
Imprimir 2 copias de la ficha de entrega		X	
Firmar y colocar la huella digital en las 2 fichas de entrega			X
Recepcionar el equipo asignado			X
Verificar los datos consignados en la ficha de entrega y colocar sello de conformidad		X	

2. Tiempo de atención en la devolución del equipo (Atención actual)

Atención en la devolución de Equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Actividades			
Acercarse a DTI (mesa de ayuda) para realizar la devolución de equipo			X
Revisar correo de orden de devolución de equipo		X	
Buscar y confirmar los datos del personal junto a su cargo (Archivo Excel)		X	
Buscar el registro del equipo (Archivo Excel)		X	
Recepcionar y verificar el estado del equipo a devolver (Revisar si es necesario las fichas anteriores de devolución)		X	
Aceptar la conformidad del estado del equipo		X	
Imprimir el formato de devolución del equipo		X	
Llenar la ficha de devolución con los datos solicitados, firma y huella digital			X
Verificar los datos consignados en la ficha de devolución y colocar sello de conformidad		X	

Universidad Peruana Unión

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca

Encuesta sobre la gestión y control de celulares corporativos antes del desarrollo del Sistema Web.

Agradezco de antemano su apoyo para el llenado de esta encuesta que se realiza con fines de investigación.

1. Tiempo de atención en la entrega de equipo (Sistema web)

Atención en la entrega de equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Acercarse a DTI (mesa de ayuda) para recojo de equipo			X
Revisar correo de orden de asignación de equipo		X	
Buscar y confirmar los datos del personal y su cargo (Sistema)		X	
Buscar el registro del equipo en el Sistema		X	
Buscar el equipo en el almacén		X	
Imprimir 2 copias de la ficha de entrega		X	
Firmar y colocar la huella digital en las 2 fichas de entrega			X
Recepcionar el equipo asignado			X
Verificar los datos consignados en la ficha de entrega y colocar sello de conformidad		X	

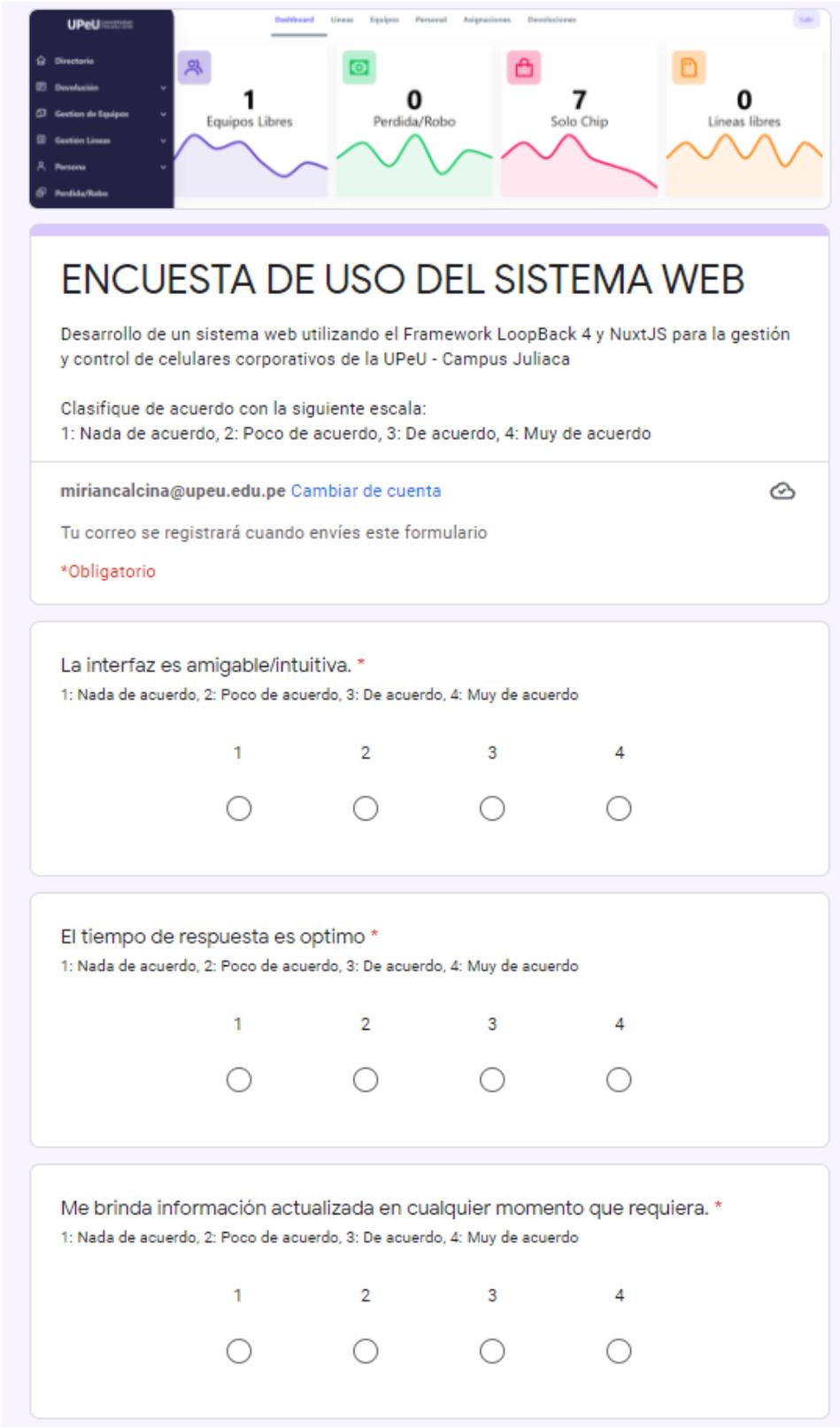
2. Tiempo de atención en la devolución del equipo (Sistema web)

Atención en la devolución de Equipo	Tiempo promedio (min.)	Participantes en el proceso	
		Mesa de Ayuda	Usuario
Acercarse a DTI (mesa de ayuda) para entrega del equipo móvil			X
Revisar correo de orden de devolución de equipo		X	
Buscar y confirmar los datos del personal y su cargo (Sistema)		X	
Verificar los datos del equipo asignado en el sistema		X	
Recepcionar y verificar el estado del equipo (En caso se tenga una observación colocar una nota)		X	
Aceptar la conformidad del estado del equipo		X	
Confirmar la devolución del equipo en el sistema		X	

3. Encuesta sobre de uso del Sistema Web: Clasifique de acuerdo a la siguiente escala: 4: Muy de acuerdo, 3: De acuerdo, 2: Poco de acuerdo, 1: Nada de acuerdo

Preguntas	1	2	3	4
La interfaz es amigable/intuitiva.				
El tiempo de respuesta es óptimo.				
Me brinda información actualizada en cualquier momento que requiera.				
Me ayuda a reducir el material físico.				
Me ayuda a gestionar los celulares corporativos con facilidad.				
La ayuda que presenta es útil.				

Anexo F. Encuesta de satisfacción del sistema web.



UPEU

Dashboard | Líneas | Equipos | Personal | Asignaciones | Devoluciones

1 Equipos Libres | 0 Perdida/Robo | 7 Solo Chip | 0 Lineas libres

ENCUESTA DE USO DEL SISTEMA WEB

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPEU - Campus Juliaca

Clasifique de acuerdo con la siguiente escala:
1: Nada de acuerdo, 2: Poco de acuerdo, 3: De acuerdo, 4: Muy de acuerdo

miriancalcina@upeu.edu.pe [Cambiar de cuenta](#)

Tu correo se registrará cuando envíes este formulario

***Obligatorio**

La interfaz es amigable/intuitiva. *
1: Nada de acuerdo, 2: Poco de acuerdo, 3: De acuerdo, 4: Muy de acuerdo

1 2 3 4

El tiempo de respuesta es optimo *
1: Nada de acuerdo, 2: Poco de acuerdo, 3: De acuerdo, 4: Muy de acuerdo

1 2 3 4

Me brinda información actualizada en cualquier momento que requiera. *
1: Nada de acuerdo, 2: Poco de acuerdo, 3: De acuerdo, 4: Muy de acuerdo

1 2 3 4

Anexo G. Prueba de confiabilidad y validación de los instrumentos utilizados.

Universidad Peruana Unión

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca

Encuesta sobre de uso del Sistema Web: Clasifique de acuerdo a la siguiente escala:
4: Muy de acuerdo, 3: De acuerdo, 2: Poco de acuerdo, 1: Nada de acuerdo

Análisis de Confiabilidad: Alfa de Cronbach

Sujeto	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Total
1	4	4	4	4	4	4	24
2	3	3	3	3	4	3	19
3	4	4	4	4	4	4	24
4	4	3	3	4	3	4	21
Varianzas	0,188	0,250	0,250	0,188	0,188	0,188	4,500

a) Mediante la varianza de los ítems (Cronbach, 1951)

$$\alpha = \frac{k}{k-1} \left[1 - \frac{\sum V_i}{V_t} \right]$$

α : Alfa de Cronbach
 k : Número de ítems
 V_i : Varianza de cada ítem
 V_t : Varianza del total

Cronbach, L.J. (1951).

k= 6
 V_i = 1,250
 V_t = 4,500
 α = 0,867

Análisis de la consistencia



Universidad Peruana Unión
Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPEU - Campus Juliaca

Analisis de Confiabilidad: R de Pearson

Encuesta sobre la gestión y control de celulares corporativos: Tiempo de atención en la entrega de equipo

PRE TEST										
Sujeto	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	0	1	2	2	3	4	1	1	2	1
2	0	1	2	1	3	4	1	1	2	1
3	0	1	1	2	2	3	1	2	1	1
4	0	1	2	2	3	4	1	1	1	1
5	0	1	1	2	3	4	1	2	1	1
6	0	1	1	2	3	3	1	1	2	1

POST TEST										
Sujeto	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	0	1	1	1	0	4	1	1	2	2
2	0	1	1	1	0	4	1	1	2	1
3	0	1	1	2	0	3	1	1	1	1
4	0	1	1	1	0	4	1	2	1	1
5	0	1	1	1	0	4	1	2	1	1
6	0	1	1	1	0	3	1	1	2	1

Coefficiente de correlación de Pearson: $r = 0,60$

$$r = \frac{cov(x,y)}{S_x S_y}$$

r : Coeficiente de correlación de Pearson
 $cov(x,y)$: Covarianza entre X e Y
 S_x : Desviación estándar de X
 S_y : Desviación estándar de Y

Encuesta sobre la gestión y control de celulares corporativos: Tiempo de atención en la devolución del equipo

PRE TEST									
Sujeto	P1	P2	P3	P4	P5	P6	P7	P8	P9
1	0	1	1	2	5	1	1	3	4
2	0	2	1	1	4	2	1	1	4
3	0	1	1	1	4	1	1	4	3
4	0	1	1	2	5	1	1	4	3
5	0	1	1	1	4	1	1	3	3
6	0	1	2	2	5	2	1	4	2

POST TEST									
Sujeto	P1	P2	P3	P4	P5	P6	P7	P8	P9
1	0	1	1	2	4	1	0	1	0,5
2	0	1	1	1	4	1	0	1	0,5
3	0	1	2	1	3	1	0	1	0,5
4	0	1	1	1	4	1	0	1	0,5
5	0	1	1	1	4	1	0	1	0,5
6	0	1	1	2	4	1	0	1	0,5

Coefficiente de correlación de Pearson: $r = 0,67$

R de Pearson



"Una relación estadística, por más fuerte y sugerente que sea, **nunca podrá establecer una conexión causal**: nuestras ideas de causalidad deben provenir de estadísticas externas y, en último término, de una u otra teoría"
 M. G. Kendall y A. Stuart, 1961.

Universidad Peruana Unión

Desarrollo de un sistema web utilizando el Framework LoopBack 4 y NuxtJS para la gestión y control de celulares corporativos de la UPeU - Campus Juliaca

Validación de Instrumentos de Recolección de Datos.

Encuesta sobre la gestión y control de celulares corporativos antes y después del desarrollo del Sistema Web.

1. Tiempo de atención en la entrega de equipo.
2. Tiempo de atención en la devolución del equipo.

Encuesta sobre el uso del Sistema Web.

Apellidos y nombres del evaluador: Gutierrez Quispe Eder

DNI: 43974466

Especialidad: Ingeniero de Sistemas

Opinión de aplicabilidad: Aplicable() Aplicable después de corregir() No aplicable()

Firma:



Apellidos y nombres del juez evaluador:

Jumbica Churo Jaime

DNI: 47714082

Especialidad: Ing Estadístico e Informativo

Opinión de aplicabilidad: Aplicable() Aplicable después de corregir() No aplicable()

Firma:

