

Universidad de las Ciencias Informáticas

Facultad 4



**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO
DE INGENIERÍA EN CIENCIAS INFORMÁTICAS**

Título: Sistema de Gestión de Medios Tecnológicos
de la Facultad 4

Autores:

Roxana Mérida Sánchez
Jesús Francisco Vázquez Biltre

Tutor:

M. Sc. Yordankis Matos López.
Ing. José Emilio Badía Valdés

Cotutor:

M. Sc. Osmany Aguilera Almaguer

La Habana, 2022

“Año 64 de la Revolución”

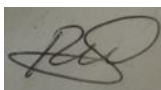
“Nunca renuncies a un sueño por el tiempo que se requiere para lograrlo. El tiempo pasará de todas formas. “

Earl Nightingale

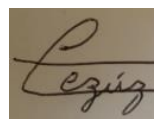
Declaración de autoría

Declaramos por este medio que somos los autores principales de la tesis “Sistema de gestión de los medios tecnológicos”, y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmamos la presente declaración jurada de autoría en La Habana a los 16 días del mes de noviembre del año 2022.



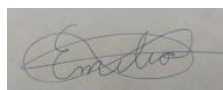
Autora: Roxana Mérida Sánchez



Autor: Jesús Francisco Vázquez Biltre



Tutor: M. Sc. Yordankis Matos López



Tutor: Ing. José Emilio Badía Valdés

Dedicatoria

Roxana

Dedico esta tesis primeramente a mi madre Nilvia, que ha estado batallando a mi lado desde el primer día, guiándome y apoyándome incondicionalmente, a mi padre Mario que también me ha acompañado. A mi tía Maruchi que a pesar de la distancia no se ha alejado de mi lado ni un segundo y por sobre todas las cosas, a mis hijos, Mía y Darío, los motores que me han impulsado a esforzarme y ser mejor cada día.

Jesús

Dedico esta tesis a mi madre Mariluz y a mi padre Jesús que siempre me han apoyado. Y a mí por el esfuerzo realizado.

Agradecimientos

Roxana

Este logro fue posible gracias a una serie de personas importantes en mi vida que llevaré marcadas para siempre.

Gracias a mis padres, en especial a mi mamá, sin ella no hubiese llegado hasta aquí, gracias por tu apoyo constante. Gracias por tus peleas para no abandonar ni desmoronarme, por tu paciencia, por tu guía, por enseñarme que los problemas se enfrentan y que los tropiezos son para levantarse.

A mi tía Maruchi, que también me ha estado apoyando día a día, que me enseñó a sacudir el polvo y volver a intentarlo. Me impulsó a no rendirme. Gracias por estar conmigo, por tus consejos, y por esos alones de orejas de lejos.

Gracias a mi familia en general, primos, tíos que han estado pendientes de cada paso. A los amigos que me han acompañado, a los que siempre me mantienen informada y se han preocupado por mí y mis hijos.

A mi esposo, Lázaro, que ha tenido tanta paciencia en estos meses y aguantar mis peleas y estrés, y apoyarme en este recorrido tan importante para mí.

Gracias a muchos profesores que me dieron la mano y entendieron. En especial a Yadira, parte importante en mi preparación. Al profesor Vásquez que me apoyó muchísimo también. A mi compañero de tesis. A mis tutores, gracias.

Gracias a Dios que no me ha abandonado, gracias a mis hijos por su amor.

A todos muchas gracias.

Agradecimientos

Jesús

Agradezco profundamente a todos los que me han ayudado, en especial a mis amigos Dennis, Julio, Daniel, Dayana, Carlos. No puedo dejar de mencionar a mis tíos Héctor, Raúl y José Ángel y mis tías Maritza, Milagro y Mariluz que siempre me han apoyado también. A mis hermanas Lianet y Lisbeth, mis primas Elizabet y Isbel y mi primo Carlos Rafael. Y en especial a mi abuela Alicia.

Gracias por todo el apoyo

Resumen

La evolución de las tecnologías de la información y las comunicaciones (TICs) ha revolucionado no sólo las empresas sino también la vida en general. Nuestra sociedad, nuestras escuelas, nuestros hábitos, y todo lo que nos rodea ha experimentado una gran evolución. [1] Las TICs han mostrado ser de gran importancia para la informatización de la sociedad actual a partir del desarrollo de los distintos sectores de esta. En nuestro país esto quedó evidenciado con la creación en 1987 de los Joven Club de Computación y Electrónica, así como la constitución de la Universidad de las Ciencias Informáticas (UCI) en el año 2002, la fundación del Ministerio para la Informática y las Comunicaciones, entre otras instituciones.

La alta disponibilidad tecnológica que posee la UCI posibilita a los estudiantes, profesores y especialistas alcanzar un alto nivel de desarrollo para la creación de sistemas informáticos dirigidos a diferentes entidades tanto nacionales como internacionales. Con el objetivo de continuar con el proceso de lograr la mayor informatización en todas las áreas de la Universidad se realiza un estudio del proceso de gestión de medios tecnológicos de la Facultad 4, el cual se lleva a cabo en el departamento de tecnología. Este proceso puede traer algunos problemas ya que es un proceso que se realiza de manera manual, conllevando a la necesidad de realizarlo entre dos personas, pudiendo en ocasiones perderse información, o guardar información incorrecta.

Por tanto, se pretende realizar un sistema que permita la mejora del proceso de gestión de los medios tecnológicos de la Facultad 4 de la UCI, y logre cubrir las carencias que manifiestan los antecedentes para responder a dicha necesidad.

Palabras clave: sistema, medios tecnológicos, TICs.

Abstract

The evolution of information and communications technologies (ICTs) has revolutionized not only businesses but also life in general. Our society, our schools, our habits, and everything around us has undergone a great evolution. [1] ICTs have proved to be of great importance for the informatization of today's society through the development of its different sectors. In our country this was evidenced by the creation in 1987 of the Young Computer and Electronics Clubs, as well as the establishment of the University of Informatics Sciences (UCI) in 2002, the foundation of the Ministry of Informatics and Communications, among other institutions.

The high technological availability of the UCI allows students, professors and specialists to reach a high level of development for the creation of computer systems for different national and international entities. In order to continue with the process of achieving greater computerization in all areas of the University, a study of the process of management of technological means of Faculty 4, which is carried out in the technology department, is carried out. This process can bring some problems because it is a process that is done manually, leading to the need to perform it between two people, and sometimes information can be lost, or incorrect information can be stored.

Therefore, it is intended to develop a system that will improve the management process of the technological resources of the Faculty 4 of the UCI, and cover the shortcomings of the background to respond to this need.

INDICE

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.....	5
1.1. Conceptos fundamentales.....	5
1.2. Sistemas informáticos existentes asociados a la gestión de medios tecnológicos.....	6
1.3. Metodología, herramientas y tecnologías a utilizar.....	9
1.3.1. Metodologías.....	9
1.3.2. Lenguajes de programación.....	12
1.3.3. Framework.....	13
1.3.4. Sistema gestor de base de datos.....	15
1.3.5. Entorno de desarrollo integrado.....	16
1.3.6. Sistema de control de versiones.....	17
1.4. Conclusiones del capítulo.....	19
CAPITULO 2. características y diseño del sistema.....	20
2.1. Descripción de la propuesta de solución.....	20
2.2. Descripción de las historias de usuario.....	20
2.3. Plan de iteraciones.....	28
2.4. Plan de entrega.....	29
2.5. Arquitectura de software.....	29
2.6. Patrones de diseño.....	30
2.7. Tarjetas Clase-Responsabilidad-Colaborador (CRC).....	33
2.8. Modelo de datos.....	42
2.9. Conclusiones del capítulo.....	44
Capítulo 3. Implementación y pruebas.....	45
3.1. Estándares de codificación.....	45
3.2. Tareas de ingeniería.....	45
Iteración 1.....	46
3.3. Pruebas.....	47
3.3.1. Pruebas unitarias.....	48
3.3.2. Pruebas de aceptación.....	50
3.4. Resultados de las pruebas.....	62
3.5. Funcionamiento del sistema.....	65
Aplicación web.....	65
Aplicación móvil.....	66
Aplicación de escritorio.....	67
3.6 Conclusiones del capítulo.....	68
Conclusiones generales.....	69

Recomendaciones	70
Bibliografía.....	71
Anexos	74
Anexo 1. Tareas de ingeniería	74
Anexo 2. Entrevista a Especialistas del área de informatización.	83
Anexo 3. Tabla comparativa metodologías.....	84
Anexo 4. Tabla comparativa lenguajes de programación.	85
Anexo 5. Tabla comparativa framework.	87
Anexo 6. Tabla comparativa sistema gestor de base de datos.	89
Anexo 7. Tabla comparativa entorno de desarrollo integrado.....	90
Anexo 8. Encuesta realizada a especialistas del área de informatización.	91
Anexo 9. Control de versiones. Aplicación móvil.	92
Anexo 10. Control de versiones. Aplicación cliente.	93
Anexo 11. Control de versiones. Aplicación web.	94

ÍNDICE DE TABLAS, FIGURAS, ILUSTRACIONES

Índice de Tablas

Tabla 1.	Comparación entre metodologías.....	10
Tabla 2.	Historia de usuario # 1.	20
Tabla 3.	Historia de usuario # 2.	21
Tabla 4.	Historia de usuario # 3.	22
Tabla 5.	Historia de usuario # 4.	23
Tabla 6.	Historia de usuario # 5.	23
Tabla 7.	Historia de usuario # 6.	24
Tabla 8.	Historia de usuario # 7.	24
Tabla 9.	Historia de usuario # 8.	25
Tabla 10.	Historia de usuario # 9.	25
Tabla 11.	Historia de usuario # 10.	26
Tabla 12.	Historia de usuario # 11.	27
Tabla 13.	Historia de usuario # 12.	28
Tabla 14.	Historia de usuario # 13.	28
Tabla 15.	Plan de Iteraciones.	29
Tabla 16.	Plan de entrega.....	29
Tabla 17.	Tarjeta CRC1 Servidor web.....	33
Tabla 18.	Tarjeta CRC 2 Servidor web.....	34
Tabla 19.	Tarjeta CRC 3 Servidor web.....	35
Tabla 20.	Tarjeta CRC 4 Servidor web.....	35
Tabla 21.	Tarjeta CRC 5 Servidor web.....	36
Tabla 22.	Tarjeta CRC 6 Servidor web.....	36
Tabla 23.	Tarjeta CRC 7 Servidor web.....	36
Tabla 24.	Tarjeta CRC 8 Servidor web.....	37
Tabla 25.	Tarjeta CRC 8 Servidor web.....	37
Tabla 26.	Tarjeta CRC 9 Servidor web.....	38
Tabla 27.	Tarjeta CRC 10 Servidor web.....	38
Tabla 28.	Tarjeta CRC 11 Servidor web.....	38
Tabla 29.	Tarjeta CRC 12 Servidor web.....	38
Tabla 30.	Tarjeta CRC 13 Servidor web.....	39
Tabla 31.	Tarjeta CRC 14 Servidor web.....	39
Tabla 32.	Tarjeta CRC 15 Servidor web.....	39
Tabla 33.	Tarjeta CRC 15 Servidor web.....	40
Tabla 34.	Tarjeta CRC 16 Servidor web.....	40
Tabla 35.	Tarjeta CRC 17 Aplicación móvil.....	41

Tabla 36.	Tarjeta CRC 18 Aplicación móvil	41
Tabla 37.	Tarjeta CRC 19 Aplicación móvil	41
Tabla 38.	Tarjeta CRC 20 Aplicación móvil	41
Tabla 39.	Tarjeta CRC 21 Aplicación cliente	42
Tabla 40.	Tarea de Ingeniería # 1	46
Tabla 41.	Tarea de Ingeniería # 2	46
Tabla 42.	Tarea de Ingeniería # 3	46
Tabla 43.	Tarea de Ingeniería # 4	46
Tabla 44.	Tarea de Ingeniería # 5	47
Tabla 45.	Tarea de Ingeniería # 6	47
Tabla 46.	Prueba de aceptación # 1	50
Tabla 47.	Prueba de aceptación # 2	50
Tabla 48.	Prueba de aceptación # 3	51
Tabla 49.	Prueba de aceptación # 4	51
Tabla 50.	Prueba de aceptación # 5	51
Tabla 51.	Prueba de aceptación # 6	52
Tabla 52.	Prueba de aceptación # 7	52
Tabla 53.	Prueba de aceptación # 8	53
Tabla 54.	Prueba de aceptación # 9	53
Tabla 55.	Prueba de aceptación # 10	54
Tabla 56.	Prueba de aceptación # 11	54
Tabla 57.	Prueba de aceptación # 12	54
Tabla 58.	Prueba de aceptación # 13	55
Tabla 59.	Prueba de aceptación # 14	55
Tabla 60.	Prueba de aceptación # 15	56
Tabla 61.	Prueba de aceptación # 16	56
Tabla 62.	Prueba de aceptación # 17	56
Tabla 63.	Prueba de aceptación # 18	57
Tabla 64.	Prueba de aceptación # 19	57
Tabla 65.	Prueba de aceptación # 20	58
Tabla 66.	Prueba de aceptación # 21	58
Tabla 67.	Prueba de aceptación # 22	58
Tabla 68.	Prueba de aceptación # 23	59
Tabla 69.	Prueba de aceptación # 24	59
Tabla 70.	Prueba de aceptación # 25	60
Tabla 71.	Prueba de aceptación # 26	60
Tabla 72.	Prueba de aceptación # 27	60
Tabla 73.	Prueba de aceptación # 28	61

Tabla 74.	Prueba de aceptación # 29	61
Tabla 75.	Prueba de aceptación # 30	61
Tabla 76.	Tarea de Ingeniería # 7	74
Tabla 77.	Tarea de Ingeniería # 8	74
Tabla 78.	Tarea de Ingeniería # 9	74
Tabla 79.	Tarea de Ingeniería # 10	74
Tabla 80.	Tarea de Ingeniería # 11	75
Tabla 81.	Tarea de Ingeniería # 12	75
Tabla 82.	Tarea de Ingeniería # 13	75
Tabla 83.	Tarea de Ingeniería # 14	76
Tabla 84.	Tarea de Ingeniería # 15	76
Tabla 85.	Tarea de Ingeniería # 16	76
Tabla 86.	Tarea de Ingeniería # 17	76
Tabla 87.	Tarea de Ingeniería # 18	77
Tabla 88.	Tarea de Ingeniería # 19	77
Tabla 89.	Tarea de Ingeniería # 20	77
Tabla 90.	Tarea de Ingeniería # 21	77
Tabla 91.	Tarea de Ingeniería # 22	78
Tabla 92.	Tarea de Ingeniería # 23	78
Tabla 93.	Tarea de Ingeniería # 24	78
Tabla 94.	Tarea de Ingeniería # 25	78
Tabla 95.	Tarea de Ingeniería # 26	79
Tabla 96.	Tarea de Ingeniería # 27	79
Tabla 97.	Tarea de Ingeniería # 28	79
Tabla 98.	Tarea de Ingeniería # 29	80
Tabla 99.	Tarea de Ingeniería # 30	80
Tabla 100.	Tarea de Ingeniería # 31	80
Tabla 101.	Tarea de Ingeniería # 32	80
Tabla 102.	Tarea de Ingeniería # 33	81
Tabla 103.	Tarea de Ingeniería # 34	81
Tabla 104.	Tarea de Ingeniería # 35	81
Tabla 105.	Tarea de Ingeniería # 36	82
Tabla 106.	Tarea de Ingeniería # 37	82
Tabla 107.	Tarea de Ingeniería # 38	82
Tabla 108.	Tarea de Ingeniería # 39	82

Índice de Ilustraciones

Ilustración 1.	<i>Software OCS Inventory NG</i> . [14]	7
Ilustración 2.	<i>Software Open-Audit</i> . [16]	8
Ilustración 3.	<i>Software GRHS</i> . [17].....	8
Ilustración 4.	Arquitectura Cliente – Servidor. [39].....	30
Ilustración 5.	Código ejemplo patrón Experto, Alta Cohesión.	32
Ilustración 6.	Código ejemplo patrón Creador.	32
Ilustración 7.	Código ejemplo patrón Controlador.....	33
Ilustración 8.	Diagrama entidad-relación	43
Ilustración 9.	Aplicación de la prueba unitaria Iteración 1.	48
Ilustración 10.	Aplicación de la prueba unitaria Iteración 2.	49
Ilustración 11.	Aplicación de la prueba unitaria Iteración 3.	49
Ilustración 12.	Aplicación de la prueba unitaria Iteración 4.	49
Ilustración 13.	Código ejemplo servidor web.	65
Ilustración 14.	Vista ejemplo página web.	66
Ilustración 15.	Vista ejemplo aplicación móvil.....	67
Ilustración 16.	Código ejemplo aplicación cliente.	67
Ilustración 17.	Código ejemplo aplicación cliente.	68
Ilustración 18.	Control de versiones apk.....	92
Ilustración 19.	Control de versiones des.....	93
Ilustración 20.	Control de versiones web.....	94

Índice de gráficos

Gráfico 1.	Pruebas unitarias.	62
Gráfico 2.	Pruebas de aceptación.....	63

Introducción

Los cambios que nuestra sociedad ha experimentado en los últimos años han sido enormes. La evolución de las tecnologías de la información y las comunicaciones (TICs) ha revolucionado no sólo las empresas sino también la vida en general. Nuestra sociedad, nuestras escuelas, nuestros hábitos, y todo lo que nos rodea ha experimentado una gran evolución. [1]

Dicha evolución ha provocado que la utilización de medios tecnológicos sea indispensable en nuestros días, lo que ha brindado gran agilidad y eficiencia en la resolución de problemas. Demostrando así su importancia en la sociedad moderna que nos rodea.

Es común emplear los medios tecnológicos en áreas como la administración de empresas, la economía o incluso la educación, sea cual sea su contexto de aplicación. Constituyendo en el mundo hipertecnológico de hoy uno de los factores más determinantes en la actividad humana, es decir, suelen marcar la diferencia entre el éxito y el fracaso. [2]

En el caso de Cuba se está aplicando la informatización de una forma más eficaz, eficiente y competitiva en todas sus esferas y procesos de la sociedad. En este sentido se ha identificado la conveniencia y necesidad de dominar e introducir en la práctica social las TICs y lograr una cultura digital como un factor clave en el desarrollo progresivo de la población. Lo anterior facilitará a nuestra sociedad acercarse más hacia el objetivo de un progreso equitativo.

Las TICs han mostrado ser de gran importancia para la informatización de la sociedad actual a partir de sus distintos sectores. En nuestro país esto quedó evidenciado con la creación en 1987 de los Joven Club de Computación y Electrónica; el establecimiento de una red de personas e instituciones que trabajan y colaboran para facilitar el acceso a la información y el conocimiento, necesarios para mejorar la salud de los cubanos y de los pueblos del mundo conocida como INFOMED en 1992; la constitución de la Universidad de las Ciencias Informáticas (UCI) en el año 2002, la fundación del Ministerio para la Informática y las Comunicaciones, la Unión de Informáticos de Cuba (UIC) y la incorporación de la enseñanza de la computación, masiva y gratuita, en las escuelas.

Como se mencionó anteriormente, la UCI fue creada en el 2002 como parte de los programas de la Batalla de Ideas, a partir de la cual se emprendieron nuevos programas destinados a elevar el nivel cultural de la población y su calidad de vida [3]. Esta tiene como objetivo producir aplicaciones y servicios informáticos, a partir de la

vinculación estudio-trabajo como modelo de formación; y servir de soporte a la industria cubana de la informática. Todo esto ha posibilitado el desarrollo de varios proyectos, a través de los cuales, se han informatizado una parte de los principales procesos que se llevan a cabo tanto del centro educacional como en la sociedad cubana e incluso a nivel internacional.

La alta disponibilidad tecnológica que posee la UCI posibilita a los estudiantes, profesores y especialistas alcanzar un alto nivel de desarrollo para la creación de sistemas informáticos dirigidos a diferentes entidades tanto nacionales como internacionales. La Universidad está constituida por seis facultades, las cuales están integradas a su vez por un conjunto de centros de desarrollo de software. Los objetivos de estas entidades se centran en torno a la misión de la Universidad y cada uno realiza su aporte en la informatización del país especializándose en diferentes áreas.

Con el objetivo de seguir con el proceso de lograr la mayor informatización en todas las áreas de la Universidad. Se realiza un estudio del proceso de gestión de medios tecnológicos de la Facultad 4, el cual se lleva a cabo en el departamento de tecnología de la misma. Revelando los siguientes problemas:

- Se desarrollan un conjunto de procesos de manera manual utilizando documentos en formato duro u hojas de cálculo electrónicas.
- En ocasiones se cometen errores al registrar o controlar un medio tecnológico, haciendo que la labor de control, además de lenta, sea extensa y agotadora.
- Es posible que la información se duplique, se pierda o sea fácilmente modificable al existir un alto riesgo de acceso, esto posibilita la ejecución de hechos delictivos, pérdidas económicas y de manera general se afecta la actividad productiva del área.
- No es posible generar de forma rápida reportes estadísticos sobre el estado actual de la tecnología en el centro y en general en la facultad, que apoyen a los directivos en la toma de decisiones.
- Los cambios en el hardware no se detectan en tiempo.
- Para realizar el conteo en un local generalmente se realiza entre dos personas, uno para buscar el número de medio básico en el papel y el otro diciéndole el número de medio básico.

Teniendo en cuenta las causas anteriores se determinó como **situación problemática** la necesidad de desarrollar un sistema que permita que mejore el proceso de gestión

de los medios tecnológicos de la Facultad 4 de la UCI, y logre cubrir las carencias que manifiestan los antecedentes para responder a dicha necesidad.

A raíz de lo antes planteado, se identifica como **problema de investigación**: ¿Cómo mejorar el proceso de gestión de los medios tecnológicos de la Facultad 4 en la UCI?

Como **objeto de estudio** se definen los procesos de monitoreo y control de los medios tecnológicos. Para dar solución al problema de investigación planteado, se define como **objetivo general**: desarrollar un sistema informático que mejore el proceso de gestión de los medios tecnológicos de la Facultad 4 en la UCI. Dentro del objeto de estudio de la investigación se precisa como **campo de acción**: Los sistemas de monitoreo y control de medios tecnológicos.

Se define como **idea a defender**: con el desarrollo de un sistema informático se contribuirá la gestión de los medios tecnológicos de la Facultad 4 de la UCI.

Para darle solución a la propuesta se definen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación sobre la gestión de medios tecnológicos de la Facultad 4 de la UCI.
2. Diseñar un sistema informático para la gestión de medios tecnológicos de la Facultad 4.
3. Implementar y validar el sistema informático para la gestión de medios tecnológicos de la Facultad 4 mediante pruebas de software.

Y como **métodos investigativos** se utilizan los siguientes:

Teóricos:

Analítico – Sintético: permitió procesar a fondo el flujo de información del Departamento de Tecnología de la Facultad 4, posibilitando así el procesamiento y diferenciación de la misma, enfocada hacia la investigación, permitiendo organizar y simplificar el análisis de todo el volumen de información.

Modelación: se utilizó para realizar el modelado del sistema que está estructurado por una aplicación web, una móvil y una de escritorio; las cuales se desarrollaron aplicando la metodología Extreme Programming (XP), a través de los artefactos correspondientes a las fases: Requisitos, Análisis y Diseño y prueba.

Empíricos:

Observación: se utilizó para obtener una información más precisa de cómo se realiza proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI.

Entrevista: se empleó con el objetivo de conocer la estructura organizativa del proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI.

Encuesta: se aplicó para la recopilación de información sobre el proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI.

El presente trabajo se encuentra estructurado de la siguiente forma:

Capítulo 1. FUNDAMENTACIÓN TEÓRICA: Se definen elementos relacionados con la ingeniería de software que permiten establecer la metodología para guiar el proceso de desarrollo del sistema y los artefactos necesarios durante el proceso. Se proponen las herramientas, lenguajes y tecnologías para el desarrollo del sistema.

Capítulo 2. CARACTERÍSTICAS Y DISEÑO DEL SISTEMA: Se describen las principales características que presenta el sistema a desarrollar, los principales componentes del mismo y su funcionamiento, mediante el marco de trabajo propuesto por la metodología de desarrollo seleccionada.

Capítulo 3. IMPLEMENTACIÓN Y PRUEBAS: Se describen el proceso de implementación y las pruebas realizadas al sistema para su validación.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS.

En el presente capítulo se definen los principales conceptos asociados a la solución en cuestión. Se realiza un análisis de las principales soluciones existentes para así identificar posibles características de las mismas que puedan aportar al diseño y desarrollo de la aplicación. Por último, se expone la metodología, así como las herramientas y tecnologías que se utilizarán en la implementación y el futuro despliegue.

1.1. Conceptos fundamentales.

Medio:

Un medio es algo que sirve para alcanzar un cierto fin y un accionar propicio y útil para obtener una cosa que se busca: “Mi padre trabaja en un medio de comunicación”, “Lo malo de este pueblo es que existen pocos medios de transporte”, “Las conversaciones de café suelen ser el medio para cerrar grandes negocios”. [5]

Medio Tecnológico:

Es un medio que se vale de la tecnología para cumplir con su propósito. Los recursos o medios tecnológicos pueden ser tangibles (como una computadora, una impresora u otra máquina) o intangibles (un sistema, una aplicación virtual). [6]

Gestión:

Del latín *gestiō*, el concepto de gestión hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera. [7]

Página Web:

Se conoce como página web al documento que forma parte de un sitio web y que suele contar con enlaces (también conocidos como hipervínculos o links) para facilitar la navegación entre los contenidos. [8]

Por otra parte, el anterior autor refiere que las páginas web están desarrolladas con lenguajes de marcado como el HTML, que pueden ser interpretados por los navegadores. De esta forma, las páginas pueden presentar información en distintos formatos (texto, imágenes, sonidos, videos, animaciones), estar asociadas a datos de estilo o contar con aplicaciones interactivas.

Sitio Web:

Sitio es un lugar que sirve para algo o un espacio ocupado (o que puede llegar a serlo). La noción de Web, por su parte, hace referencia a Internet, una red de redes que permite la interconexión de computadoras mediante un conjunto de protocolos denominado TCP/IP. Un sitio web, por lo tanto, es un espacio virtual en Internet. Se trata de un conjunto de páginas web que son accesibles desde un mismo dominio o subdominio. [9]

Servidor Web:

Un servidor web o Servidor HTTP es una pieza de *software* de comunicaciones que intermedia entre el servidor en el que están alojados los datos solicitados y el computador del cliente, permitiendo conexiones bidireccionales o unidireccionales, síncronas o asíncronas, con cualquier aplicación del cliente, incluso con los navegadores que traducen un código (renderizable) a una página web determinada. O sea, se trata de programas que median entre el usuario de Internet y el servidor en donde está la información que solicita.[10]

Aplicación móvil:

Es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles y que permite al usuario efectuar una tarea concreta de cualquier tipo profesional, de ocio, educativa, de acceso a servicios, etc., facilitando las gestiones o actividades a desarrollar. [11]

Aplicaciones cliente:

Una aplicación cliente es un paquete de software que funciona sobre el propio sistema operativo de un dispositivo (que puede ser por ejemplo un *smartphone*, una laptop o un equipo de escritorio). Es decir, que la aplicación se instala y corre «dentro» del computador, o lo que solemos llamar también «de forma local». En un computador, estas aplicaciones se instalan en el disco duro, donde guardan toda la información. [12]

1.2. Sistemas informáticos existentes asociados a la gestión de medios tecnológicos.

En el presente subepígrafe se describen un grupo de herramientas relacionadas a la gestión de medios tecnológicos. El estudio de las mismas se utilizará para establecer las bases de la propuesta de solución.

OCS Inventory NG

Es un *software* gratuito que permite a los usuarios inventariar los activos de Tecnología de Información (TI). OCS Inventory NG recopila información sobre el *hardware* y el *software* de las máquinas que ejecutan el programa cliente OCS (Agente de inventario OCS). OCS Inventory NG puede visualizar el inventario a través de una interfaz web e incluye la capacidad de implementar aplicaciones en computadoras de acuerdo con los criterios de búsqueda. Algunas de sus características claves son, llevar información relevante del inventario, escaneo de la red, soporte de múltiples sistemas operativos (Windows, Linux, BSD, Sun Solaris, IBM AIX, HP-UX, MacOSX). [13]



Ilustración 1. *Software OCS Inventory NG.* [14]

Open-Audit

Es una aplicación para contar exactamente qué está en tu red, cómo está configurado y cuando cambia. Open-Audit se ejecutará en sistemas Windows y Linux. Esencialmente, Open-Audit es una base de datos de información, que se puede consultar a través de una interfaz web. Los datos sobre la red se insertan a través de un Bash Script (Linux) o VBScript (Windows). Toda la aplicación está escrita en php, bash y vbscript. Todos estos son lenguajes de 'scripting': sin compilación y código fuente legible para humanos. Hacer cambios y personalizaciones es rápido y fácil.

Open-Audit se puede configurar para escanear su red y dispositivos automáticamente. Se recomienda un escaneo diario para sistemas, con escaneos de red cada dos horas. Con esta configuración, puede estar seguro de recibir una notificación si algo cambia (día a día) en una PC, o incluso antes, si aparece algo "nuevo" en su red. [15] No permite llevar registros sucesivos de los cambios hechos en los componentes de una computadora.

The screenshot shows the Open-Audit web interface. At the top, there is a navigation bar with 'HOME', 'QUERIES', 'ADMIN', and 'HELP'. Below it, the title 'List Devices' is displayed. A table lists various systems with columns for 'Hostnames', 'Descripciones', 'IP Address', 'Type', 'OS / Device', and 'Tags'. A tooltip is visible over the 'system-3' row, showing details for a 'Stallion Production' device, including manufacturer 'VMware, Inc.', model 'VMware Virtual Platform', and serial number '32345678'.

Hostnames	Descripciones	IP Address	Type	OS / Device	Tags
system-1	Workstation	172.16.0.1		Microsoft Windows XP Professional	172.16.0.0, All, Windows, XP
system-2		172.16.0.10		Debian Sarge	172.16.0.0, All, Debian, Linux
system-3				Stallion Production Manufacturer: VMware, Inc. Model: VMware Virtual Platform Serial: 32345678	All, Linux, Virtual
system-2				7 Ultimate	172.16.0.0, All, Win7, Windows
system-3					172.16.0.0, All, Linux, Ubuntu
system-4					10.255.0.0, All, FreeBSD
system-5	Printer	10.255.0.5		HP Laserjet 4300n	10.255.0.0, All, Printers
system-6	Workstation	192.168.0.6		Microsoft Windows XP Professional	192.168.0.0, All, Windows, XP
system-7		10.255.0.7		Microsoft Windows 2000 Server	10.255.0.0, All, Win 2000, Win S
system-8	Router	192.168.0.8		Cisco 1843 Router (IOS v15.1)	192.168.0.0, All, Routers
system-9	Router	192.168.0.9		Cisco 1843 Router (IOS v15.1)	192.168.0.0, All, Routers

Ilustración 2. Software Open-Audit. [16]

El Gestor de Recursos de Hardware y Software (GRHS)

Permite realizar el inventario de *hardware* y *software* de las computadoras donde se encuentre instalado, así como la detección de algunas incidencias que se puedan producir en los clientes alertando a los interesados. Está basado en la arquitectura cliente-servidor con el objetivo de realizar el inventario de hardware y software en una red de computadoras.

Está compuesta por tres aplicaciones: *gclient*, *gserver* y *gadmin*. La aplicación *gclient* se encarga de recopilar la información de los clientes, detectar los cambios, detectar las incidencias y tomar las acciones de control. La aplicación *gserver* recopila la información de todos los inventarios, almacena las configuraciones y tiene la información de todos los clientes. La aplicación *gadmin* es la consola de administración de GRHS. Esta es utilizada para establecer las configuraciones, enviar órdenes a los clientes, mostrar reportes de los inventarios y mostrar la información de los clientes. Esta solución fue desarrollada en el Centro de Telemática de la Universidad de las Ciencias Informáticas, Cuba. [17]

The screenshot shows the GRHS web interface. At the top, there is a navigation bar with 'Sistema', 'Agentes', 'Inventario', 'Incidencias', 'Acción', 'Traces', and 'Configuración'. Below it, the title 'Colectores activos' is displayed. A table lists various hardware components with columns for 'Número de Serie', 'Versión', 'Velocidad', 'Familia', and 'Localización'. The table shows details for a 'Monitor' device with serial number '230F3000FF7E8BF', version 'Intel(R) Pentium(R) 4 CPU 2.40GHz', speed '2400', family 'Pentium 4', and location 'Doc2Fac-Lab315'.

Número de Serie	Versión	Velocidad	Familia	Localización
230F3000FF7E8BF	Intel(R) Pentium(R) 4 CPU 2.40GHz	2400	Pentium 4	Doc2Fac-Lab315

Ilustración 3. Software GRHS. [17]

Después de haber profundizado sobre las características de las aplicaciones anteriores. Se observa, que la propuesta de solución ofrece funcionalidades que estas no presentan como: tener una aplicación móvil que facilite el proceso de chequeo de los medios tecnológicos, la impresión de la ficha técnica de los medios tecnológicos

con un código QR, y un expediente. Open Audit no permite llevar registros sucesivos de los cambios hechos en los componentes de una computadora. Así mismo, son aplicaciones que con el tiempo han quedado obsoletas.

Por otra parte, se tomaron las siguientes funcionalidades de estos softwares como requisitos funcionales para la solución; tales como, en caso de OCS Inventory y Open-Audit, llevar información relevante del inventario y soporte de múltiples sistemas operativos, mientras que GRHS recopila la información de los clientes y detecta los cambios e incidencias; de manera general, la obtención de datos del *Hardware*, y la forma de mostrar estos datos mediante una interfaz web. De estos también se toma la estructura arquitectónica Cliente-Servidor.

1.3. Metodología, herramientas y tecnologías a utilizar.

Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Esta tarea se vuelve más amena utilizando la metodología, herramientas y tecnologías correctas. Ya que las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. [18] Y las herramientas y tecnologías facilitan la resolución de tareas de manera más rápida y sencilla.

1.3.1. Metodologías.

Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del *software* que se produce en todas y cada una de sus fases de desarrollo. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas o tradicionales. [18]

En las metodologías tradicionales el principal problema es que nunca se logra planificar bien el esfuerzo requerido para seguir la metodología. Pero entonces, si se logra definir métricas que apoyen la estimación de las actividades de desarrollo, muchas prácticas de metodologías tradicionales podrían ser apropiadas. Es importante tener en cuenta que el uso de un método ágil no vale para cualquier proyecto. Sin embargo, una de las principales ventajas es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante agradables.

Tabla 1. Comparación entre metodologías.

Tradicional	Ágil
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
Más artefactos	Pocos artefactos
Más roles	Pocos roles
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Documentación exhaustiva	Poca documentación
Pocos ciclos de entrega	Muchos ciclos de entrega

Como se pudo observar en la tabla anterior, la metodología ágil es la más adecuada, ya que no necesita de amplio equipo de desarrollo, extensa documentación ni tantos artefactos para los procesos de software, el cliente es parte del equipo de desarrollo a diferencia del tradicional.

Podemos definir las metodologías ágiles como un conjunto de tareas y procedimientos dirigidos a la gestión de proyectos. Son aquellos métodos de desarrollo en los cuales tanto las necesidades como las soluciones a estas evolucionan con el pasar del tiempo, a través del trabajo en equipo de grupos multidisciplinarios. [19]

Dentro de las metodologías ágiles destacan las siguientes:

- **Scrum:** la cual trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto por pares de forma periódica, aplicando las buenas prácticas de trabajo colaborativo (en equipo), facilitando el hallazgo de soluciones óptimas a los problemas que pueden ir surgiendo en el proceso de desarrollo del proyecto. Con Scrum se realizan entregas regulares y parciales (sprint) del producto final, todas ellas con una prioridad previamente establecida que nace según el beneficio que aporten al cliente, minimizando los riesgos que pueden surgir de desarrollos extremadamente largos. Es por tal motivo, que Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesitan obtener resultados de manera inmediata y donde son fundamentales los siguientes aspectos: la innovación, la productividad, la flexibilidad y la competitividad. [19]
- **Kanban:** proveniente de una palabra japonesa cuyo significado es “Tarjeta Visual” es un marco de trabajo que requiere una comunicación en tiempo real sobre la capacidad del equipo, utilizado para controlar el avance de trabajo en una línea de producción, en la cual se clasifican las tareas en subestatus, esto con la intención de determinar los niveles de productividad en cada fase del proyecto. Para el desarrollo de software, gracias a su sencillez KANBAN, simplifica la planificación y la asignación de responsabilidades, en un tablero se representan los procesos del flujo de trabajo, cómo mínimo deben existir tres columnas (Pendiente, En Progreso, Terminado), la cantidad de tarjetas en estatus pendiente, forma parte de lo solicitado por el cliente, aquellas colocadas en progreso dependerán de la capacidad del equipo de trabajo. [19]
- **Extreme Programming (XP):** esta tiene como características estar centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. [18]

Después de analizar estas metodologías (*ver Anexo 3*) se optó por utilizar Extreme Programming (XP), ya que está basada en un conjunto de reglas y buenas prácticas para el desarrollo de software en ambientes muy cambiantes. Estos ambientes poseen requisitos imprecisos, por ende, está enfocada en la retroalimentación continua entre

el equipo de desarrollo y el cliente. Entre sus artefactos se encuentran Historias de usuario, tarjetas CRC y tareas de ingeniería.

Las historias de usuario representan una breve descripción del comportamiento del sistema, utilizando lenguaje natural, se realiza para cada característica principal del sistema. Las tarjetas CRC están divididas en tres secciones clases, responsabilidades y colaboradores.

Para la generación de los artefactos se utilizó el Lenguaje de Modelado Unificado (UML) y como herramienta de Ingeniería de *Software* Asistida por Computadora (CASE) *Visual Paradigm* for UML.

1.3.2. Lenguajes de programación.

Un lenguaje de programación es cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por un ordenador o computadora. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. [20]

Kotlin

Es un lenguaje de programación multiplataforma, de tipo estático, de uso general, orientado a objetos que también tiene muchos elementos de programación funcional. Desde el lado orientado a objetos, admite subtipo nominal con polimorfismo paramétrico delimitado (similar a genéricos) y varianza en sitios mixtos. Desde el lado de la programación funcional, tiene soporte de primera clase para funciones de orden superior y literales lambdas. [21]

Dart

Dart es un lenguaje optimizado para el cliente para desarrollar aplicaciones rápidas en cualquier plataforma. Su objetivo es ofrecer el lenguaje de programación más productivo para el desarrollo multiplataforma, junto con una plataforma flexible de tiempo de ejecución para marcos de aplicaciones. Utiliza la verificación de tipo estático para garantizar el valor de una variable, siempre coincide con el tipo estático de la variable, además utiliza un modelo de memoria administrado, donde la memoria no utilizada es recuperada por un recolector de basura. [22]

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y

dinámico. Caracterizado por ser simple, no hace falta tener conocimientos avanzados de programación para aprender a manejar JavaScript y es recomendado por muchos expertos a la hora de encontrar un lenguaje para comenzar a programar. [23]

Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de Lenguaje de Programación C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible. [24]

PHP

Es un lenguaje de programación de código abierto del lado del servidor que se utiliza principalmente para crear páginas web dinámicas. La abreviatura nació originariamente de “Personal Home Page Tools”, aunque hoy en día se ha convertido en el acrónimo recursivo para “PHP: Hypertext Preprocessor”. PHP presenta una sintaxis sencilla fundamentalmente para los principiantes, al igual que ofrece una cantidad remarcable de funciones. Este lenguaje de programación se distingue por su amplio soporte a bases de datos, puede utilizarse en todo tipo de plataformas y está cubierto por una licencia PHP especial que permite su libre utilización y modificación del código fuente.

Para el desarrollo de la solución se emplearán los lenguajes Kotlin como lenguaje de programación para el servidor web, la aplicación cliente; Dart en la construcción de la aplicación móvil y JavaScript en la página web. Ya que las características de estos (ver Anexo 4) permiten una programación estructurada, flexible y una curva de aprendizaje poco inclinada.

1.3.3. Framework.

Un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. [25]

Spring boot

Es un marco de desarrollo, con el cual se trabaja en JAVA, que proporciona un conjunto de herramientas para construir rápidamente aplicaciones de Spring que sean fáciles de configurar. Lo que facilita la creación de aplicaciones autocontenidas, con las que se puede olvidar de la arquitectura y enfocarse únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar el servicio o aplicación a un servidor de aplicaciones y enfocarse únicamente en crear el código. [26]

Apache Struts

Apache Struts es un marco MVC gratuito y de código abierto para crear aplicaciones web Java elegantes y modernas. Favorece la convención sobre la configuración, es extensible utilizando una arquitectura de complemento y se envía con complementos para admitir REST, AJAX y JSON. [27]

Flutter

Flutter es un kit de herramientas de interfaz de usuario multiplataforma diseñado para permitir la reutilización de código en todos los sistemas operativos como iOS y Android, al tiempo que permite que las aplicaciones interactúen directamente con los servicios de plataforma subyacentes. El objetivo es permitir a los desarrolladores ofrecer aplicaciones de alto rendimiento que se sientan naturales en diferentes plataformas, abrazando las diferencias donde existan mientras comparten la mayor cantidad de código posible. [28] En comparación con otras tecnologías de desarrollo móvil, como React Native y Xamarin y Córdoba, es el marco multiplataforma más rápido para construir una aplicación, presenta un alto rendimiento y es el más sencillo de utilizar ya que es más resistente a las actualizaciones del sistema.

React

Es una biblioteca escrita en Java Script de código abierto, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. Se utiliza en Facebook para la producción de componentes, e Instagram está escrito enteramente en React. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos. Según el servicio de análisis JavaScript Libscore, ReactJS es actualmente utilizado en 1455 sitios webs, entre ellos se sitúan productos de las empresas Netflix, PayPal, Airbnb, Uber entre otras. [29]

Después del estudio realizado se selecciona Spring Boot como framework de desarrollo para el servidor web, React JS para el desarrollo de las páginas web y Flutter en el desarrollo de la aplicación móvil. Ya que estos marcos de trabajo son rápidos, flexibles y sus características (*Ver Anexo 5*) permiten mostrar progresos de una manera más ágil y eficiente.

1.3.4. Sistema gestor de base de datos.

Un Sistema Gestor de Base de Datos (SGBD, en inglés DBMS: *Data Base Management System*) es un sistema de *software* que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. [30]

Microsoft SQL Server.

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) que admite una amplia variedad de aplicaciones de procesamiento de transacciones, inteligencia empresarial y análisis en entornos informáticos corporativos. Microsoft SQL Server es una de las tres tecnologías de bases de datos líderes del mercado, junto con Oracle Database y DB2 de IBM. Al igual que otros programas RDBMS, Microsoft SQL Server se basa en SQL, un lenguaje de programación estandarizado que los administradores de bases de datos (DBA) y otros profesionales de TI utilizan para gestionar las bases de datos y consultar los datos que contienen. *SQL Server* está vinculado a Transact-SQL (T-SQL), una implementación de SQL de *Microsoft* que añade un conjunto de extensiones de programación propias al lenguaje estándar. [31]

PostgreSQL.

Sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de Código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (*PostgreSQL Global Development Group*). Este gestor presenta características como alta concurrencia lo que le permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. La posibilidad de que los usuarios creen sus

propios tipos de datos y el soporte de funciones que pueden ser escritos en varios lenguajes diferentes. [32]

MySQL

Es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información. Fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos. También es muy destacable, la condición de *open source* de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente.[33]

El SGBD seleccionado es PostgreSQL debido a que es un sistema de alto rendimiento, flexible y multiplataforma. Además, ofrece gran capacidad de almacenamiento, ahorros considerables de costos de operación y está diseñado para entornos con altos volúmenes de tráfico. (Ver Anexo 6)

1.3.5. Entorno de desarrollo integrado.

Un entorno de desarrollo integrado (IDE) es un sistema de software para el diseño de aplicaciones que combina herramientas comunes para desarrolladores en una sola interfaz de usuario gráfica (GUI). [34]

Visual Studio Code (versión 1.6)

Es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con *Git*, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. Inicialmente incluye un mínimo de componentes y funciones básicas de un editor con soporte nativo para JavaScript/TypeScript y Node.js, sin embargo, es personalizable con los cientos de *plugins* o extensiones disponibles para escribir código en diferentes lenguajes. Incluye una terminal con todas las funciones, la cual se inicia fácilmente en el directorio de trabajo. La terminal integrada puede utilizar cualquier Shell instalado en el equipo, como PowerShell, Bash o cualquier otro. Contar con una terminal en el propio editor es

de gran utilidad para ejecutar diferentes comandos necesarios cuando se esté desarrollando. [35]

Eclipse (versión 4.8)

Es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Mayoritariamente se utiliza para desarrollar lo que se conoce como Aplicaciones de Cliente Enriquecido, entorno de desarrollo integrado, opuesto a las aplicaciones Cliente-liviano, entorno de desarrollo integrado basadas en navegadores. Es una potente y completa plataforma de Programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. No es más que un entorno de desarrollo integrado (IDE) en el que encontrarás todas las herramientas y funciones necesarias para tu trabajo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. [36]

NetBeans (versión 8.2)

Es un entorno de desarrollo integrado de código abierto y gratuito para el desarrollo de aplicaciones en los sistemas operativos Windows, Mac, Linux y Solaris. Simplifica el desarrollo de aplicaciones web, empresariales, de escritorio y móviles que utilizan las plataformas Java y HTML5. Además, ofrece soporte para el desarrollo de aplicaciones PHP y C/C++.

Es el primer IDE que admite las últimas versiones de JDK, Java EE y JavaFX. Brinda resúmenes inteligentes que le ayudan a comprender y administrar sus aplicaciones, incluida la compatibilidad inmediata con tecnologías populares, como Maven. Gracias a sus características de desarrollo de aplicaciones integrales, la mejora constante de Java Editor y las mejoras continuas de velocidad y rendimiento, NetBeans IDE marca el ritmo para el desarrollo de aplicaciones con tecnologías novedosas listas para usar.

Como entorno de desarrollo se seleccionó *Visual Studio Code* ya que ofrece todas las herramientas necesarias para desarrollar la solución informática. Es un software libre con mejor autocompletamiento de código comparado con los editores anteriormente estudiados. Además, tiene alta compatibilidad con los lenguajes y framework a utilizar. (Ver Anexo 7)

1.3.6. Sistema de control de versiones.

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Facilitan la administración de las distintas versiones de cada producto desarrollado. (Ver Anexos 9 y 10)

Git: es un *software* de control de versiones desarrollado por Linus Torvalds, pensando en la eficiencia, confiabilidad y compatibilidad del mantenimiento de controles de versiones de aplicaciones que tienen un gran número de archivos de código fuente. Representa sistema de control de versiones distribuido.

Cada vez que se guarda el trabajo, Git crea una confirmación. Una confirmación es una instantánea de todos los archivos en un momento dado. Si un archivo no ha cambiado de una confirmación a la siguiente, Git usa el archivo almacenado anteriormente. Este diseño difiere de otros sistemas que almacenan una versión inicial de un archivo y mantienen un registro de deltas a lo largo del tiempo.

Entre sus ventajas se encuentran:

Soporte técnico sólido de la comunidad: Git es de código abierto y se ha convertido en el estándar para el control de versiones.

Desarrollo simultáneo: Todos tienen su propia copia local de código y pueden trabajar simultáneamente en sus propias ramas. Git funciona dsin conexión, ya que casi todas las operaciones son locales.

Versiones más rápidas: Al separar la rama de versión del desarrollo en curso, es más fácil administrar código estable y enviar actualizaciones más rápidamente.

Integración integrada: Debido a su popularidad, se integra en la mayoría de las herramientas y productos. Esta integración simplifica el flujo de trabajo diario.

Git funciona con cualquier equipo: El uso de Git con una herramienta de administración de código fuente aumenta la productividad de un equipo fomentando la colaboración, aplicando directivas, automatizando procesos y mejorando la visibilidad y la rastreabilidad del trabajo.

Solicitudes de incorporación de cambios: Use solicitudes de incorporación de cambios para analizar los cambios de código con el equipo antes de combinarlos en la rama principal.

Directivas de rama: Las directivas de rama protegen ramas importantes mediante la prevención de inserciones directas, la necesidad de revisores y la garantía de compilaciones limpias.

1.4. Conclusiones del capítulo.

En el transcurso del capítulo se definieron los fundamentos teóricos que acompañarán a la propuesta de solución del problema planteado, concluyendo que:

- El análisis de los sistemas informáticos asociados a la gestión de medios tecnológicos, definió las principales funcionalidades a tener en cuenta en la solución, y se concluyó que no son posibles utilizarlas en el presente sistema.
- Se seleccionaron los lenguajes de programación Kotlin, JavaScript y Dart ya que estos brindan las mejores comodidades a la hora de implementar la solución.
- Se establecieron Spring boot, React y Flutter como frameworks de desarrollo, al ser la forma más eficiente y rápida para brindar la solución.

CAPITULO 2. CARACTERÍSTICAS Y DISEÑO DEL SISTEMA.

En este capítulo se abordarán los principales aspectos de la propuesta de solución. Para esto se definen los requisitos funcionales, y patrón de diseño por los que se ha de regir la solución.

2.1. Descripción de la propuesta de solución.

Para dar solución a la situación problemática planteada, se desea implementar un sistema de gestión de los medios tecnológicos en la Facultad 4 de la Universidad de las Ciencias Informáticas; este constará de tres elementos: aplicación web, aplicación móvil y aplicación cliente. La aplicación web será la encargada de llevar el inventario y un registro de los posibles cambios de los medios tecnológicos de la Facultad 4; la misma tendrá la funcionalidad de imprimir la ficha técnica que contendrá el código QR requerido para la aplicación móvil, y un expediente para el control de los mismos.

La aplicación cliente recopilará información del medio tecnológico en el que se encuentre instalada y la transmitirá a la aplicación web. La aplicación móvil se empleará en el chequeo del inventario de los medios en los laboratorios, escaneando el código QR que antes se obtuvo desde la aplicación web.

2.2. Descripción de las historias de usuario.

Las Historias de Usuario (HU) son representaciones de requisitos, elaboradas en una o dos frases y recogidas en un lenguaje común y entendible por el usuario. También conocidas como *User Stories* o simplemente US se han convertido en un estándar a la hora de definir requisitos. A continuación, se presentan las historias de usuario definidas para llevar a cabo la implementación de la solución. Algunas de ellas se encuentran agrupadas en CRUD (*Create, Read, Update, Delete*), ya que el framework de desarrollo Spring Boot facilita la creación de aplicaciones CRUD a través de una capa de repositorios CRUD estándar basados en JPA (*Java Persistence API*). Estas estarán agrupadas por la aplicación que pertenecen.

Historias de usuario relacionadas a la aplicación web.

Tabla 2. Historia de usuario # 1.

Número: 1	Nombre: Autenticar usuario
Usuario: usuario, administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 1
Programador asignado: Jesús Francisco Vázquez Biltre	
Descripción: La aplicación web debe permitir al usuario y al administrador	

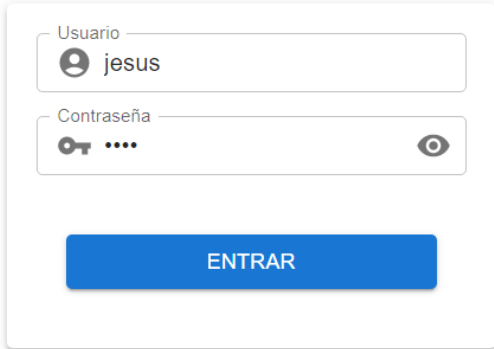
autenticarse en el sistema mediante la entrada del usuario y la contraseña.
<p>Observaciones:</p> <p>Si alguno de los campos está vacío el sistema debe mostrar un mensaje de advertencia.</p>
<p>Prototipo de interfaz:</p> <div style="text-align: center;">  </div>

Tabla 3. Historia de usuario # 2.

Número: 2	Nombre: Gestionar usuario
Usuario: administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteraciones asignadas: 1
Programador asignado: Jesús Francisco Vázquez Biltre	
Descripción: La aplicación web debe ofrecer una vista al administrador con la cual este pueda listar, crear, modificar, eliminar y filtrar los usuarios. Así como gestionar los roles de los mismos.	
Observaciones: Para que el administrador realice esta acción debe de estar autenticado en la aplicación.	
Prototipo de interfaz:	

GMT DISPOSITIVOS LOCALES USUARIOS				jesus	
<input type="checkbox"/> Nombre	Usuario	Rol	Acción		
<input type="checkbox"/> Arian León Benitez	arian	Admin			
<input type="checkbox"/> Carlos Enrique Piñero Cárdenas	carlos	User			
<input type="checkbox"/> Jesús F. Vázquez Biltre	jesus	Admin			

Filas por página: 100 1-3 de 3

Tabla 4. Historia de usuario # 3.

Número: 3	Nombre: Gestionar local																														
Usuario: usuario, administrador																															
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo																														
Puntos estimados: 2	Iteraciones asignadas: 2																														
Programador asignado: Jesús Francisco Vázquez Biltre																															
Descripción: La aplicación web debe brindar al usuario y al administrador una vista donde se permita listar y filtrar los locales registrados en el sistema. Así como permitir que el administrador realice las acciones de crear, modificar y borrar los locales.																															
Observaciones: Para realizar estas acciones tanto el usuario como el administrador deben de estar autenticados.																															
Prototipo de interfaz:																															
<table border="1"> <thead> <tr> <th colspan="4">GMT DISPOSITIVOS LOCALES USUARIOS</th> <th colspan="2">jesus</th> </tr> <tr> <th><input type="checkbox"/> Número o Nombre</th> <th>Ip</th> <th>Número de Dispositivo</th> <th colspan="3">Acción</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> Laboratorio 201</td> <td>10.8.44.00</td> <td>0</td> <td></td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/> Laboratorio 202</td> <td>127.0.0.0</td> <td>4</td> <td></td> <td></td> <td></td> </tr> <tr> <td><input type="checkbox"/> Laboratorio 204</td> <td>10.8.45.00</td> <td>1</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p style="text-align: right;">Filas por página: 100 1-3 de 3</p>		GMT DISPOSITIVOS LOCALES USUARIOS				jesus		<input type="checkbox"/> Número o Nombre	Ip	Número de Dispositivo	Acción			<input type="checkbox"/> Laboratorio 201	10.8.44.00	0				<input type="checkbox"/> Laboratorio 202	127.0.0.0	4				<input type="checkbox"/> Laboratorio 204	10.8.45.00	1			
GMT DISPOSITIVOS LOCALES USUARIOS				jesus																											
<input type="checkbox"/> Número o Nombre	Ip	Número de Dispositivo	Acción																												
<input type="checkbox"/> Laboratorio 201	10.8.44.00	0																													
<input type="checkbox"/> Laboratorio 202	127.0.0.0	4																													
<input type="checkbox"/> Laboratorio 204	10.8.45.00	1																													

Tabla 5. Historia de usuario # 4.

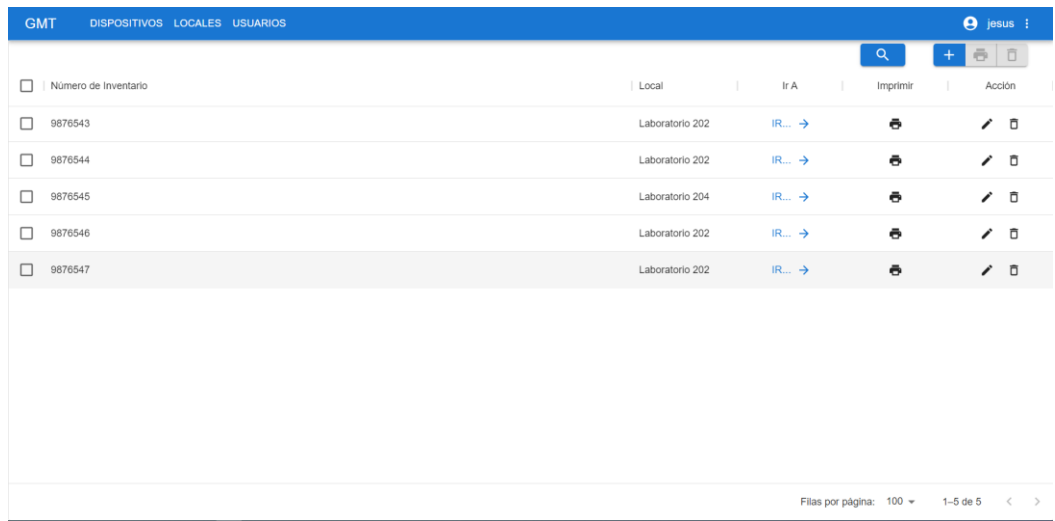
Número: 4	Nombre: Gestionar medio tecnológico		
Usuario: usuario, administrador			
Prioridad en negocio: Alta		Riesgo en desarrollo: Medio	
Puntos estimados: 2		Iteraciones asignadas: 2	
Programador asignado: Jesús Francisco Vázquez Biltre			
<p>Descripción: La aplicación web debe brindar al usuario y al administrador una vista donde se permita listar y filtrar los medios tecnológicos registrados en el sistema. Así como permitir que el administrador realice las acciones de crear, modificar y borrar dichos medios.</p>			
<p>Observaciones: Para realizar estas acciones tanto el usuario como el administrador deben de estar autenticados.</p>			
Prototipo de interfaz:			
			

Tabla 6. Historia de usuario # 5.

Número: 5	Nombre: Gestionar cambios		
Usuario: usuario, administrador			
Prioridad en negocio: Alta		Riesgo en desarrollo: Medio	
Puntos estimados: 2		Iteraciones asignadas: 2	
Programador asignado: Jesús Francisco Vázquez Biltre, Roxana Mérida Sánchez			
<p>Descripción: La aplicación web debe brindar al usuario y al administrador una vista donde se permita listar y filtrar los cambios de un medio tecnológico. Así como permitir que el administrador realice las acciones de crear, modificar, y borrar los cambios de un medio determinado.</p>			
<p>Observaciones: Para realizar estas acciones tanto el usuario como el administrador deben de estar autenticados.</p>			

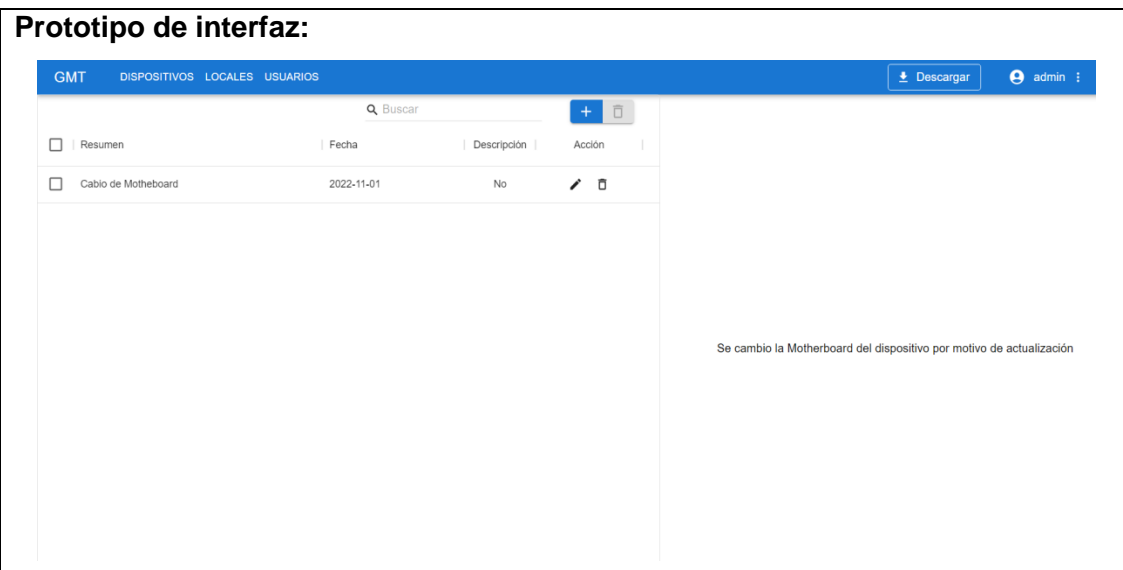


Tabla 7. Historia de usuario # 6.

Número: 6	Nombre: Detectar cambios
Usuario: sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador asignado: Jesús Francisco Vázquez Biltre	
Descripción: La aplicación web debe ser capaz de detectar cambios en el hardware, a partir de los datos recibidos de la aplicación cliente., y agregarlos a la base de datos.	
Observaciones:	

Tabla 8. Historia de usuario # 7.

Número: 7	Nombre: Imprimir ficha técnica y expediente.
Usuario: administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador asignado: Roxana Mérida Sánchez	
Descripción: La aplicación web debe permitir al administrador imprimir la ficha técnica y el expediente de uno o varios medios tecnológicos. La ficha técnica a imprimir aparte de contener los datos fundamentales del medio tecnológico constará de un código QR que identifique dicho medio.	
Observaciones: Para realizar estas acciones el administrador debe de estar autenticado.	
Prototipo de interfaz:	

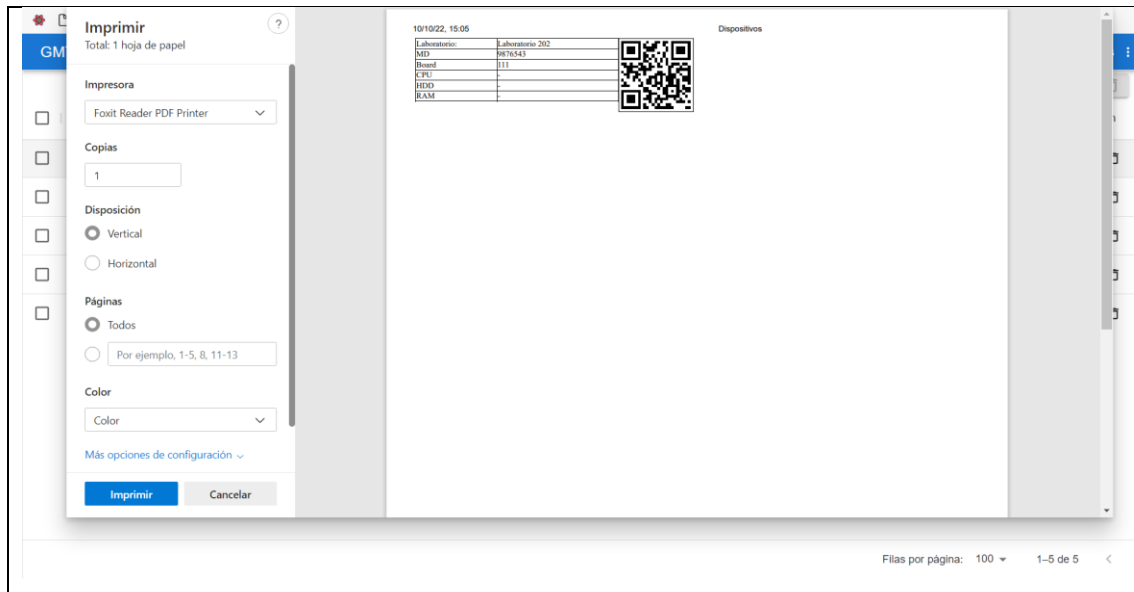


Tabla 9. Historia de usuario # 8.

Número: 8	Nombre: Exportar base de datos
Usuario: administrador	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 2
Programador asignado: Jesús Francisco Vázquez Biltre, Roxana Mérida Sánchez	
Descripción: La aplicación web debe brindar al administrador la funcionalidad de exportar una parte o la base de datos completa.	
Observaciones: Para realizar estas acciones el administrador debe de estar autenticado.	
Prototipo de interfaz:	

Historias de usuario relacionadas a la aplicación móvil.

Tabla 10. Historia de usuario # 9.

Número: 9	Nombre: Administrar medio tecnológico
Usuario: usuario	

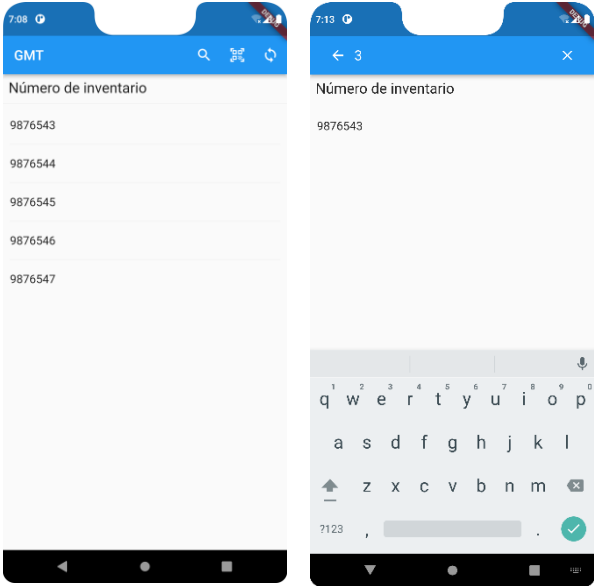
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteraciones asignadas: 3
Programador asignado: Roxana Mérida Sánchez	
Descripción: La aplicación móvil debe brindar al usuario una vista donde se muestren y filtren los medios tecnológicos registrados en la aplicación web.	
Observaciones:	
Prototipo de interfaz:	
	

Tabla 11. Historia de usuario # 10.

Número: 10	Nombre: Comprobar existencia de medio tecnológico.
Usuario: usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1.5	Iteraciones asignadas: 3
Programador asignado: Roxana Mérida Sánchez	
Descripción: La aplicación móvil debe brindar al usuario la opción escanear el código QR que se encuentra en la ficha técnica del medio tecnológico, para comprobar la existencia del mismo y la veracidad de sus datos.	
Observaciones:	
Prototipo de interfaz:	

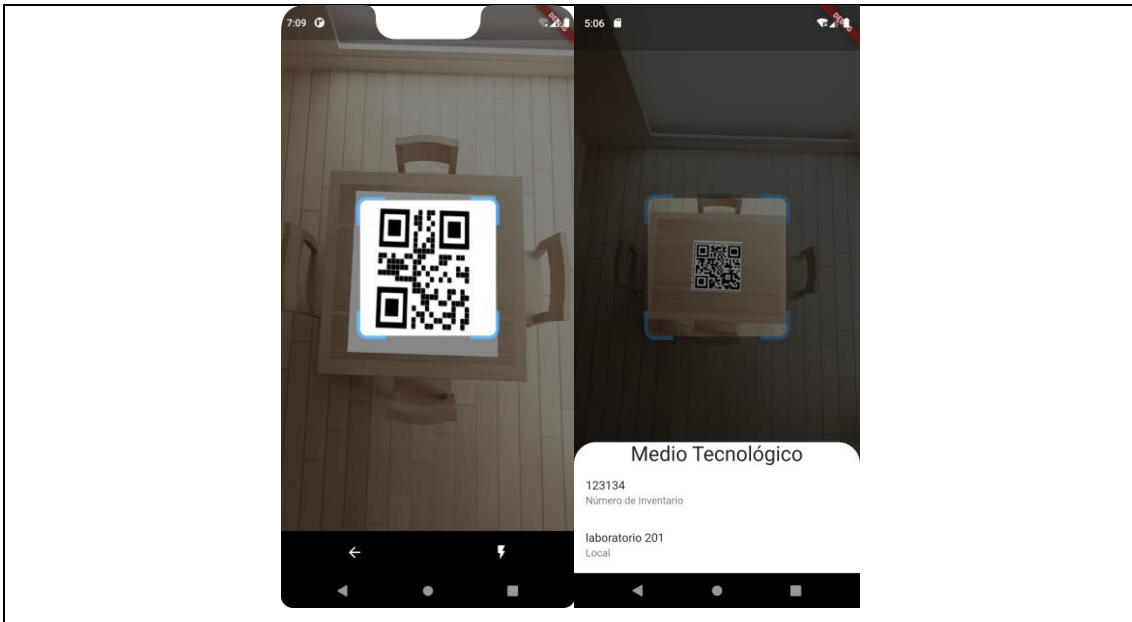


Tabla 12. Historia de usuario # 11.

Número: 11	Nombre: Cargar base de datos
Usuario: usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteraciones asignadas: 3
Programador asignado: Roxana Mérida Sánchez	
Descripción: La aplicación debe permitir al usuario cargar la base de datos desde el almacenamiento del dispositivo móvil.	
Observaciones: la base de datos debe estar copiada en el dispositivo móvil.	
Prototipo de interfaz:	

Historias de usuario relacionadas a la aplicación cliente.

Tabla 13. Historia de usuario # 12.

Número: 12	Nombre: Obtener datos del medio tecnológico.		
Usuario: sistema			
Prioridad en negocio: Alta		Riesgo en desarrollo: Medio	
Puntos estimados: 1		Iteraciones asignadas: 4	
Programador asignado: Jesús Francisco Vázquez Biltre			
<p>Descripción: La aplicación cliente debe obtener los datos del medio tecnológico en el que se encuentra ejecutándose. Los datos que se deben obtener son los siguientes:</p> <ul style="list-style-type: none"> • Número de inventario, que se obtiene desde una variable del sistema. • Número de sello que se obtiene desde una variable del sistema. • Número de serie y modelo de <i>la Motherboard</i>. • Modelo del Procesador. • Capacidad y número de serie del Disco Duro. • El tipo y la cantidad de memoria RAM • El modelo de la fuente de alimentación. • El número de serie y modelo de la Unidad Óptica. • El modelo y número de serie de la tarjeta de video. • El modelo y número de serie de la tarjeta de red. 			
Observaciones:			

Tabla 14. Historia de usuario # 13.

Número: 13	Nombre: Envío de los datos		
Usuario: sistema			
Prioridad en negocio: Alta		Riesgo en desarrollo: Medio	
Puntos estimados: 1		Iteraciones asignadas: 4	
Programador asignado: Jesús Francisco Vázquez Biltre			
<p>Descripción: La aplicación cliente debe de enviar los datos obtenidos, así como los cambios detectados, a la aplicación web para que esta los almacene y los incorpore al sistema.</p>			
Observaciones:			

2.3. Plan de iteraciones.

Una vez definidas las Historias de Usuario y realizada una previa estimación de esfuerzos, se procede a la planificación de la etapa de implementación del sistema. Este plan, tiene como finalidad, mostrar la duración de cada iteración y el orden y prioridad de las HU. La

siguiente tabla refleja la duración (en semanas) y el orden en que se deberán implementar las iteraciones:

Tabla 15. Plan de Iteraciones.

Iteración	HU a implementar	Duración total (Semanas)
1	1,2	3
2	3,4,5,6,7,8	9
3	9,10,11	4
4	12,13	2
Total		18

2.4. Plan de entrega.

En el plan de entrega que se plantea a continuación se hace una propuesta de la cantidad de historias de usuarios que se implementarán en cada iteración y el porcentaje que representan estas del total.

Tabla 16. Plan de entrega.

Iteración	1	2	3	4
Cantidad de HU	2 (15%)	6 (61%)	3 (79%)	2 (100%)
Fecha inicio	13 de junio 2022	5 de julio 2022	7 de septiembre 2022	5 de octubre 2022
Fecha fin	4 de julio 2022	6 de septiembre 2022	5 de octubre 2022	20 de octubre 2022

2.5. Arquitectura de software.

La arquitectura de software es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software.

Para utilizar la arquitectura de software se sigue un conjunto de patrones arquitectónicos, entre los cuales podemos encontrar: Blackboard, Modelo entre capas. Intérprete, Orientado a servicios y Cliente-Servidor. [37] Este último es el que se estará empleando en la propuesta de solución; consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. [38]



Ilustración 4. Arquitectura Cliente – Servidor. [39]

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores, estos últimos responden a la demanda del cliente que la produjo. [38]

Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. [38]

2.6. Patrones de diseño.

A continuación, se exponen los patrones de diseño. Los que al aplicar y utilizar correctamente podrá corregir diferentes problemas que presenta el código de una manera segura, estable y testada por cientos de programadores de todo el mundo. [40]

Dentro de estos patrones se destacan los **Patrones de Asignación de Responsabilidades (GRASP)**. Los cuales son muy empleados en la programación orientada a objeto. Estos ayudan a elegir las clases adecuadas y decidir como estas clases deben interactuar. En la siguiente lista se establece los patrones GRASP

utilizados en la implementación de la solución y ejemplos de la utilización de los mismos.

- **Experto:** es un patrón que se usa más que cualquier otro al asignar responsabilidades. Es un principio básico que suele utilizarse en el diseño orientado a objetos. Da origen a diseños donde el objeto de software realiza las operaciones que normalmente se aplican a la cosa real que representa, por lo que ofrece una analogía con el mundo real. [38] Se evidencia en las clases controladoras como DeviceController, User y en los modelos como Motherboard y Local. (Ilustración 5)
- **Creador:** guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. [38] Se evidencia en la clase DeviceService (Ilustración 6).
- **Controlador:** ofrece una guía para tomar decisiones apropiadas sobre como el sistema debe manejar los eventos de entrada externa. Definiendo que la misma clase controlador debería utilizarse con todos los eventos sistémicos de un caso de uso, de modo que se pueda conservar la información referente al estado del caso. [38] Se evidencia en la clase UserController (Ilustración 7).
- **Alta Cohesión y bajo acoplamiento:** es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Una clase de alta cohesión posee un número relativamente pequeño, con una importante funcionalidad relacionada y poco trabajo que hacer. Colabora con otros objetos para compartir el esfuerzo si la tarea es grande. [41]

Después de analizar las características de los distintos patrones. Se muestra a continuación una serie de imágenes donde se evidencia su puesta en práctica.

Experto, Alta Cohesión y Bajo acoplamiento:

```
Jesus F. Vazquez Biltre
@Entity
@Table(name = "Usuario")
class User : MyEntity() {

    Jesus F. Vazquez Biltre
    enum class Role { User, Admin }

    @Column(nullable = false)
    var name: String? = null

    @Column(nullable = false, unique = true)
    var username: String? = null

    @Column(nullable = false)
    var password: String? = null

    @Column(nullable = false)
    @Enumerated(EnumType.STRING)
    var role: Role? = null
}
```

Ilustración 5. Código ejemplo patrón Experto, Alta Cohesión.

Creador

```
@Service
class DeviceServiceImplementation : DeviceService {

    @Autowired
    final lateinit var deviceRepository: DeviceRepository

    Jesus F. Vazquez Biltre
    override fun list(): List<Device> {...}

    Jesus F. Vazquez Biltre
    override fun save(device: Device): Device {
        return deviceRepository.save(device)
    }
}
```

Ilustración 6. Código ejemplo patrón Creador.

Controlador:

```
@RestController
@CrossOrigin
@RequestMapping("/user")
class UserController {

    @Autowired
    private lateinit var userService: UserService

    @Autowired
    private lateinit var userMapper: UserMapper

    // Jesús F. Vázquez Biltre +1
    @GetMapping()
    fun list(): List<UserResponse> {...}

    // Jesús F. Vázquez Biltre
    @PostMapping
    fun save(@RequestBody saveUserRequest: SaveUserRequest): Boolean {...}

    // Jesús F. Vázquez Biltre
    @PostMapping("/role")
    fun saveRole(@RequestBody saveRoleRequest: SaveRoleRequest): String? {...}

    // Jesús F. Vázquez Biltre
    @PutMapping
    fun update(@RequestBody updateUserRequest: UpdateUserRequest): Boolean {...}

    // Jesús F. Vázquez Biltre
    @DeleteMapping
    fun delete(@RequestBody ids: List<Int>): Boolean {...}
}
```

Ilustración 7. Código ejemplo patrón Controlador.

2.7. Tarjetas Clase-Responsabilidad-Colaborador (CRC).

La utilidad de las Tarjetas CRC radica en que facilita al equipo entender el proceso de diseño en una manera fácil y concisa. Es una práctica que ayuda a romper la costumbre de los programadores a pensar en los problemas en términos de código y a comenzar a abstraer los conceptos que terminarán siendo las piezas fundamentales de nuestra aplicación o sistema. Del mismo modo resultan útiles evitando problemas que de lo contrario podrían escaparse del dominio del equipo en la etapa de diseño lógico. [42]

Tabla 17. Tarjeta CRC1 Servidor web

Clase: Device	
Responsabilidades	Clases relacionadas
Es una clase entidad que representa los datos persistentes referentes a los medios tecnológicos en la base de datos de la aplicación web.	<ul style="list-style-type: none">• Local• Motherboard• Processor• HardDisk• Ram• PowerSupply• CDReader

	<ul style="list-style-type: none"> • VideoCard • NetworkCard • Change
--	--

Tabla 18. Tarjeta CRC 2 Servidor web

Clase: DeviceService	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + +list() :encargado de devolver la lista de los medios tecnológicos que se encuentran en el sistema. + +save(device): encargado de guardar un nuevo dispositivo en el sistema. + +update(device): encargado de modificar los datos de los dispositivos. + +delete(ids): encargado de borrar dispositivos según la lista de id que pasen por parámetros. + +getById(id): encargado de devolver un dispositivo dado un Id. + +addCDReader(id,cdreader): encargado de añadir un cdreader a un dispositivo. + addMotherboard(id,motherboard):encargado de añadir una motherboard a un dispositivo. + addProcessor(id,processor): encargado de añadir un procesador a un dispositivo. + addHardDisk(id,hardDisk): encargado de añadir hardDisk a un dispositivo + addRam(id, ram): encargado de añadir una ram a un dispositivo. + addPowerSupply(id,powerSupply): encargado de añadir un powerSupply a un dispositivo. + addVideoCard(id,videoCard): encargado de añadir una tarjeta de video a un dispositivo. + addNetworkCard(id, networkCard): encargado de añadir un networkCard a un dispositivo. + getAllById(ids): según una lista de id devuelve 	<ul style="list-style-type: none"> • LocalService • Device • CDReader • Motherboard • Processor • HardDisk • Ram • PowerSupply • VideoCard • ChangeService • NetworkCard

<p>una lista de dispositivos.</p> <p>+ exist(device): devuelve verdadero si el dispositivo se encuentra en la base de datos.</p>	
--	--

Tabla 19. Tarjeta CRC 3 Servidor web

Clase: DeviceController	
Responsabilidades	Clases relacionadas
<p>+ list(): devuelve una lista de todos los dispositivos que se encuentran en la base de datos .</p> <p>+ getByld(id): según su id devuelve un dispositivo.</p> <p>+ getAllByld(ids):según la lista de id que recibe devuelve una lista de dispositivos.</p> <p>+ chekDevice(stockNumber): devuelve verdadero si se encuentra el dispositivo con el número de inventario pasado por parámetro.</p> <p>+ save(device): guarda un dispositivo sin componentes en el sistema.</p> <p>+ saveComponent(device,ip):guarda dispositivo con todos sus componentes en un local dado.</p> <p>+ update(device): actualiza los datos de los dispositivos</p> <p>+ delete(ids): elimina los dispositivos.</p>	<ul style="list-style-type: none"> • DeviceService • Device

Tabla 20. Tarjeta CRC 4 Servidor web

Clase: Local	
Responsabilidades	Clases relacionadas

Es una clase entidad que representa los datos persistentes referentes a los locales en la base de datos de la aplicación web.	<ul style="list-style-type: none"> • Device
---	--

Tabla 21. Tarjeta CRC 5 Servidor web

Clase: LocalController	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + list (): devuelve una lista de todos los locales que se encuentran en la base de datos + save(local): guarda un local sin componentes en el sistema. + update(local): actualiza los datos de los locales. + delete (ids): elimina los locales. 	<ul style="list-style-type: none"> • LocalService • Local

Tabla 22. Tarjeta CRC 6 Servidor web

Clase: LocalService	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + list (): devuelve una lista de todos los locales que se encuentran en la base de datos + save(local): guarda un local sin componentes en el sistema. + update (local): actualiza los datos de los locales. + delete (ids): elimina los locales. + getByIP(ip): encargado de obtener un local dado un IP 	<ul style="list-style-type: none"> • LocalController • Local

Tabla 23. Tarjeta CRC 7 Servidor web

Clase: User	
Responsabilidades	Clases relacionadas
Es la clase entidad que se encarga de representar los usuarios en la base de	

datos	
-------	--

Tabla 24. Tarjeta CRC 8 Servidor web

Clase: UserService	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + save(user): encargado de guardar en la base de datos el usuario que se le pasa por parámetro. + getByUsername(username): dado un nombre de usuario recibe un usuario. + exist(user): comprueba si existe o no el usuario en la base de datos. + list(): lista todos los usuarios que están en la base de datos. + saveRole(id,role): agrega o modifica el rol que tenga el usuario. + delete(ids): dada la lista de id borra todos los usuarios. + upppdate(user): actualiza los datos de los usuarios. <p>Es el encargado de toda la manipulación de datos q realiza el servidor.</p>	<ul style="list-style-type: none"> • User • PasswordEncoder

Tabla 25. Tarjeta CRC 8 Servidor web

Clase: UserController	
Responsabilidades	Clases relacionadas
<p>Es el encargado de establecer la conexión de la vista con el servidor web.</p> <ul style="list-style-type: none"> + list(): lista todos los usuarios que están en la base de datos. + save(user): encargado de guardar en la base de datos el usuario que se le pasan por 	<p>UserService</p> <p>User</p>

parámetro. + delete(ids): dada la lista de id borra todos los usuarios. + update(user): actualiza los datos de los usuarios.	
--	--

Tabla 26. Tarjeta CRC 9 Servidor web

Clase: ExportController	
Responsabilidades	Clases relacionadas
+ exportByLocalIds(ids): encargado de devolver dado una lista de id de los locales un ExportModel. + exportByDeviceIds(ids): encargado de devolver dado una lista de id de los dispositivos un ExportModel.	<ul style="list-style-type: none"> • ExportService • ExportModel

Tabla 27. Tarjeta CRC 10 Servidor web

Clase: ExportService	
Responsabilidades	Clases relacionadas
+ byLocalIds(ids): encargado de devolver dado una lista de id de los locales un ExportModel. + byDeviceIds(ids): encargado de devolver dado una lista de id de los dispositivos un ExportModel.	<ul style="list-style-type: none"> • LocalService • ExportModel • DeviceService

Tabla 28. Tarjeta CRC 11 Servidor web

Clase: ExportModel	
Responsabilidades	Clases relacionadas
Modelo que permite exportar una lista de locales y dispositivos.	<ul style="list-style-type: none"> • Local • Device

Tabla 29. Tarjeta CRC 12 Servidor web

Clase: Motherboard	
Responsabilidades	Clases relacionadas

Es una clase entidad que representa los datos persistentes referentes a las motherboards de los medios tecnológicos en la base de datos.	<ul style="list-style-type: none"> • Device
--	--

Tabla 30. Tarjeta CRC 13 Servidor web

Clase: MotherboardService	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + update(motherboard): actualiza los datos de una motehrboard en la base de datos. + save(idDevice,motherboard): encargado de guardar los datos de la motherboard del dispositivo en la base de datos. 	<ul style="list-style-type: none"> • Motherboard • DeviceService • ChangeService

Tabla 31. Tarjeta CRC 14 Servidor web

Clase: Change	
Responsabilidades	Clases relacionadas
Es la clase entidad que se encarga de representar los cambios en la base de datos	<ul style="list-style-type: none"> • Device

Tabla 32. Tarjeta CRC 15 Servidor web

Clase: ChangeController	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + list (): devuelve una lista de todos los cambios que se encuentran en la base de datos + save(local): guarda un cambio en el sistema. + update(local): actualiza los datos del cambio. + delete (ids): elimina los cambios. 	<ul style="list-style-type: none"> • Change • ChangeService

Tabla 33. Tarjeta CRC 15 Servidor web

Clase: ChangeService	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + list (): devuelve una lista de todos los cambios que se encuentran en la base de datos + save(local): guarda un cambio en el sistema. + update(local): actualiza los datos del cambio. + delete (ids): elimina los cambios. 	<ul style="list-style-type: none"> • Change • Device

Tabla 34. Tarjeta CRC 16 Servidor web

Clase: MotherboardController	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + save(motherboard): encargado de guardar los datos de la motherboard del dispositivo en la base de datos. + update(motherboard): actualiza los datos de una motehrboard en la base de datos 	<ul style="list-style-type: none"> • Motherboard • MotherboardService

Las clases CDReader, HardDisk, NetworkCard, PowerSupply, Processor, Ram y VideoCard, presentan la misma estructura que la clase Motherboard. Por otra parte, las clases CDReaderService, HardDiskService, NetworkCardService, PowerSupplyService, ProcessorService, RamService y VideoCardService, presentan una estructura similar a la clase MotherboardService. Y las clases CDReaderController, HardDiskController, NetworkCardController, PowerSupplyController, ProcessorController, RamController y VideoCardController; están estructuradas de la misma forma que la clase MotherboardController.

Tabla 35. Tarjeta CRC 17 Aplicación móvil

Clase: DeviceEntity	
Responsabilidades	Clases relacionadas
Es una clase entidad la que representa los datos persistentes referentes a los medios tecnológicos en la base de datos de la aplicación móvil.	

Tabla 36. Tarjeta CRC 18 Aplicación móvil

Clase: DeviceApi	
Responsabilidades	Clases relacionadas
getAll():devuelve la lista de dispositivos que se encuentran en la aplicación web.	<ul style="list-style-type: none"> • DeviceEntity

Tabla 37. Tarjeta CRC 19 Aplicación móvil

Clase: DeviceStore	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + init(): encargado de obtener todos los devices de la base de datos de la aplicación móvil. + sync():encargado de sincronizar la base de datos de la aplicación móvil con la de la aplicación web. + filter(stockNumber): encargado de filtrar los dispositivos por los números de inventario. + exist(device): chequea si existe un dispositivo en la base de datos. 	<ul style="list-style-type: none"> • DeviceApi • DeviceDao • DeviceEntity

Tabla 38. Tarjeta CRC 20 Aplicación móvil

Clase: DeviceDao	
Responsabilidades	Clases relacionadas
<ul style="list-style-type: none"> + save(device): encargado de guardar en base de datos el dispositivo que se le pasa por parámetro. + saveAll(devices): guarda en base 	<ul style="list-style-type: none"> • DeviceEntity

<p>de datos la lista de dispositivos que se le pasan por parámetros.</p> <p>+ findAll(): encargado de listar todos los dispositivos que estén en la base de datos de la aplicación móvil.</p> <p>+ remove(device): borra de la base de datos el dispositivo que se le pasa por parámetro.</p> <p>+ removeAll():borra todos los dispositivos que estén en la base de datos de la aplicación móvil.</p>	
---	--

Tabla 39. Tarjeta CRC 21 Aplicación cliente

Clase: Service	
Responsabilidades	Clases relacionadas
<p>+ send(): encargado de enviar los datos a la aplicación web .</p> <p>+ start(): obtiene los datos del dispositivo y usa el método send() para enviar estos datos y los imprime en consola mediante el método print().</p> <p>+ print(): encargado de imprimir los datos en consola.</p>	<ul style="list-style-type: none"> • Device

Las clases Device, Motherboard, CDReader, HardDisk, NetworkCard, PowerSupply, Processor, Ram y VideoCard, tienen la misma estructura que las del servidor web.

2.8. Modelo de datos.

A partir de los datos obtenidos a lo largo de la investigación se generó el siguiente diagrama de entidad-relación el cual refleja cómo está estructurada la base de datos de la solución planteada.

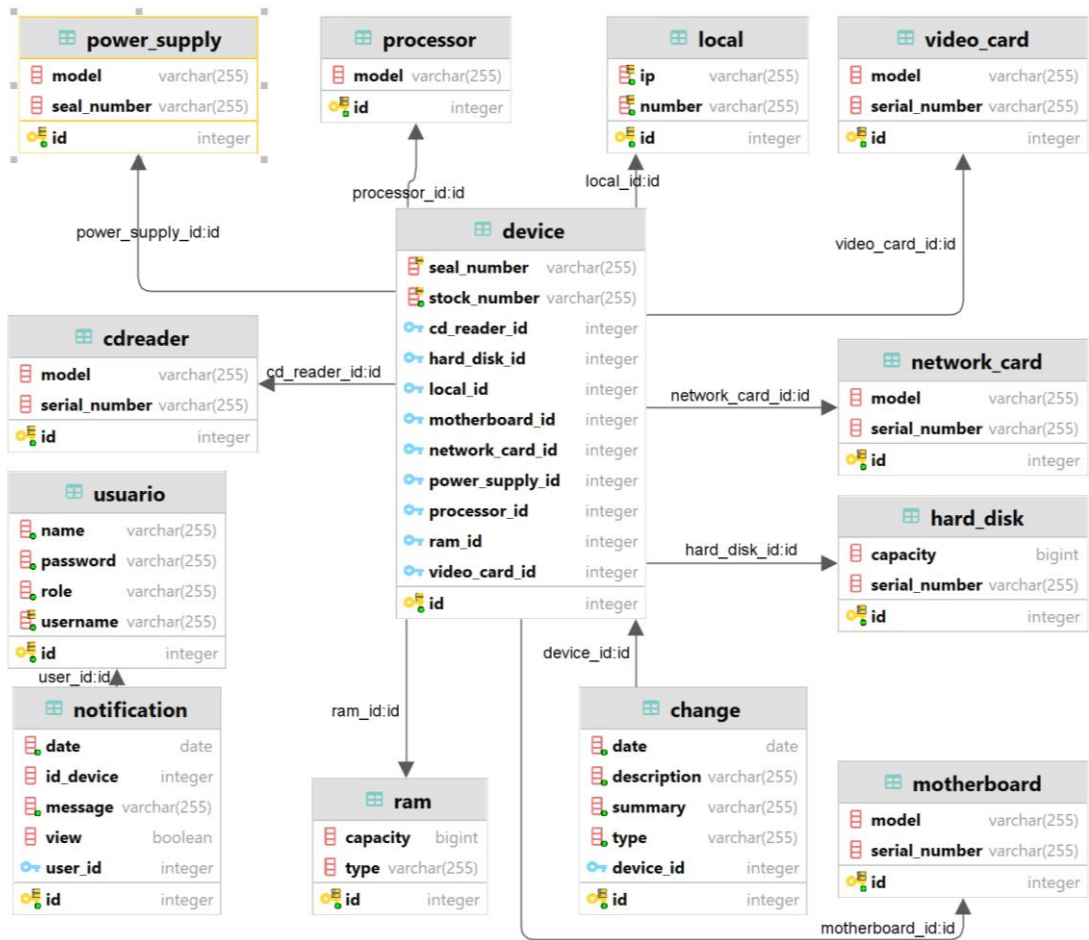


Ilustración 8. Diagrama entidad-relación

2.9. Conclusiones del capítulo.

Después de realizar los análisis correspondientes al presente capítulo se ha podido llegar a las siguientes conclusiones:

- Con la captura de los requisitos funcionales mediante la técnica de historias de usuario, se logró establecer las principales funcionalidades del sistema.
- Con el establecimiento de la arquitectura propuesta se alcanza a crear una aplicación más robusta y menos propensa a daños.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS.

El presente capítulo abarca las etapas de implementación y pruebas. Para ello, se detallan los pasos que se realizaron en el proceso de implementación, donde se exponen las tareas generadas por cada HU, para obtener al final de cada iteración, un producto que cumpla con los requisitos funcionales. Se exponen las diferentes pruebas realizadas para un producto final que satisfaga las expectativas del cliente.

3.1. Estándares de codificación.

Entendemos como estándar de código a un conjunto de convenciones establecidas de ante mano (denominaciones, formatos, etc.) para la escritura de código. [43]

Comentario de código:

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.

Máxima longitud de las líneas:

- Se limitarán todas las líneas a un máximo de 150 caracteres.
- Dentro de paréntesis, corchetes o llaves se puede utilizar la continuación implícita para cortar las líneas largas.

Líneas en blanco:

- Las definiciones de métodos dentro de una clase deben separarse por una línea en blanco.
- Se puede utilizar líneas en blanco escasamente para separar secciones lógicas.

Importaciones:

- Las importaciones deben estar en líneas separadas.
- Las importaciones siempre deben colocarse al comienzo del archivo.

3.2. Tareas de ingeniería.

Las tareas de ingeniería tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente. [44] A continuación, se describen las

tareas de ingeniería correspondientes a las HU de la 1ra iteración, las demás se ubican en el Anexos 2. En estas tarjetas los puntos estimados equivalen a horas.

Iteración 1

Tabla 40. Tarea de Ingeniería # 1

Tarea	
Número de la tarea: 1	Número de Historia de Usuario: 1
Nombre de la Tarea: Implementar autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad que mediante la entrada del usuario y contraseña permita autenticarse en la aplicación web.	

Tabla 41. Tarea de Ingeniería # 1

Tarea	
Número de la tarea: 2	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar listar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita visualizar nombre, usuario y rol de cada usuario registrado en el sistema.	

Tabla 42. Tarea de Ingeniería # 2

Tarea	
Número de la tarea: 3	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar crear usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que, mediante la entrada del nombre, usuario y rol, permita registrar un usuario en la aplicación.	

Tabla 43. Tarea de Ingeniería # 3

Tarea	
Número de la tarea: 4	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar modificar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	

Descripción: Se implementa una funcionalidad en la aplicación web que permita modificar nombre, usuario o rol de un usuario ya registrado.

Tabla 44. Tarea de Ingeniería # 5

Tarea	
Número de la tarea: 5	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar eliminar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita eliminar uno o varios usuarios ya registrados.	

Tabla 45. Tarea de Ingeniería # 6

Tarea	
Número de la tarea: 6	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar filtrar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita filtrar los usuarios ya registrados mediante el nombre, usuario o rol.	

3.3. Pruebas.

Uno de los pilares de la Programación Extrema (XP a partir de ahora) es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. Generalmente las pruebas del sistema son desarrolladas por los programadores para verificar que su sistema se comporta de la manera esperada, por lo que podrían encajar dentro de la definición de pruebas unitarias que propone XP. Sin embargo, las pruebas del sistema tienen como objetivo verificar que el sistema cumple los requisitos establecidos por el usuario por lo que también pueden encajar dentro de la categoría de pruebas de aceptación. [45]Con el objetivo de comprobar que los sistemas desarrollados funcionan de acuerdo a las

especificaciones descritas por el cliente, se realizaron diferentes pruebas teniendo en cuenta las características de los módulos.

3.3.1. Pruebas unitarias.

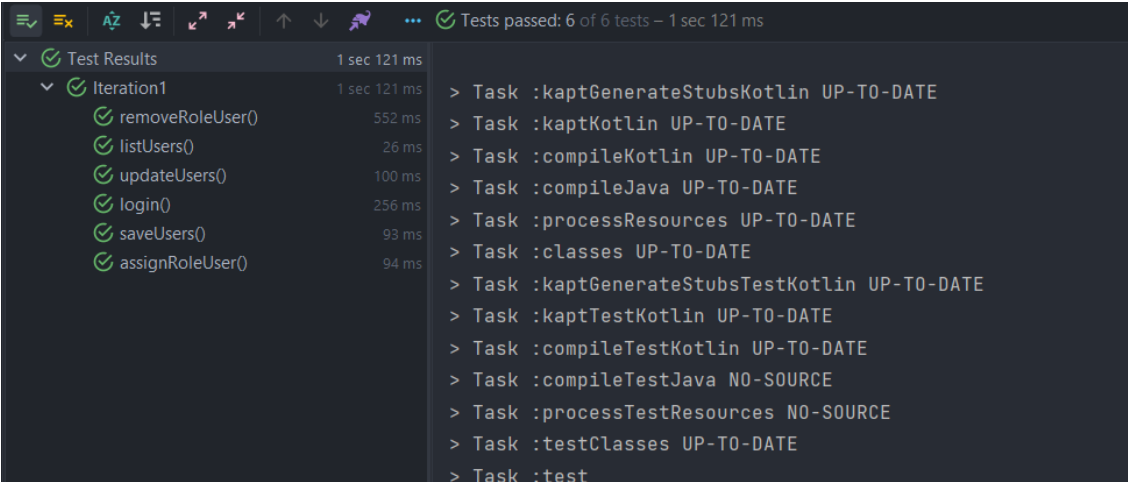
Las pruebas unitarias o *unittesting* son una forma de comprobar que un fragmento de código funciona correctamente. Es un procedimiento más de los que se llevan a cabo dentro de una metodología ágil de trabajo. Estas consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños *tests* que validan el comportamiento de un objeto y la lógica. [46]

Para la aplicación de las pruebas unitarias a la solución se empleó el framework JUnit el cual permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente. [47]

A continuación, se muestran los resultados de aplicar las pruebas al concluir cada iteración.

Iteración 1:

En esta iteración se realizaron 6 pruebas unitarias, las cuales fueron exitosas. En la ilustración siguiente se muestran los resultados de estas pruebas.



```
Tests passed: 6 of 6 tests - 1 sec 121 ms
Test Results 1 sec 121 ms
  Iteration1 1 sec 121 ms
    removeRoleUser() 552 ms
    listUsers() 26 ms
    updateUsers() 100 ms
    login() 256 ms
    saveUsers() 93 ms
    assignRoleUser() 94 ms
  > Task :kaptGenerateStubsKotlin UP-TO-DATE
  > Task :kaptKotlin UP-TO-DATE
  > Task :compileKotlin UP-TO-DATE
  > Task :compileJava UP-TO-DATE
  > Task :processResources UP-TO-DATE
  > Task :classes UP-TO-DATE
  > Task :kaptGenerateStubsTestKotlin UP-TO-DATE
  > Task :kaptTestKotlin UP-TO-DATE
  > Task :compileTestKotlin UP-TO-DATE
  > Task :compileTestJava NO-SOURCE
  > Task :processTestResources NO-SOURCE
  > Task :testClasses UP-TO-DATE
  > Task :test
```

Ilustración 9. Aplicación de la prueba unitaria Iteración 1.

Iteración 2:

En esta iteración se realizaron 7 pruebas unitarias, una de ellas tuvo dos fallos los cuales fueron solucionados satisfactoriamente. En la ilustración siguiente se muestran los resultados de la aplicación de estas pruebas

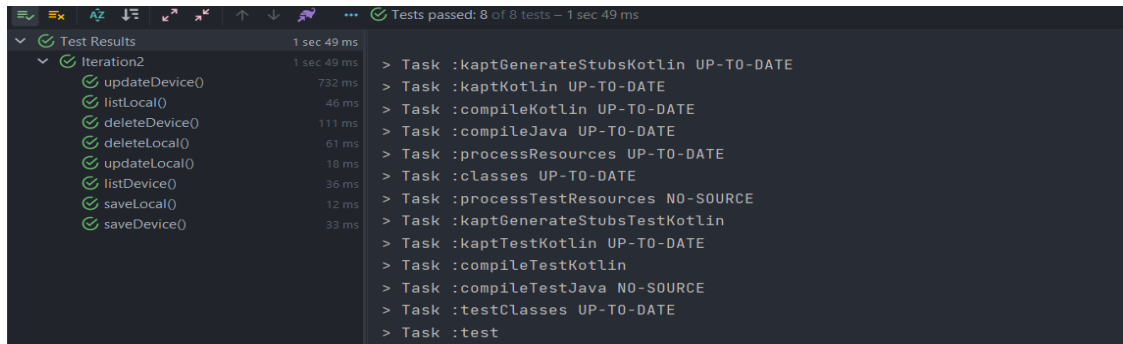


Ilustración 10. Aplicación de la prueba unitaria Iteración 2.

Iteración 3:

En esta iteración se realizaron 3 pruebas unitarias, las cuales no se encontraron fallos. A continuación, se muestran los resultados obtenidos.

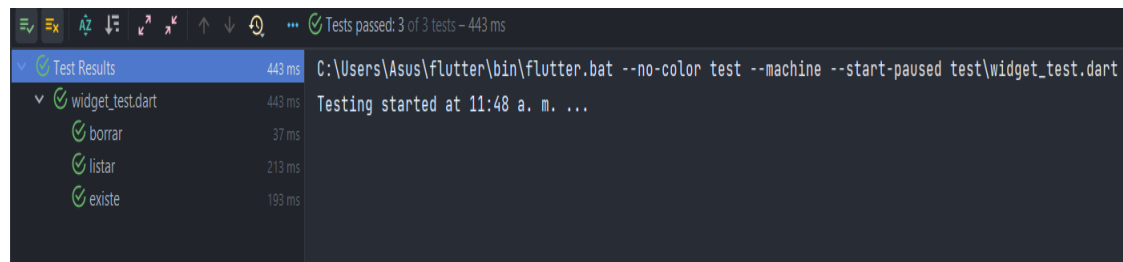


Ilustración 11. Aplicación de la prueba unitaria Iteración 3.

Iteración 4:

En esta iteración se realizaron 3 pruebas unitarias, las cuales no se encontraron fallos. A continuación, se muestran los resultados obtenidos.

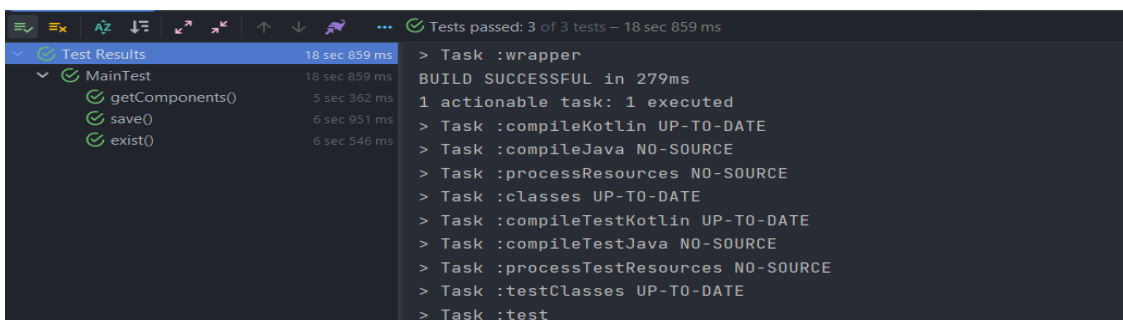


Ilustración 12. Aplicación de la prueba unitaria Iteración 4.

3.3.2. Pruebas de aceptación.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario. Estas pruebas van dirigidas a comprobar que el sistema cumple los requisitos de funcionamiento esperado, recogidos en el catálogo de requisitos y en los criterios de aceptación del sistema de información, y conseguir así la aceptación final del sistema por parte del usuario. [48]

A continuación, se muestran las pruebas de aceptación realizadas a cada iteración.

Iteración 1:

Tabla 46. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba la funcionalidad de autenticar usuario	
Condiciones de ejecución: No presenta	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe rellenar los campos del formulario con los valores correspondientes.2. El usuario presionará el botón de entrar.3. Como respuesta el sistema mostrará la página de gestionar locales.	
Resultado: Satisfactorio	

Tabla 47. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: 2
Nombre: Listar usuario	
Descripción: Prueba la funcionalidad de listar usuario	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar autenticado.• El usuario tiene que tener el rol de administrador.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe navegar a la página para listar usuario utilizando la barra de navegación.2. Como respuesta el sistema mostrará la página de gestionar usuario la cual mostrará una lista de los usuarios que se encuentra en el sistema.	
Resultado: Satisfactorio	

Tabla 48. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU2_P3	Historia de usuario: 2
Nombre: Crear usuario	
Descripción: Prueba la funcionalidad de crear usuario	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar usuarios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón para crear usuario, el cual activará un panel flotante. 2. El usuario completa los datos del formulario y presionará el botón de aceptar. 3. Como respuesta el sistema se mostrará los datos del nuevo usuario en la tabla. 	
Resultado: Satisfactorio	

Tabla 49. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU2_P4	Historia de usuario: 2
Nombre: Modificar usuario	
Descripción: Prueba la funcionalidad de asignarle roles a los usuarios	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar usuarios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón para modificar el cual activará un panel flotante con los datos a modificar. 2. El usuario introduce nuevos datos y presiona el botón de aceptar. 3. Como respuesta el sistema mostrará los nuevos datos del usuario en la tabla. 	
Resultado: Satisfactorio	

Tabla 50. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU2_P5	Historia de usuario: 2

Nombre: Eliminar usuario
Descripción: Prueba la funcionalidad de eliminar usuarios
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar usuarios.
Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario debe presionar el botón para eliminar el cual mostrará el mensaje “Borrar usuario”. 2. El usuario presiona el botón de aceptar. 3. Como respuesta el sistema eliminará el usuario de la tabla.
Resultado: Satisfactorio

Tabla 51. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU2_P6	Historia de usuario: 2
Nombre: Filtrar usuario	
Descripción: Prueba la funcionalidad de filtrar usuarios	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar usuarios. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario debe presionar el botón buscar. 2. Como respuesta el sistema muestra panel flotante con las opciones de filtrado. 3. El usuario introduce los datos a filtrar. 4. El sistema muestra una lista con los resultados después de aplicado el filtro. 	
Resultado: Satisfactorio	

Iteración 2:

Tabla 52. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU3_P7	Historia de usuario: 3
Nombre: Crear local	
Descripción: Prueba la funcionalidad de crear local.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. 	

<ul style="list-style-type: none"> • El usuario debe estar en la página para gestionar locales.
Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Añadir. 2. Como respuesta el sistema muestra panel flotante con los campos a llenar. 3. El usuario introduce los datos y presiona botón Añadir. 4. El sistema muestra una tabla con los nuevos datos insertados.
Resultado: Satisfactorio

Tabla 53. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU3_P8	Historia de usuario: 3
Nombre: Modificar local	
Descripción: Prueba la funcionalidad de modificar local.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar locales. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Modificar. 2. Como respuesta el sistema muestra panel flotante con los campos a modificar. 3. El usuario modifica los datos y presiona botón Modificar. 4. El sistema muestra una tabla con los nuevos datos insertados. 	
Resultado: Satisfactorio	

Tabla 54. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU3_P9	Historia de usuario: 3
Nombre: Eliminar local	
Descripción: Prueba la funcionalidad de eliminar local.	
Condiciones de ejecución: <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar locales. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Eliminar. 2. Como respuesta el sistema muestra el mensaje flotante "Borrar local". 3. El usuario presiona botón Borrar. 4. El sistema como respuesta elimina local de la tabla. 	
Resultado: Satisfactorio	

Tabla 55. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU3_P10	Historia de usuario: 3
Nombre: Listar local	
Descripción: Prueba la funcionalidad de listar local	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página gestionar locales. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe navegar a la página para listar locales utilizando la barra de navegación. 2. Como respuesta el sistema mostrará la página de gestionar locales la cual mostrará una lista de los locales que se encuentra en el sistema. 	
Resultado: Satisfactorio	

Tabla 56. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU3_P11	Historia de usuario: 3
Nombre: Filtrar locales	
Descripción: Prueba la funcionalidad de filtrar locales	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página para gestionar locales. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón buscar. 2. Como respuesta el sistema muestra panel flotante con las opciones de filtrado. 3. El usuario introduce los datos a filtrar. 4. El sistema muestra una lista con los resultados después de aplicado el filtro. 	
Resultado: Satisfactorio	

Tabla 57. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU4_P12	Historia de usuario: 4
Nombre: Crear medios tecnológicos	
Descripción: Prueba la funcionalidad de crear medios tecnológicos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar medios tecnológicos. 	

<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Añadir. 2. Como respuesta el sistema muestra panel flotante con los campos a llenar. 3. El usuario introduce los datos y presiona botón Añadir. 4. El sistema muestra una tabla con los nuevos datos insertados.
<p>Resultado: Satisfactorio</p>

Tabla 58. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU4_P13	Historia de usuario: 4
Nombre: Modificar medios tecnológicos	
Descripción: Prueba la funcionalidad de modificar medios tecnológicos.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar medios tecnológicos. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Modificar. 2. Como respuesta el sistema muestra panel flotante con los campos a modificar. 3. El usuario modifica los datos y presiona botón Modificar. 4. El sistema muestra una tabla con los nuevos datos insertados. 	
Resultado: Satisfactorio	

Tabla 59. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU4_P14	Historia de usuario: 4
Nombre: Eliminar medios tecnológicos	
Descripción: Prueba la funcionalidad de eliminar medios tecnológicos.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar medios tecnológicos. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario debe presionar el botón Eliminar. 2. Como respuesta el sistema muestra el mensaje flotante “Borrar medio tecnológico”. 3. El usuario presiona botón Borrar. 4. El sistema como respuesta elimina medio tecnológico de la tabla. 	
Resultado: Satisfactorio	

Tabla 60. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU4_P15	Historia de usuario: 4
Nombre: Listar medios tecnológicos	
Descripción: Prueba la funcionalidad de listar medios tecnológicos	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página gestionar medios tecnológicos. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe navegar a la página para listar medios tecnológicos utilizando la barra de navegación. 2. Como respuesta el sistema mostrará la página de gestionar medios tecnológicos la cual mostrará una lista de los medios tecnológicos que se encuentra en el sistema. 	
Resultado: Satisfactorio	

Tabla 61. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: HU4_P16	Historia de usuario: 4
Nombre: Filtrar medios tecnológicos	
Descripción: Prueba la funcionalidad de filtrar medios tecnológicos	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página para gestionar medios tecnológicos. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón buscar. 2. Como respuesta el sistema muestra panel flotante con las opciones de filtrado. 3. El usuario introduce los datos a filtrar. 4. El sistema muestra una lista con los resultados después de aplicado el filtro. 	
Resultado: Satisfactorio	

Tabla 62. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: HU5_P17	Historia de usuario: 5
Nombre: Crear cambios	
Descripción: Prueba la funcionalidad de crear cambios.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar cambios. 	

Pasos de ejecución:
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón Añadir. 2. Como respuesta el sistema muestra panel flotante con los campos a llenar. 3. El usuario introduce los datos y presiona botón Añadir. 4. El sistema muestra una tabla con los nuevos datos insertados.
Resultado: Satisfactorio

Tabla 63. Prueba de aceptación # 18

Caso de prueba de aceptación	
Código: HU5_P18	Historia de usuario: 5
Nombre: Modificar cambios	
Descripción: Prueba la funcionalidad de modificar cambios.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar cambios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón Modificar. 2. Como respuesta el sistema muestra panel flotante con los campos a modificar. 3. El usuario modifica los datos y presiona botón Modificar. 4. El sistema muestra una tabla con los nuevos datos insertados. 	
Resultado: Satisfactorio	

Tabla 64. Prueba de aceptación # 19

Caso de prueba de aceptación	
Código: HU5_P19	Historia de usuario: 5
Nombre: Eliminar cambios	
Descripción: Prueba la funcionalidad de eliminar cambios.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario tiene que tener el rol de administrador. • El usuario debe estar en la página para gestionar cambios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón Eliminar. 2. Como respuesta el sistema muestra el mensaje flotante "Borrar cambio". 3. El usuario presiona botón Borrar. 4. El sistema como respuesta elimina el cambio de la tabla. 	
Resultado: Satisfactorio	

Tabla 65. Prueba de aceptación # 20

Caso de prueba de aceptación	
Código: HU5_P20	Historia de usuario: 5
Nombre: Listar cambios	
Descripción: Prueba la funcionalidad de listar cambios	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página gestionar cambios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe navegar a la página para listar los cambios. 2. Como respuesta el sistema mostrará la página de gestionar cambios la cual mostrará una lista de los cambios que se encontraron en el sistema. 	
Resultado: Satisfactorio	

Tabla 66. Prueba de aceptación # 21

Caso de prueba de aceptación	
Código: HU5_P21	Historia de usuario: 5
Nombre: Filtrar cambios	
Descripción: Prueba la funcionalidad de filtrar cambios	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe estar en la página para gestionar cambios. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario debe presionar el botón buscar. 2. Como respuesta el sistema muestra panel flotante con las opciones de filtrado. 3. El usuario introduce los datos a filtrar. 4. El sistema muestra una lista con los resultados después de aplicado el filtro. 	
Resultado: Satisfactorio	

Tabla 67. Prueba de aceptación # 22

Caso de prueba de aceptación	
Código: HU7_P17	Historia de usuario: 7
Nombre: Imprimir ficha técnica	
Descripción: Prueba la funcionalidad de imprimir ficha técnica	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe tener el rol de administrador. • El usuario debe estar en la página gestionar medios tecnológicos. 	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. El usuario presiona botón imprimir. 2. Como respuesta el sistema muestra un panel flotante con las opciones de impresión y la vista de las fichas técnicas. 3. El usuario selecciona botón Imprimir. 4. El sistema envía los datos a la impresora.
Resultado: Satisfactorio

Tabla 68. Prueba de aceptación # 23

Caso de prueba de aceptación	
Código: HU7_P23	Historia de usuario: 7
Nombre: Imprimir expediente	
Descripción: Prueba la funcionalidad de imprimir expediente	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe tener el rol de administrador. • El usuario debe estar en la página gestionar medios tecnológicos. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario presiona botón imprimir. 2. Como respuesta el sistema muestra un panel flotante con las opciones de impresión y la vista de los expedientes. 3. El usuario selecciona botón Imprimir. 4. El sistema envía los datos a la impresora. 	
Resultado: Satisfactorio	

Tabla 69. Prueba de aceptación # 24

Caso de prueba de aceptación	
Código: HU8_P24	Historia de usuario: 8
Nombre: Exportar base de datos	
Descripción: Prueba la funcionalidad exportar la base de datos.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • El usuario debe estar autenticado. • El usuario debe tener el rol de administrador. • El usuario debe estar en la página gestionar medios tecnológicos o locales. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. El usuario presiona botón Guardar BD. 2. Como respuesta el sistema debe descargar la base de datos como un archivo llamado <i>apk.db</i>. 	
Resultado: Satisfactorio	

Iteración 3:

Tabla 70. Prueba de aceptación # 25

Caso de prueba de aceptación	
Código: HU9_P25	Historia de usuario: 9
Nombre: Listar medios tecnológicos	
Descripción: Prueba la funcionalidad de listar medios tecnológicos	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar utilizando la aplicación móvil.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe acceder a la aplicación móvil desde el cajón de aplicaciones de su dispositivo.2. La aplicación muestra listado de medios tecnológicos.	
Resultado: Satisfactorio	

Tabla 71. Prueba de aceptación # 26

Caso de prueba de aceptación	
Código: HU9_P26	Historia de usuario: 9
Nombre: Filtrar medios tecnológicos	
Descripción: Prueba la funcionalidad de filtrar medios tecnológicos	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar utilizando la aplicación móvil.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe presionar botón buscar.2. El sistema muestra barra de búsqueda.3. El usuario introduce los datos4. El sistema muestra resultado de la búsqueda.	
Resultado: Satisfactorio	

Tabla 72. Prueba de aceptación # 27

Caso de prueba de aceptación	
Código: HU10_P27	Historia de usuario: 10
Nombre: Comprobar existencia de medios tecnológicos	
Descripción: Prueba la funcionalidad de comprobar existencia de medios tecnológicos	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar utilizando la aplicación móvil.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe presionar botón escanear QR.2. El sistema muestra la cámara para escanear.3. El usuario escanea el código QR que se encuentra en la ficha técnica.4. El sistema muestra si está registrado o no el medio tecnológico.	

Resultado: Satisfactorio

Tabla 73. Prueba de aceptación # 28

Caso de prueba de aceptación	
Código: HU11_P28	Historia de usuario: 11
Nombre: Cargar base de datos.	
Descripción: Prueba la funcionalidad de cargar la base de datos.	
Condiciones de ejecución: <ul style="list-style-type: none">• El usuario debe estar utilizando la aplicación móvil.	
Pasos de ejecución: <ol style="list-style-type: none">1. El usuario debe presionar botón cargar.2. El sistema debe abrir el panel seleccionar archivo.3. El usuario selecciona archivo <i>apk.db</i>.4. La aplicación debe mostrar los datos contenidos en el archivo.	
Resultado: Satisfactorio	

Iteración 4:

Tabla 74. Prueba de aceptación # 29

Caso de prueba de aceptación	
Código: HU12_P29	Historia de usuario: 12
Nombre: Obtener datos de medios tecnológicos	
Descripción: Prueba la funcionalidad de obtener datos de medios tecnológicos	
Condiciones de ejecución: <ul style="list-style-type: none">• La aplicación cliente debe estar ejecutándose en un medio tecnológico.	
Pasos de ejecución: <ol style="list-style-type: none">1. La aplicación debe obtener datos del<ul style="list-style-type: none">• Número de inventario• Motherboard• Procesador• Disco duro• RAM• Fuente de alimentación• Unidad óptica• Tarjeta de video• Tarjeta de red2. La aplicación debe mostrar estos datos por consola.	
Resultado: Satisfactorio	

Tabla 75. Prueba de aceptación # 30

Caso de prueba de aceptación	
Código: HU13_P30	Historia de usuario: 13

Nombre: Enviar datos
Descripción: Prueba la funcionalidad de enviar datos del medio tecnológico.
Condiciones de ejecución: <ul style="list-style-type: none"> • Se debe estar ejecutando la aplicación cliente.
Pasos de ejecución: <ol style="list-style-type: none"> 1. La aplicación debe comprobar si los datos existen en el sistema, en caso contrario los envía. 2. Debe mostrar mensaje de confirmación que se enviaron los datos correctamente por consola.
Resultado: Satisfactorio

3.4. Resultados de las pruebas.

A las HU correspondientes a cada iteración implementada se le realizaron varias pruebas unitarias y de aceptación donde algunas de las dificultades detectadas fueron solucionadas en un corto plazo, mientras que otras no tienen un nivel alto de criticidad para el desarrollo del sistema, cumpliéndose correctamente todos los requisitos solicitados por el cliente. En la siguiente ilustración se muestran los resultados de las pruebas, donde por cada iteración implementada se muestra la cantidad de pruebas que fueron satisfactorias o no. Se pone en evidencia mediante el color azul la cantidad de pruebas que se desarrollaron de manera satisfactoria, mediante la columna naranja la cantidad de pruebas no satisfactorias a la hora de su aplicación y la gris muestra la cantidad de errores solucionados.

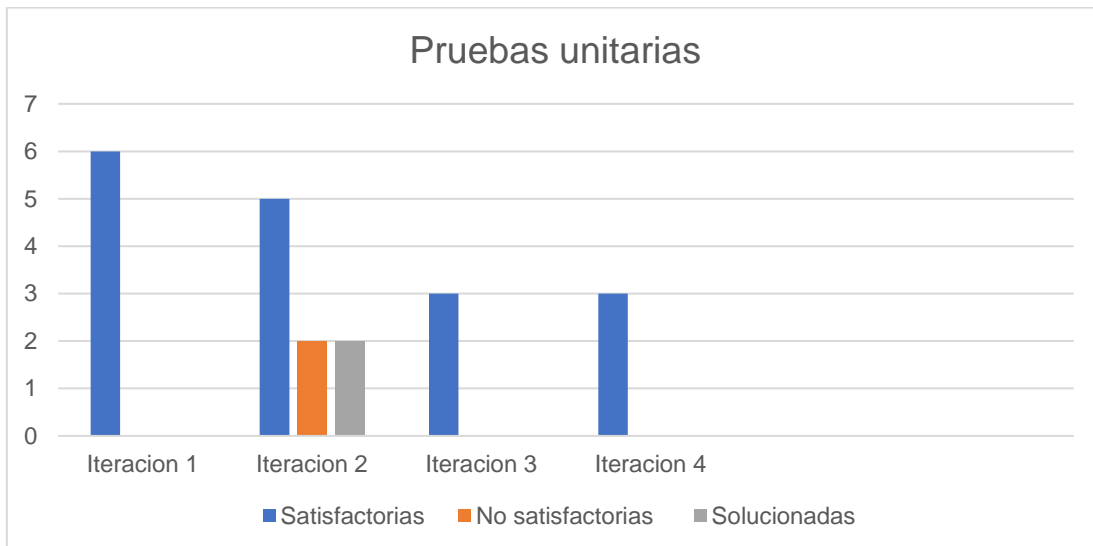


Gráfico 1. Pruebas unitarias.

El anterior gráfico representa el resultado de la aplicación de las pruebas unitarias. El mismo revela que se realizaron un total de 19 pruebas unitarias, de las cuales 17

resultaron satisfactorias, mientras que dos presentaron algunos errores, que finalmente fueron solucionados.

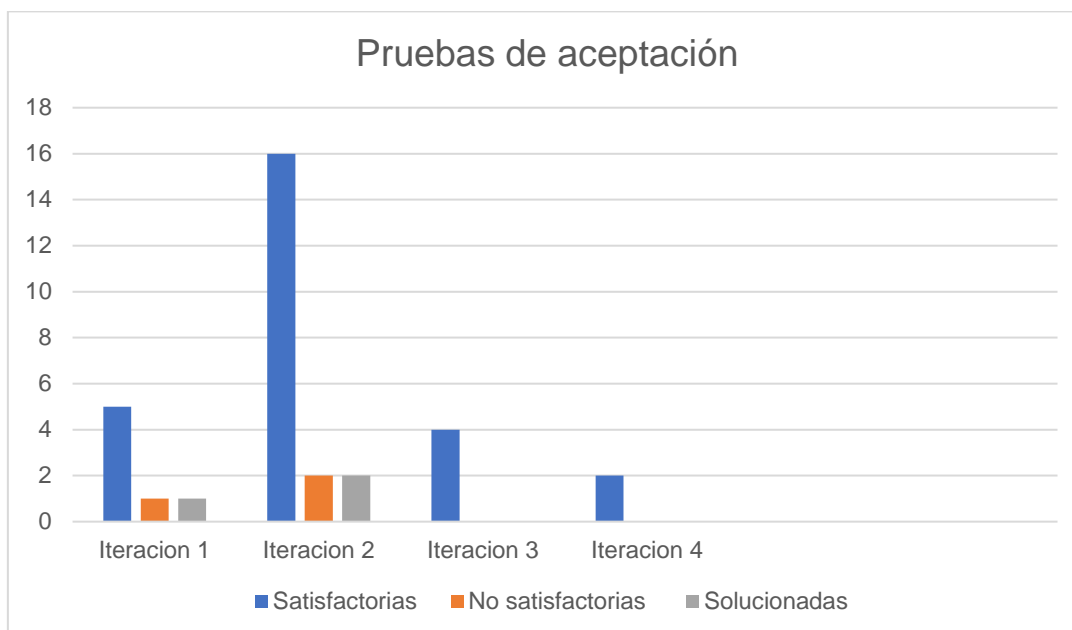


Gráfico 2. Pruebas de aceptación.

Este gráfico representa de manera visual como se comportó la aplicación de las pruebas de aceptación en cada iteración al desarrollar la solución informática. Realizándose 30 pruebas, de las cuales 27 resultaron de manera satisfactoria, cumpliendo todos los requisitos, mientras que 3 presentaron una serie de errores, que fueron solucionadas en un corto período de tiempo.

Criterios de validez realizados por los especialistas a partir de las técnicas aplicadas.

Para constatar la validez de la propuesta se aplicaron métodos empíricos de investigación como la entrevista y la encuesta. Para ello se utilizó una muestra de 3 especialistas del área de tecnología de la Facultad 4 de la UCI.

A través de la entrevista (ver anexo 3) se pudo verificar que los tres especialistas emitieron su criterio por separado y aun así coinciden en los siguientes aspectos:

- Se resolverán los problemas de gestión de los medios del departamento de tecnología de la Facultad 4 de la UCI con la implementación de una aplicación informática, pues admitiría tener un control total de donde están ubicados los medios, saber de cualquier movimiento y tener un informe del estado actual de cada uno.
- Los procesos actualmente se realizan de forma manual sujetos a errores al registrar o controlar un equipo o tecnología. El formato duro no permite la

edición de los contenidos a no ser que se vuelva a realizar este proceso, al igual que se tienen grandes riesgos de perder, duplicar o cometer errores con la información recolectada.

- Para el diseño e implementación del software si se necesita de alguna tecnología específica. Las más adecuadas para ello sería la tecnología web ya que es la que brinda mayor flexibilidad y adaptabilidad.
- Los problemas que se presentan a la hora de realizar estos procesos de forma manual están dados por la posibilidad de que la información se duplique, se pierda o sea fácilmente modificable al existir un alto riesgo de acceso, esto permite la ejecución de hechos delictivos, pérdidas económicas y de manera general se afecta la actividad productiva del área.

En la encuesta realizada (ver anexo 10) a especialistas con el propósito de recopilar información sobre el proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI, a partir del diseño e implementación del software propuesto, se pudo constatar los siguientes criterios:

1. La metodología utilizada para el diseño e implementación del software propuesto es la más adecuada, ya que se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.
2. Consideran que el software propuesto posee las características necesarias para ser utilizado con el fin que se creó, ya que este brinda mayor seguridad y rapidez a la hora de manejar los datos de los diferentes medios con que cuenta el departamento de informatización de la Facultad 4 de la UCI.
3. Expresan que el diseño e implementación del software resulta beneficioso para el cumplimiento de los objetivos propuestos, por ser el mismo robusto y enfocado hacia la solución a la problemática existente.
4. Reflexionan que el diseño de la interfaz gráfica del software constituye el más adecuado para este, pues cumple con los estándares internacionales más actuales para este tipo de tecnología.
5. Atribuyen que el diseño e implementación del software propuesto es adecuado.

3.5. Funcionamiento del sistema.

El sistema está compuesto por tres aplicaciones. Las mismas son: aplicación web, aplicación móvil y aplicación de escritorio. Tienen como objetivo mejorar la gestión de los medios tecnológicos de la Facultad 4 de la UCI. Permitiendo tomar los datos de los dispositivos, guardarlos, imprimirlos y verificarlos de manera más rápida y eficiente.

Aplicación web

Está compuesta por un servidor y página web. El servidor fue realizado con el framework Spring Boot y lenguaje de programación Kotlin, cuyo principal objetivo es darles un acceso central a todas las aplicaciones. Para su ejecución es necesario tener instalado el JDK de JAVA versión superior a 11, con un espacio de almacenamiento de 500 Mb, y un 1Gb de RAM.

A continuación, se presenta un fragmento del código del controlador de locales, que es el encargado de dar acceso a la información de los locales a las restantes aplicaciones y permite además modificar estos datos mediante la clase *LocalController*. Sus principales métodos son listar, guardar, modificar y borrar.

```
-
@CrossOrigin
@RequestMapping("/Local")
@PreAuthorize("hasRole('ADMIN')")
class LocalController {

    @Autowired
    lateinit var localService: LocalService

    @Autowired
    lateinit var localMapper: LocalMapper

    // Jesus F. Vazquez Biltre +1
    @GetMapping
    @ResponseBody
    @PreAuthorize("hasRole('USER') or hasRole('ADMIN')")
    fun list(): List<LocalNotDeviceWhitNumberDevicesDTO> {
        return localMapper.map(localService.list())
    }

    // Jesus F. Vazquez Biltre +1
    @PostMapping
    @ResponseBody
    fun save(@RequestBody local: LocalNotIdNoDevicesDTO): LocalNotDeviceWhitNumberDevicesDTO {
        return localMapper.map(localService.save(localMapper.map(local)))
    }

    // Jesus F. Vazquez Biltre +1
    @DeleteMapping
    @ResponseBody
    fun delete(@RequestBody ids: Array<Int>) {
        localService.delete(ids)
    }

    // Jesus F. Vazquez Biltre +1
    @PutMapping
    @ResponseBody
    fun update(@RequestBody local: LocalNotDevicesDTO): LocalNotDeviceWhitNumberDevicesDTO {
        return localMapper.map(localService.update(localMapper.map(local)))
    }
}
```

Ilustración 13. Código ejemplo servidor web.

En el código anterior se evidencia la ruta establecida para acceder y gestionar los recursos de los locales: también la gestión de roles mediante la notación

@PreAutorize, también se pone en práctica la inyección de dependencia usando la notación @Autowired.

La página web muestra diferentes vistas relacionadas con la gestión de los datos de los medios tecnológicos. Está compuesta por gestión de usuarios, de locales y dispositivos. Escrito en JavaScript utilizando el framework React. Cuenta con diferentes componentes como tablas, botones, checkbox, íconos, que ayudan que la gestión de los datos sea más agradable a la vista. La imagen siguiente muestra la vista de gestión de dispositivos.

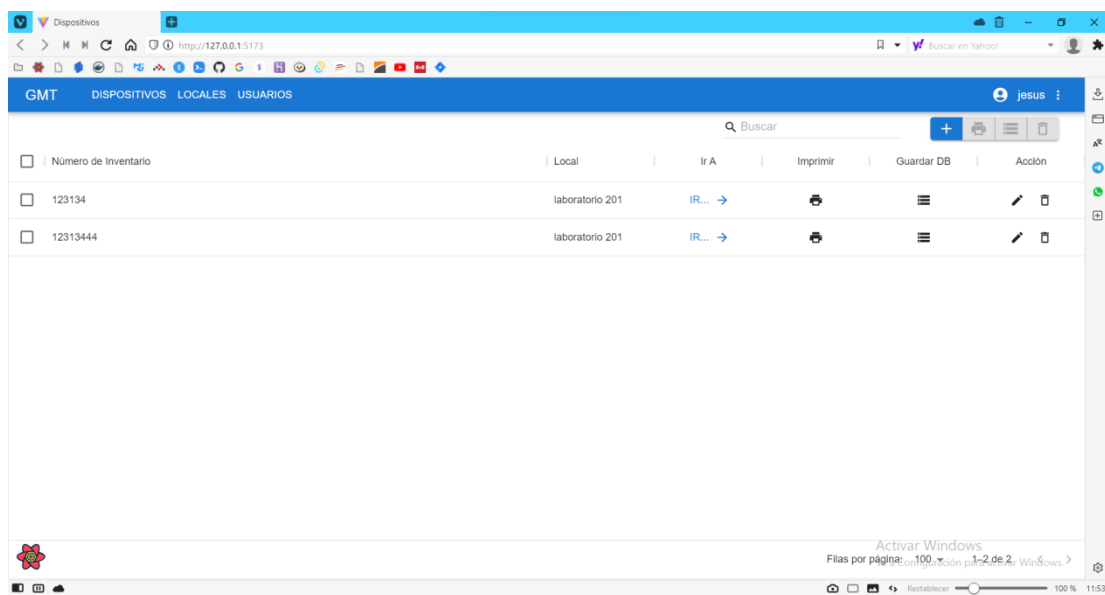


Ilustración 14. Vista ejemplo página web.

Se aprecian los botones que realizan funcionalidades tales como: añadir, modificar, eliminar e imprimir estos medios tecnológicos que aparecen listados junto al laboratorio al que pertenecen. Además del botón exportar Base de datos, el cual permite que se genere un archivo con el nombre *apk.db*.

Aplicación móvil

Esta aplicación permite buscar y cargar una base de datos en el almacenamiento del dispositivo donde fue instalada, muestra los datos de los medios tecnológicos y mediante el escaneo del código QR verifica si existen o no en la base de datos cargada previamente. Está desarrollada con el framework Flutter y lenguaje de programación Dart. Está pensada para dispositivos *android* con versión superior a 4.0. Cuenta con un botón para filtrar, otro para escanear activando la cámara, y uno para cargar su base de datos.

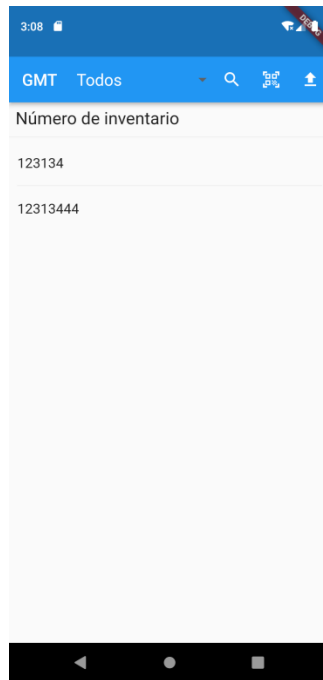


Ilustración 15. Vista ejemplo aplicación móvil.

Aplicación de escritorio

Esta aplicación es la encargada de obtener los datos de los medios tecnológicos para enviarlos al servidor web. Para ejecutarla es necesario tener instalado la máquina virtual de JAVA versión superior a la 11, con un espacio de almacenamiento mínimo de 20Mb. Es necesario que en el dispositivo donde se esté ejecutando se encuentre una variable de entorno con el nombre MB, que almacena el medio básico y otra con el nombre NS la cual almacena el número de sello del medio. Además, se necesita que se le pase por parámetros la dirección IP del servidor web.

```

fun send(): Boolean {
    try {
        Unirest.post( uri: "http://${Constant.baseUrl}/device/component").body(device) RequestBodyEntity!
            .header( name: "Authorization", Constant.token).header( name: "Content-Type", value: "application/json").asEmpty() HttpResponse<raw> Any!>!
            .ifSuccess { it: Any!
                println()
                println("***** Success connect to Server ip: ${Constant.baseUrl} *****")
            }.ifFailure { it: Any!
                println()
                println("***** Failure connect to Server ip: ${Constant.baseUrl} *****")
                println("***** Cause: server error ")
            }
        return true
    } catch (e: Exception) {
        println()
        println("***** Failure connect to Server ip: ${Constant.baseUrl} *****")
        println("***** Cause: no connection ")
        return false
    }
}

```

Ilustración 16. Código ejemplo aplicación cliente.

En la imagen se muestra un fragmento del código de la clase Service donde se evidencia el método send que se encarga de enviar los datos del dispositivo al servidor.

```
class Motherboard : OperativeSystem() {  
    var serialNumber: String = hardware.computerSystem.baseboard.serialNumber  
    var model: String = hardware.computerSystem.baseboard.model  
}
```

Ilustración 17. Código ejemplo aplicación cliente.

En este caso se evidencia la clase Motherboard que es la encargada de obtener los datos de la *Motherboard* del dispositivo.

3.6 Conclusiones del capítulo.

En el presente capítulo se mostraron los resultados de la implementación y pruebas realizadas al sistema de gestión de medios tecnológicos, así como los criterios de validez realizados por los especialistas a partir de las técnicas aplicadas, por lo que se pudo constatar que:

- Al definirse las tareas de ingeniería correspondientes a cada HU se obtuvo un nivel más detallado de la implementación de la aplicación.
- Se realizaron pruebas unitarias en conjunto con las de aceptación que permitieron detectar errores durante las iteraciones. Luego de ser corregidas en su totalidad, validaron su correcto funcionamiento en relación a los requisitos definidos por el cliente.
- Los especialistas consideran adecuado el diseño e implementación de software, al cumplir el mismo con los estándares internacionales, presentar una interfaz fluida y adaptada a la resolución de la problemática existente, así como cuenta con un diseño robusto y enfocado al cliente.

CONCLUSIONES GENERALES

Al finalizar la presente investigación se puede concluir que:

- El estudio de los procesos de gestión y control de los medios tecnológicos de la Facultad 4, permitió el diseño de un sistema que se encargue de realizar dichos procesos.
- El desarrollo del sistema para la gestión de los medios tecnológicos de la facultad 4. Permite un mejor control de los mismos.
- El estudio de sistemas similares, demostró que hay tecnologías que dan solución a la problemática planteada, pero son obsoletas, por lo que se hizo necesario la creación de un nuevo sistema de este tipo.
- Se logró una efectiva ingeniería de software y gestión del proyecto, gracias a la utilización de la metodología XP.
- Como resultado se obtuvieron un sistema informático que consta de tres aplicaciones funcionales (probadas durante su proceso de desarrollo y validadas utilizando pruebas de aceptación), desarrollada con tecnologías libres y provista de elementos de seguridad, estándares y un entorno amigable, gracias a la utilización de los frameworks seleccionados Flutter, React y Spring Boot.

RECOMENDACIONES

A partir de los resultados obtenidos se recomienda:

- Agregar la funcionalidad de gestionar otros componentes de un medio tecnologico como monitor, teclado e impresoras.
- Utilizar el sistema en todas las facultades de la Universidad de las Ciencias Informáticas.

BIBLIOGRAFÍA

- [1] «La evolución de las TICs desde el ábaco al Smartphone», *VIU*. <https://www.universidadviu.com/int/actualidad/nuestros-expertos/la-evolucion-de-las-tics-desde-el-abaco-al-smartphone> (accedido 23 de mayo de 2022).
- [2] «Recursos Tecnológicos - Qué son, tipos, importancia y ejemplos», *Concepto*. <https://concepto.de/recursos-tecnologicos/> (accedido 23 de mayo de 2022).
- [3] «Universidad de las Ciencias Informáticas - EcuRed». https://www.ecured.cu/Universidad_de_las_Ciencias_Inform%C3%A1ticas (accedido 24 de mayo de 2022).
- [4] I. Fernández Correa, «Aplicación móvil para el monitoreo pasivo de los recursos de hardware en la plataforma Nova-LTSP», jun. 2019, Accedido: 17 de junio de 2022. [En línea]. Disponible en: <https://repositorio.uci.cu/jspui/handle/123456789/10073>
- [5] «Definición de medio — Definicion.de», *Definición.de*. <https://definicion.de/medio/> (accedido 22 de mayo de 2022).
- [6] «Definición de recursos tecnológicos — Definicion.de», *Definición.de*. <https://definicion.de/recursos-tecnologicos/> (accedido 22 de mayo de 2022).
- [7] «Concepto de gestión — Definicion.de», *Definición.de*. <https://definicion.de/gestion/> (accedido 22 de mayo de 2022).
- [8] «Definición de página web — Definicion.de», *Definición.de*. <https://definicion.de/pagina-web/> (accedido 22 de mayo de 2022).
- [9] «Definición de sitio web — Definicion.de», *Definición.de*. <https://definicion.de/sitio-web/> (accedido 22 de mayo de 2022).
- [10] «Servidor Web - Concepto, usos y características», *Concepto*. <https://concepto.de/servidor-web/> (accedido 16 de septiembre de 2022).
- [11] XPERTOSOLUTIONS.COM, «¿Qué es una Aplicación Móvil?» <https://www.xpertosolutions.com:443/x/noticia/item/que-es-una-aplicacion-movil> (accedido 17 de junio de 2022).
- [12] «En informática ¿Qué es una aplicación cliente y qué es un servicio en la nube?», *BrandWidget Blog*, 19 de agosto de 2011. <https://brandwidget.wordpress.com/2011/08/18/en-informatica-%c2%bfque-es-una-aplicacion-cliente-y-que-es-un-servicio-en-la-nube/> (accedido 17 de junio de 2022).
- [13] «OCS Inventory NG — NethServer 6.9». <https://docs.nethserver.org/es/stable/ocs.html> (accedido 22 de mayo de 2022).
- [14] «OCS Inventory NG – A plataforma para registrar PCs e software», *Pplware*, 17 de julio de 2014. <https://pplware.sapo.pt/linux/ocs-inventory-ng-a-plataforma-para-registrar-pcs-e-software/> (accedido 9 de noviembre de 2022).
- [15] «Open-Audit: el inventario de red, la auditoría, la documentación y la herramienta de gestión.» <https://www.open-audit.org/index.php> (accedido 22 de mayo de 2022).
- [16] «[ayuda] software libre para inventario de equipos - Inf...» https://www.taringa.net/+computacion/ayuda-software-libre-para-inventario-de-equipos_sdy5a (accedido 9 de noviembre de 2022).
- [17] A. E. M. Rodríguez, J. T. González, L. S. Labrada, y D. S. López, «GESTOR DE RECURSOS DE HARDWARE Y SOFTWARE», p. 6.
- [18] «Metodologías de desarrollo de Software - EcuRed». https://www.ecured.cu/Metodologias_de_desarrollo_de_Software (accedido 22 de mayo de 2022).
- [19] «Conoce las 3 metodologías ágiles más usadas», *OpenWebinars.net*, 8 de marzo de 2018. <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/> (accedido 18 de junio de 2022).

- [20] «Lenguaje de programación (informática) - EcuRed». [https://www.ecured.cu/Lenguaje_de_programaci%C3%B3n_\(inform%C3%A1tica\)](https://www.ecured.cu/Lenguaje_de_programaci%C3%B3n_(inform%C3%A1tica)) (accedido 22 de mayo de 2022).
- [21] «Syntax and grammar - Kotlin language specification». <https://kotlinlang.org/spec/syntax-and-grammar.html#syntax-and-grammar> (accedido 22 de mayo de 2022).
- [22] «Dart overview». <https://dart.dev/overview.html> (accedido 22 de mayo de 2022).
- [23] J. P. de la familia L. de programación interpretado L. de programación que permite a los desarrolladores crear acciones en sus páginas web C. B. E. D. por N. C. Corp y M. F. L. C. C. Attribution, «JavaScript - EcuRed». <https://www.ecured.cu/JavaScript> (accedido 18 de junio de 2022).
- [24] J. L. de programación orientado a objetos C. S. M. F. de C. 1991 S. web J. en Español, «Java (lenguaje de programación) - EcuRed». [https://www.ecured.cu/Java_\(lenguaje_de_programaci%C3%B3n\)](https://www.ecured.cu/Java_(lenguaje_de_programaci%C3%B3n)) (accedido 18 de junio de 2022).
- [25] F. L. palabra inglesa framework define, E. T. Generales, U. C. E. D. Conceptos, P. Y. C. P. E. U. T. D. P. Particular, y Q. S. C. R. P. E. Y. R. N. P. D. Í. Similar, «Framework - EcuRed». <https://www.ecured.cu/Framework> (accedido 22 de mayo de 2022).
- [26] «¿Qué es Spring Boot?», *Codmind*, 17 de marzo de 2021. <https://blog.codmind.com/que-es-spring-boot/> (accedido 22 de mayo de 2022).
- [27] «Bienvenido al proyecto Apache Struts». <https://struts.apache.org/> (accedido 10 de agosto de 2022).
- [28] «Flutter architectural overview». <https://docs.flutter.dev/resources/architectural-overview> (accedido 22 de mayo de 2022).
- [29] R. J. P. de la familia J. B. de programación C. J. W. L. inicial 29 de mayo de 2013 Ú. versión estable 17 0 1 S. O. compatibles M. L. L. B. + P. C. S. web <https://reactjs.org>, «React js - EcuRed». https://www.ecured.cu/React_js (accedido 18 de junio de 2022).
- [30] S. G. de B. de D. U. S. G. de B. de D. es un sistema de software que permite la definición de bases de datos, «Sistema Gestor de Base de Datos - EcuRed». https://www.ecured.cu/Sistema_Gestor_de_Base_de_Datos (accedido 22 de mayo de 2022).
- [31] «¿Qué es Microsoft SQL Server? - Definición en WhatIs.com», *ComputerWeekly.es*. <https://www.computerweekly.com/es/definicion/Microsoft-SQL-Server> (accedido 18 de junio de 2022).
- [32] P. S. de gestión de base de datos relacional orientada a objetos y libre C. P. G. D. G. Ú. versión estable 9 4-18 de D. de 2014 G. R. S. O. compatibles M. L. B. S. web <http://www.postgresql.org>, «PostgreSQL - EcuRed». <https://www.ecured.cu/PostgreSQL> (accedido 22 de mayo de 2022).
- [33] «¿Qué es MySQL?» <https://www.espestudio.com/noticias/que-es-mysql> (accedido 16 de septiembre de 2022).
- [34] «El concepto de IDE». <https://www.redhat.com/es/topics/middleware/what-is-ide> (accedido 22 de mayo de 2022).
- [35] «Qué es Visual Studio Code y qué ventajas ofrece», *OpenWebinars.net*, 22 de julio de 2022. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/> (accedido 10 de octubre de 2022).
- [36] «Eclipse, entorno de desarrollo integrado - EcuRed». https://www.ecured.cu/Eclipse,_entorno_de_desarrollo_integrado (accedido 18 de junio de 2022).
- [37] «Arquitectura de software - EcuRed». https://www.ecured.cu/Arquitectura_de_software (accedido 19 de junio de 2022).
- [38] «Arquitectura Cliente Servidor - EcuRed». https://www.ecured.cu/Arquitectura_Cliente_Servidor (accedido 19 de junio de 2022).

- [39] «Arquitectura Cliente-Servidor | Mapa Mental». <https://www.goconqr.com/mapamental/25270764/arquitectura-cliente-servidor> (accedido 9 de noviembre de 2022).
- [40] N. Soto, «¿Qué son los patrones de diseño en programación? 2022», *Craft - Code | La Academia de las Buenas Prácticas*, 2 de julio de 2021. <https://craft-code.com/que-son-los-patrones-de-diseno/> (accedido 23 de mayo de 2022).
- [41] «Patrones de Asignación de Responsabilidades - EcuRed». https://www.ecured.cu/Patrones_de_Asignaci%C3%B3n_de_Responsabilidades (accedido 23 de mayo de 2022).
- [42] M. R, «Tarjetas CRC (Clase-Responsabilidad-Colaborador)», *Medium*, 8 de diciembre de 2018. <https://medium.com/@marcosrrg9813/tarjetas-crc-clase-responsabilidad-colaborador-81924cec3af0> (accedido 23 de mayo de 2022).
- [43] «Estándares de codificación - ¡Mejora tu código!», *Ohmyroot!*, 12 de enero de 2017. <https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/> (accedido 20 de junio de 2022).
- [44] M. N. L. Santovenia, «Módulos Graficación y Reportes para la plataforma web Medicando», p. 80.
- [45] J. J. Gutiérrez, M. J. Escalona, M. Mejías, y J. Torres, «PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA», p. 12.
- [46] «👉 ¿Qué son las pruebas unitarias y cómo llevar una a cabo?», *Yeeply*, 22 de agosto de 2019. <https://www.yeeply.com/blog/que-son-pruebas-unitarias/> (accedido 20 de junio de 2022).
- [47] «JUnit - EcuRed». <https://www.ecured.cu/JUnit> (accedido 20 de junio de 2022).
- [48] T. 03-P. del desarrollo T. de planificación M. de desarrollo L. metodología M. – T. dice, «Pruebas de Aceptación», *manuel.cillero.es*. <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/aceptacion/> (accedido 20 de junio de 2022).

ANEXOS

Anexo 1. Tareas de ingeniería

Iteración 2

Tabla 76. Tarea de Ingeniería # 7

Tarea	
Número de la tarea: 7	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar crear local	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita insertar un local en el sistema mediante la entrada de su nombre y dirección IP.	

Tabla 77. Tarea de Ingeniería # 8

Tarea	
Número de la tarea: 8	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar modificar local	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita modificar nombre o dirección IP de un local ya registrado en el sistema.	

Tabla 78. Tarea de Ingeniería # 9

Tarea	
Número de la tarea: 9	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar borrar local	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita borrar uno o varios locales registrado en el sistema.	

Tabla 79. Tarea de Ingeniería # 10

Tarea	
Número de la tarea: 10	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar listar local	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	

Descripción: Se implementa una funcionalidad en la aplicación web que permita mostrar los locales registrados en el sistema.

Tabla 80. Tarea de Ingeniería # 11

Tarea	
Número de la tarea: 11	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar filtrar local	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita filtrar los locales registrado en el sistema.	

Tabla 81. Tarea de Ingeniería # 12

Tarea	
Número de la tarea: 12	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar crear medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
<p>Descripción: Se implementa una funcionalidad en la aplicación web que permita ingresar los datos de un medio tecnológico tales como:</p> <ul style="list-style-type: none"> • Número de inventario • Número de sello • Local al que pertenece • Número de serie y modelo de la Motherboard • Modelo del procesador • Tipo y capacidad total de la Ram • Número de serie, marca y capacidad total del disco duro • Modelo de la fuente de energía • Número de serie y modelo del lector de CD • Número de serie y modelo de la tarjeta de video • Número de serie y modelo de la tarjeta de red <p>Y los registra en el sistema.</p>	

Tabla 82. Tarea de Ingeniería # 13

Tarea	
Número de la tarea: 13	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar modificar medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4

Programador responsable: Jesús Francisco Vázquez Biltre
Descripción: Se implementa una funcionalidad en la aplicación web que permita modificar los datos de un medio tecnológico registrado en el sistema, con excepción del número de inventario, que no es modificable.

Tabla 83. Tarea de Ingeniería # 14

Tarea	
Número de la tarea: 14	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar borrar medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita borrar uno o varios medios tecnológicos.	

Tabla 84. Tarea de Ingeniería # 15

Tarea	
Número de la tarea: 15	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar listar medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita visualizar los datos de los medios tecnológicos.	

Tabla 85. Tarea de Ingeniería # 16

Tarea	
Número de la tarea: 16	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar filtrar medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita filtrar los medios tecnológicos listados.	

Tabla 86. Tarea de Ingeniería # 17

Tarea	
Número de la tarea: 17	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar crear cambios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	

Descripción: Se implementa una funcionalidad en la aplicación web que permita crear un cambio a un medio tecnológico determinado, a partir de la fecha, tipo de cambio y descripción del mismo.

Tabla 87. Tarea de Ingeniería # 18

Tarea	
Número de la tarea: 18	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar modificar cambio	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita modificar un cambio ya registrado.	

Tabla 88. Tarea de Ingeniería # 19

Tarea	
Número de la tarea: 19	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar borrar cambio	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita borrar uno o varios cambios realizados.	

Tabla 89. Tarea de Ingeniería # 20

Tarea	
Número de la tarea: 20	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar listar cambios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita listar los cambios realizados a un medio tecnológico.	

Tabla 90. Tarea de Ingeniería # 21

Tarea	
Número de la tarea: 21	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar filtrar cambios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Jesús Francisco Vázquez Biltre	

Descripción: Se implementa una funcionalidad en la aplicación web que permita filtrar los cambios realizados a un medio tecnológico.

Tabla 91. Tarea de Ingeniería # 22

Tarea	
Número de la tarea: 22	Número de Historia de Usuario: 6
Nombre de la Tarea: Implementar funcionalidad detectar cambios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita comparar los datos recibidos desde la aplicación cliente, de un medio tecnológico, en un periodo de tiempo dado y si estos datos sufren alguna variación actualiza la base de datos.	

Tabla 92. Tarea de Ingeniería # 23

Tarea	
Número de la tarea: 23	Número de Historia de Usuario: 7
Nombre de la Tarea: Implementar imprimir ficha técnica	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita imprimir una o varias fichas técnicas, junto con un código QR que represente el medio tecnológico a la que pertenece.	

Tabla 93. Tarea de Ingeniería # 24

Tarea	
Número de la tarea: 24	Número de Historia de Usuario: 7
Nombre de la Tarea: Implementar imprimir expediente.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación web que permita imprimir uno o varios expedientes, que contengan los datos del medio tecnológico, así como su historial de cambio.	

Tabla 94. Tarea de Ingeniería # 25

Tarea	
Número de la tarea: 25	Número de Historia de Usuario: 8
Nombre de la Tarea: Exportar base de datos	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Jesús Francisco Vázquez Biltre, Roxana Mérida Sánchez.	
Descripción: Se implementa una funcionalidad en la aplicación web que permita exportar la base de datos en un fichero .db .	

Iteración 3

Tabla 95. Tarea de Ingeniería # 26

Tarea	
Número de la tarea: 26	Número de Historia de Usuario: 9
Nombre de la Tarea: Implementar listar medios tecnológicos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa en la aplicación móvil donde se permita visualizar los medios tecnológicos registrados en el sistema.	

Tabla 96. Tarea de Ingeniería # 27

Tarea	
Número de la tarea: 27	Número de Historia de Usuario: 9
Nombre de la Tarea: Implementar filtrar medios tecnológicos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.75
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación móvil que permita filtrar los medios tecnológicos listados a través de su número de inventario.	

Tabla 97. Tarea de Ingeniería # 28

Tarea	
Número de la tarea: 28	Número de Historia de Usuario: 10
Nombre de la Tarea: Implementar funcionalidad comprobar existencia de medio tecnológico	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Programador responsable: Jesús Francisco Vázquez Biltre	

Descripción: Se implementa una funcionalidad en la aplicación móvil que compruebe si un medio tecnológico se encuentra registrado en el sistema y si sus datos están correctos. Esta acción se llevará a cabo a partir del escaneo de un código QR ubicado en la ficha técnica del medio tecnológico.

Tabla 98. Tarea de Ingeniería # 29

Tarea	
Número de la tarea: 29	Número de Historia de Usuario: 11
Nombre de la Tarea: Cargar base de datos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación móvil que permita buscar y cargar desde el almacenamiento del dispositivo móvil el fichero .db con la base de datos.	

Iteración 4

Tabla 99. Tarea de Ingeniería # 30

Tarea	
Número de la tarea: 30	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener número de inventario y número de sello.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el número de inventario y de sello, de un medio tecnológico desde una variable de entorno.	

Tabla 100. Tarea de Ingeniería # 31

Tarea	
Número de la tarea: 31	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Motherboard	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el número de serie y modelo de la Motherboard de un medio tecnológico.	

Tabla 101. Tarea de Ingeniería # 32

Tarea

Número de la tarea: 32	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Procesador	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el modelo del Procesador de un medio tecnológico.	

Tabla 102. Tarea de Ingeniería # 33

Tarea	
Número de la tarea: 33	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Disco Duro	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener capacidad, marca y número de serie del Disco duro de un medio tecnológico.	

Tabla 103. Tarea de Ingeniería # 34

Tarea	
Número de la tarea: 34	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Memoria RAM	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el tipo y capacidad de la Memoria RAM de un medio tecnológico.	

Tabla 104. Tarea de Ingeniería # 35

Tarea	
Número de la tarea: 35	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos fuente de alimentación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el modelo de la fuente de alimentación de un medio tecnológico.	

Tabla 105. Tarea de Ingeniería # 36

Tarea	
Número de la tarea: 36	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Unidad Óptica	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el número de serie y modelo de la Unidad Óptica de un medio tecnológico.	

Tabla 106. Tarea de Ingeniería # 37

Tarea	
Número de la tarea: 37	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Tarjeta de Video	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el número de serie y modelo de la Tarjeta de Video de un medio tecnológico.	

Tabla 107. Tarea de Ingeniería # 38

Tarea	
Número de la tarea: 38	Número de Historia de Usuario: 12
Nombre de la Tarea: Implementar funcionalidad obtener datos Tarjeta de Red	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita obtener el número de serie y modelo de la Tarjeta de Red de un medio tecnológico.	

Tabla 108. Tarea de Ingeniería # 39

Tarea	
Número de la tarea: 39	Número de Historia de Usuario: 13
Nombre de la Tarea: Implementar funcionalidad enviar datos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Jesús Francisco Vázquez Biltre	
Descripción: Se implementa una funcionalidad en la aplicación cliente que permita enviar los datos recolectados, cada una hora, al servidor, para incorporarlos al sistema.	

Anexo 2. Entrevista a Especialistas del área de informatización.

Objetivo: conocer la estructura organizativa del proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI.

Estimado especialista:

Estamos realizando una entrevista con el propósito de conocer la estructura organizativa del proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI. Le sugerimos que contribuya con nosotros.

Muchas gracias.

1. ¿Se resolverán los problemas de gestión de los medios del departamento de tecnología de la Facultad 4 de la UCI con la implementación de una aplicación informática? ¿Por qué?
2. ¿Los procesos de gestión de medios se realizan de forma manual o están automatizados?
3. ¿Para el diseño e implementación del software se necesita de alguna tecnología específica? ¿Cuáles serían las más adecuadas?
4. ¿Qué problemas se presentan a la hora de realizar estos procesos de forma manual?

Anexo 3. Tabla comparativa metodologías.

Metodología	Ventajas	Desventajas
Extreme Programming (XP)	<ul style="list-style-type: none"> • Es muy eficiente durante el proceso de prueba y planificación. • Tasa de errores muy pequeña. • Facilita los cambios • Origina una programación muy organizada • Fomenta la comunicación entre los desarrolladores y los clientes. 	<ul style="list-style-type: none"> • Mayor esfuerzo de trabajo • El cliente se implica en el proceso • Requiere control de versiones • Requiere autodisciplina en la aplicación
Scrum	<ul style="list-style-type: none"> • Gestión de las expectativas del usuario. • Resultados anticipados. • Flexibilidad y adaptación a los contextos. • Gestión sistemática de riesgos. • Visión global. 	<ul style="list-style-type: none"> • Se aplica a equipos reducidos • Requiere una exhaustiva definición de las tareas y sus plazos • Requiere de perfiles senior en su aplicación • Difícil escalabilidad • No se integra fácilmente con enfoque clásico de gestión de proyectos
Kanban	<ul style="list-style-type: none"> • Evita los excesos de producción. • Puedes tener información en tiempo real acerca de la producción. • Ayuda a detectar rápidamente los problemas en la cadena de producción. • Los pedidos se entregan en el momento oportuno, lo que ayuda a aprovechar cada rincón del almacén. • Al estar todo etiquetado, puedes tener certeza de lo que está funcionando bien y lo que está generando problemas o retrasos. 	<ul style="list-style-type: none"> • Kanban apuesta por lo justo, lo que significa que siempre vas a trabajar con los recursos precisos lo que puede provocar producción insuficiente. • En condiciones extremas, cuando los plazos de entrega no pueden cumplirse, se ve afectado el calendario de producción para el resto de pedidos. Esto puede ocasionar una cadena de incumplimiento. • Cuando hay demasiadas etiquetas y poca preparación, los miembros del equipo pueden confundirse fácilmente.

Fuente: Elaboración propia.

Anexo 4. Tabla comparativa lenguajes de programación.

Lenguaje	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • Tiene una curva de aprendizaje corta. • Está orientado a los objetos. • Cuenta con liberación de memoria. • Es compatible con librerías estándar y editores. • Ofrece una gestión de error 	<ul style="list-style-type: none"> • Es de lenguaje Interpretado. • Requiere de experiencia en programación. • Es de sintaxis compleja. • Se ejecuta solo en dispositivos y equipos aptos. • Requiere un espacio de memoria significativo
Kotlin	<ul style="list-style-type: none"> • Curva de aprendizaje corta, gracias a la fácil lectura de su simple sintaxis. • Eliminación de los NPE (Null Pointer Exception) del código. • Posibilidad de manejar un constructor primario y de ser necesario constructores secundarios. • Admite soporte para variables genéricas. • Permite desarrollos multiplataforma, haciendo que sea interoperable. 	<ul style="list-style-type: none"> • Tamaño extra de tiempo de ejecución. • Pequeña comunidad de desarrolladores, por lo que carece de materiales de aprendizaje y asistencia profesional • Velocidad de compilación más lenta que Java
JavaScript	<ul style="list-style-type: none"> • Es un lenguaje muy sencillo. • Es rápido, por lo tanto, tiende a ejecutar las funciones inmediatamente. • Cuenta con múltiples opciones de efectos visuales. • Es soportado por los navegadores más populares y es compatible con los dispositivos más modernos, incluyendo iPhone, móviles y PS3. • Es muy versátil, puesto que es muy útil para desarrollar páginas dinámicas y aplicaciones web. • Es multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil. • Es el único lenguaje que permite trabajar modo FullStack en cualquier tipo de desarrollo de programación. 	<ul style="list-style-type: none"> • En el FrontEnd sus códigos son visibles, por lo tanto, pueden ser leídos por cualquier usuario. • Tiende a introducir gran cantidad de fragmentos de código en los sitios web. • No es compatible en todos los navegadores de manera uniforme. • Los usuarios tienen la opción de desactivar JavaScript desde su navegador.
Dart	<ul style="list-style-type: none"> • Programación estructurada y flexible 	<ul style="list-style-type: none"> • Es relativamente nuevo. • Disponibilidad de pocos

	<ul style="list-style-type: none"> • Lenguaje familiar y fácil de aprender • Lenguaje basado en clases e interfaces o POO • Detrás de su programación se encuentra Google, lo que ofrece perspectivas a largo plazo para el desarrollo del lenguaje. • Es de código abierto • Funciona en todos los navegadores móviles y de escritorio actuales. 	<p>materiales de aprendizaje.</p>
PHP	<ul style="list-style-type: none"> • Es que es de código abierto y no tiene costos. • Las aplicaciones basadas en PHP pueden ejecutarse en cualquier sistema operativo como UNIX, Linux y Windows, etc. • Tiene menos curva de aprendizaje, porque es sencillo y sencillo de usar. • Es más estable desde unos pocos años con la ayuda de proporcionar soporte continuo a varias versiones. • Ayuda a administrar el código fácilmente. 	<ul style="list-style-type: none"> • No es tan seguro debido a su código abierto, ya que los archivos de texto ASCII suelen estar fácilmente disponibles. • No es adecuado para aplicaciones web gigantes basadas en contenido. • Tiene un tipo débil, que puede causar datos y conocimientos incorrectos al usuario. • Los frameworks PHP aprendieron a usar las funcionalidades integradas de PHP para evitar escribir código adicional. • El uso de más funciones del marco y las herramientas de PHP provoca un rendimiento deficiente de las aplicaciones en línea. • PHP no permite cambios o modificaciones en el comportamiento principal de las aplicaciones en línea.

Fuente: Elaboración propia.

Anexo 5. Tabla comparativa framework.

Framework	Ventajas	Desventajas
Spring boot	<ul style="list-style-type: none"> • Lógica aplicable con POJO • Código base listo que requiere pocos (o ningún) cambio • No se necesita un servidor de aplicaciones • Permite pruebas unitarias y de integración • Arquitectura MVC • Inyección de dependencia (regulación de la dependencia externa) • Gestión programática de transacciones 	<ul style="list-style-type: none"> • Framework complejo que te exige tener claros muchos conceptos. • Como su funcionamiento es out of the box (fuera de la caja), así que te abstraer completamente de la complejidad intrínseca de Spring.
Apache Struts	<ul style="list-style-type: none"> • Implementa el patrón de arquitectura MVC (Model-View-Controller) cuyo propósito principal es separar el código de la capa de presentación del código de la capa de negocio facilitando la asignación de roles y la especialización defunciones dentro del marco de un proyecto. • Potencia la reutilización de código y soporta múltiples interfaces de usuario (Html, sHtml, Wml, Desktop applications, etc.). • Ya que Struts está basado en tecnologías estándares como Java Servlets, JavaBeans, XML, hereda el ambiente de estos estándares y puede operar con cualquier tecnología basada en JSP o Servlet. • Beneficio en términos de costo por licencia: Struts es un Open Source Software (OSS), lo que significa que su uso es libre. 	<ul style="list-style-type: none"> • Se requiere por parte de los desarrolladores un conocimiento profundo y detallado de Servlets y JSPs lo que representa una curva de aprendizaje, que de requerirse no se recomienda en proyectos cortos. • En comparación con la tecnología de Servlets y JSPs, los desarrolladores principiantes encuentran que la documentación en línea de www.apache.org es confusa y pobremente organizada. Existen menos libros de Struts que de Servlets y JSPs. • El tiempo de deployment de las aplicaciones desarrolladas con Struts se incrementa, ya que el servidor de aplicaciones tiene que buscar la ubicación.

Flutter	<ul style="list-style-type: none"> • Una única base de código para las principales plataformas de destino. • Lenguaje de programación Dart fácil de aprender. • El paradigma todo es un widget ofrece numerosas posibilidades. • Ejecución potente de las aplicaciones nativas en los smartphones. • Bibliotecas amplias con elementos de interfaz gráfica prefabricados. • Implementación sencilla de flujos de datos para proporcionar información actual a todos los usuarios. • Hot Reload acelera las pruebas durante el desarrollo. 	<ul style="list-style-type: none"> • El código del programa puede volverse confuso al integrar los <i>widgets</i>. • En caso de actualizar aspectos del diseño en los sistemas operativos, hay que actualizar los módulos Flutter. Como los módulos se integran en el programa de manera fija, también hay que compilar el programa e instalarlo en los dispositivos. • Todavía es un lenguaje nuevo y poco extendido, cuenta con una comunidad reducida.
React	<ul style="list-style-type: none"> • Facilita el proceso general de escritura de componentes mediante la utilización de JSX. • Aumenta la productividad y facilita un mayor mantenimiento. • Asegura un renderizado más rápido. • Garantiza código estable. • Es compatible con SEO 	<ul style="list-style-type: none"> • Es solo una capa de vista, por lo que necesita de librerías de terceros para otras funcionalidades. • La biblioteca en sí es bastante grande. • La curva de aprendizaje puede ser empinada.

Fuente: Elaboración propia.

Anexo 6. Tabla comparativa sistema gestor de base de datos.

Sistema gestor de base de datos	Ventajas	Desventajas
Microsoft SQL Server	<ul style="list-style-type: none"> • Aumenta la seguridad de los datos. • Facilidad de configuración. • Almacenamiento de datos optimizado • Soporte de recuperación de datos. 	<ul style="list-style-type: none"> • Costo. • Compatibilidad restringida. • Restricciones de hardware.
PostgreSQL	<ul style="list-style-type: none"> • Open source. • Altamente ampliable • En gran parte conforme con el estándar SQL. • Permite procesar tipos complejos de datos (p. ej., datos geográficos). • Amplia compatibilidad con varios lenguajes (Python, Java, Perl, PHP, C, C++, etc.) • Multiplataforma. 	<ul style="list-style-type: none"> • No está disponible por defecto en todos los hosters. • La documentación es mejorable y está solo disponible en inglés. • La velocidad de lectura es menor que en otros gestores.
MySQL	<ul style="list-style-type: none"> • MySQL es de uso libre y gratuito. • Bajo costo en requerimientos para la elaboración y ejecución del programa. • No se necesita disponer de Hardware o Software de alto rendimiento para la ejecución del programa. • Velocidad al realizar las operaciones y buen rendimiento. • Entorno con seguridad y encryptación. 	<ul style="list-style-type: none"> • Al ser de Software Libre, muchas de las soluciones para las deficiencias del software no están documentados ni presentan documentación oficial. • Se debe controlar/monitorizar el rendimiento de las aplicaciones en busca de fallos. • No es tan eficaz en aplicaciones que requieran de una constante modificación de escritura en BD.

Fuente: Elaboración propia.

Anexo 7. Tabla comparativa entorno de desarrollo integrado.

Parámetros	Eclipse	Visual Studio Code	NetBeans
Requisitos mínimos del sistema para instalar el IDE	Se recomienda 4 GB de RAM, 300 MB a 1 GB de disco duro, velocidad del procesador de 800 MHz a 1,5 GHz	Se requieren de 1 GB de RAM, procesador con 1.6 GHz de frecuencia. 1 GB de almacenamiento.	Se recomienda un mínimo de 2 a 4 GB de RAM, 750 MB de almacenamiento y procesador con velocidad de 800 MHz a 1,5 GHz
Costo	Eclipse se ofrece gratis	Se ofrece de forma gratuita.	No tiene costo
Generación automática de códigos	Eclipse proporciona el mecanismo para la generación automática de código. Sin embargo, no es eficaz y más fácil para el programador trabajar con él.	Cuenta con la herramienta <i>IntelliSense</i> , que proporciona sugerencias de código y terminaciones inteligentes en base a los tipos de variables, funciones, etc.	Proporciona mecanismos para la generación automática de código.
Flujo de trabajo	Maneja la implementación y se utiliza para proyectos más grandes.	Se puede realizar cualquier tipo de aplicaciones, desde las de escritorio hasta las móviles, pero está enfocado en aplicaciones web.	Se centra principalmente en el desarrollo de proyectos de Java
Lenguajes soportados	Java, C++, JavaScript, Python, PHP	Java, C, C++, JavaScript, Python, PHP, Kotlin, Dart, Go	Java, C++, C, JavaScript, Python, PHP

Fuente: Elaboración propia.

Anexo 8. Encuesta realizada a especialistas del área de informatización.

Objetivo: recopilar información sobre el proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI, a partir del diseño e implementación del software propuesto.

Estimado especialista:

Estamos realizando una encuesta con el propósito de recopilar información sobre el proceso de gestión de los medios tecnológicos del departamento de tecnología de la Facultad 4 de la UCI, a partir del diseño e implementación del software propuesto. Le sugerimos que contribuya con nosotros.

Muchas gracias.

6. La metodología utilizada para el diseño e implementación del software propuesto es la más adecuada.

Si__ No__ ¿Por qué? _____

7. Consideran usted que el software propuesto posee las características necesarias para ser utilizado con el fin que se creó.

Si__ No__ ¿Por qué? _____

8. El diseño e implementación del software resulta beneficioso para el cumplimiento de los objetivos propuestos.

Si__ No__ ¿Por qué? _____

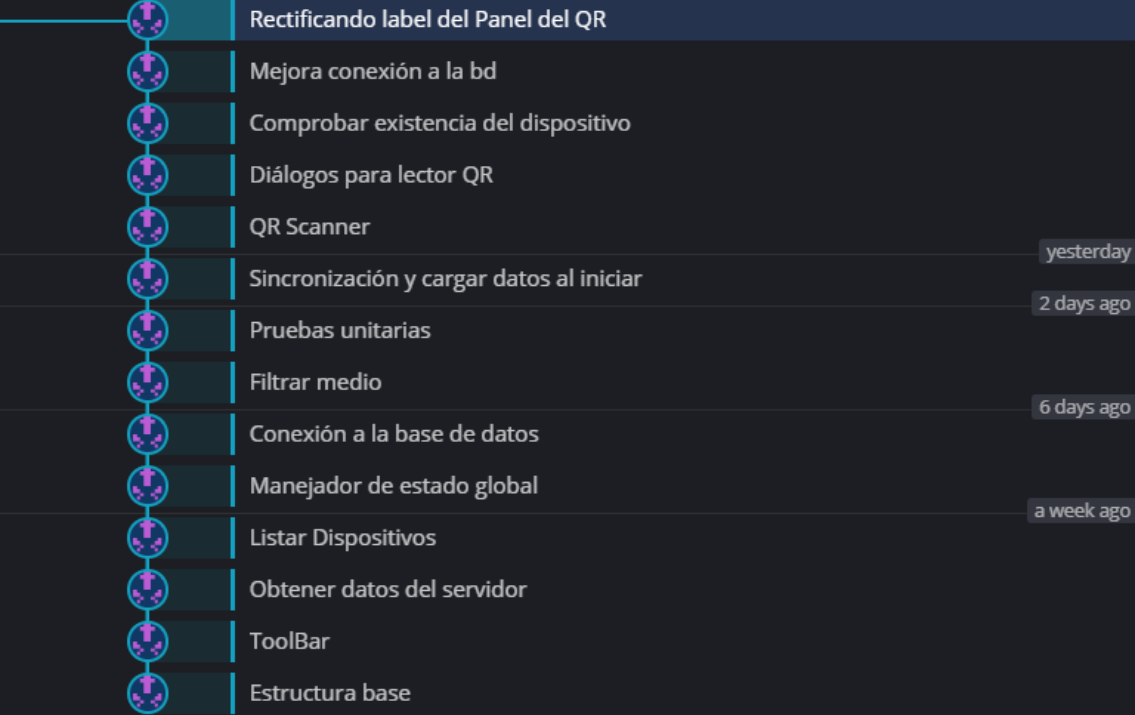
9. Considera usted que el diseño de la interfaz gráfica del software constituye el más adecuado para esto.

Si__ No__ ¿Por qué? _____

10. Como considera usted el diseño e implementación del software propuesto.

Adecuado__ Muy Adecuado__ Poco Adecuado__ No Adecuado__

Anexo 9. Control de versiones. Aplicación móvil.



7811111	Rectificando label del Panel del QR	
7811111	Mejora conexión a la bd	
7811111	Comprobar existencia del dispositivo	
7811111	Diálogos para lector QR	
7811111	QR Scanner	yesterday
7811111	Sincronización y cargar datos al iniciar	2 days ago
7811111	Pruebas unitarias	
7811111	Filtrar medio	6 days ago
7811111	Conexión a la base de datos	
7811111	Manejador de estado global	a week ago
7811111	Listar Dispositivos	
7811111	Obtener datos del servidor	
7811111	ToolBar	
7811111	Estructura base	

Ilustración 18. Control de versiones apk.
Fuente Elaboración propia.

Anexo 10. Control de versiones. Aplicación cliente.

	Eliminando serialNumber	
	Local por ip	
	Actualizar optencion de datos de los componentes ram y motherboar	
	Corrigiendo errores a la hora de obtener datos	
	Obtener datos mediante librería	
	Obtener datos pc mediante consola	
	Coger el numero de serie de la variable de entorno	3 weeks ago
	Chequeo que el dispositivo existe	3 months ago
	Envió de datos	
	Obtener los datos de la PC	
	Configuration del cliente http	
	Estructura Inicial	

Ilustración 19. Control de versiones des.
Fuente Elaboración propia.

Anexo 11. Control de versiones. Aplicación web.

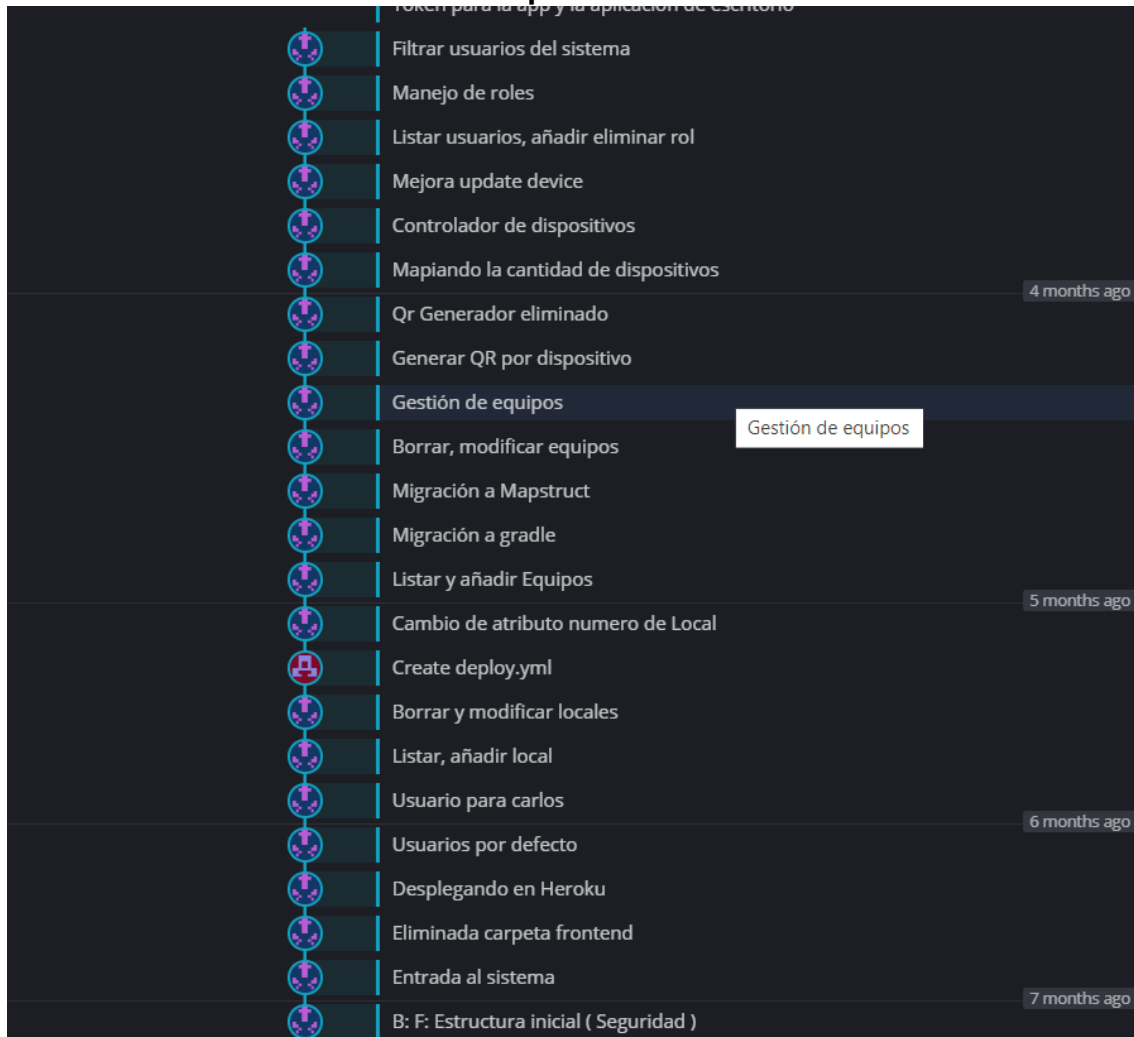


Ilustración 20. Control de versiones web.

Fuente Elaboración propia.