



Universidad de las Ciencias Informáticas
Facultad de Ciencias y Tecnologías Computacionales

“Módulo Control de Calidad de la aplicación Sis_Control para la Liberación del Plasma”

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor(es): Iván Lázaro Isa Regueiro

Tutor(es): Ing. Enmanuel Cristian De La Cruz Mora

Consultante: Ing. Alberto López González

La Habana, noviembre de 2023

Año 65 de la Revolución

Frase



**“Los sueños verdaderamente importantes son los
que tienes cuando estás despierto”**

David Lynch

DECLARACIÓN DE AUTORÍA

*Declaro ser el único autor de la presente tesis “**Módulo de la aplicación Sis_Control para el Control de Calidad en la Liberación del Plasma**” y autorizo a la UEB SH+ y a la Empresa AICA uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo de la aplicación resultante.*

Para que así conste firmo la presente a los _____ días del mes de _____ del año _____.

Iván Lázaro Isa Regueiro

Firma del Autor

Enmanuel Cristian De La Cruz Mora

Firma del Autor

Alberto López González

Firma del Consultante

DATOS DE CONTACTO

Nombre: Enmanuel Cristian de la Cruz Mora

Formación: Ingeniero en Ciencias Informáticas

Lugar de trabajo: Universidad de las Ciencias Informáticas, Centro de Desarrollo de Software Libre (CESOL)

Correo: ecdelacruz@uci.cu

Nombre: Alberto López González

Formación: Ingeniero en Informática

Lugar de trabajo: Laboratorios AICA +

Correo: alberto89@aica.cu

AGRADECIMIENTOS

A mis padres, sin necesidad de porqué.

A mi familia, y a los que no lo son los considero como tal.

A mi hermana y mi primo, por estar siempre apoyando y preguntando por la tesis.

A mi tutor por todo su tiempo entregado para la realización de este trabajo de diploma

A mis amigos, que han borrado las quejas de mis días.

A todos mis profesores, de todos estos años.

A todos mis compañeros de trabajo que soportaron en el día a día que los molestara en más de 5 ocasiones en el día para aclarar mis dudas

DEDICATORIA

Quiero dedicar este logro a ustedes que con tanto sacrificio emprendieron un largo camino para que mis sueños se realizaran, para que de proyectos se obtuvieran frutos. Se los dedico porque un día preguntaron ¿quieres seguir con viejas costumbres, o quieres hacer historia? A ustedes que son mis padres le dedico este trabajo de diploma, con todo mi cariño y admiración. Los quiero.

RESUMEN

La UEB SH+ “Suero y Hemoderivados” tiene como objetivo la producción de productos Sueros y hemoderivados de la sangre; cumpliendo con buenas prácticas de fabricación, certificada para la producción nacional y exportación de productos genéricos biosimilares e innovadores en las formas farmacéuticas. El presente trabajo de diploma propone informatizar los procesos de liberación del plasma en dicha UEB. A su vez, la aplicación que se propone debe facilitar el flujo de trabajo en la gestión de los registros analíticos, por una aplicación web que agilice el flujo de trabajo del laboratorio disminuyendo el tiempo empleado en el procesamiento y consolidación de los datos, facilite la obtención de los datos de interés, eleve los niveles de composición de la información, controle la gestión desde su inicio del proceso de liberación , y garantice su disponibilidad durante el proceso de producción de productos. Para el desarrollo de la propuesta de solución se emplea como modelo de procesos de desarrollo de software Extreme Programming (XP) como método ágil para la identificación y consecución de los requerimientos. También, se emplea PHP como lenguaje de programación, la infraestructura de creación de aplicaciones web Laravel 5.5 Framework, el gestor de bases de datos MySql, paquete de gestión Xampp y tecnologías de código abierto con reconocimiento en la comunidad de software libre.

Palabras claves: Analíticos, Calidad, Control; Gestión, Plasma, Registro, Productos.

ABSTRACT

The UEB SH+ “Serum and Blood Derivatives” aims to produce serum and blood derivative products; complying with good manufacturing practices, certified for the national production and export of biosimilar and innovative generic products in pharmaceutical forms. This diploma project proposes to computerize the plasma release processes in said UEB. In turn, the proposed application must facilitate the workflow in the management of analytical records, through a web application that streamlines the laboratory's workflow by reducing the time spent processing and consolidating data, facilitating obtaining the data of interest, raise the levels of information composition, control management from the beginning of the release process, and guarantee its availability during the product production process. For the development of the solution proposal, Extreme Programming (XP) is used as a software development process model as an agile method for identifying and achieving requirements. Also, PHP is used as a programming language, the Laravel 5.5 Framework web application creation infrastructure, the MySQL database manager, the Xampp management package and open source technologies with recognition in the free software community.

Keywords: *Keywords: Analytics, Quality, Control; Management, Plasma, Registration, Products.*

Índice

Introducción	1
Capítulo 1: Fundamentación teórica	6
1.1 Conceptos y definiciones asociados al dominio del problema.....	6
1.2 Objetivos estratégicos de la organización	7
1.3 Estudio de sistemas homólogos.	8
1.4 Metodología de desarrollo de software.....	12
1.5 Herramientas y tecnológicas actuales	14
1.5.1 Lenguaje	15
1.5.2 Marco de Trabajo	17
1.5.3 Entorno de desarrollo (Herramientas).	19
1.6 Conclusiones del capítulo.....	21
Capítulo 2: Análisis y Diseño de la Solución:.....	22
2.1 Fase: Exploración.....	22
2.1.1 Propuesta de Solución al módulo Control de Calidad	22
2.1.2 Especificación de requisitos	22
2.1.2.1 Requisitos Funcionales	22
2.1.2.2 Requisitos no funcionales	23
2.1.3 Validación de requisitos	24
2.2 Fase: Planificación.	25
2.2.1 Historias de usuarios.....	25
2.2.2 Estimación del tiempo	26
2.2.3 Plan de iteraciones.....	27
2.2.5 Plan de entrega	29
2.3 Fase 3: Diseño	29
2.3.1 Patrón MVC.....	31
2.3.2 Patrones de Diseño.....	33
2.3.3 Prototipo interfaz de usuario	34

2.3.4 Tarjetas CRC.....	36
2.3.5 Modelo de Datos	37
2.4 Conclusiones del capítulo.....	38
Capítulo 3: Implementación y Pruebas de la Solución.....	39
3.1 Estándares de codificación.....	39
3.2 Diagrama de Despliegue	40
3.3 Pruebas de una Webapp.....	42
3.4. Pruebas unitarias.....	43
3.4.1. Resultados de las pruebas unitarias.	44
3.5. Pruebas de interfaz de usuario.....	46
3.5.1. Resultados de las pruebas de interfaz de usuarios.....	49
3.6. Pruebas de rendimiento.	50
3.6.1. Resultados de las pruebas de rendimiento.	50
3.7. Pruebas de aceptación.....	52
3.8 Conclusiones del capítulo.....	53
Conclusiones Finales.....	54
Recomendaciones	55
Referencias Bibliográficas	56
Anexos.....	59
Anexos Historias de Usuarios.....	59
Anexos Prototipos de Interfaz de Usuario	62

Índice de Tablas

Tabla 1: Análisis Sistemas Homólogos.....	12
Tabla 2: Requisitos Funcionales.....	22
Tabla 3: Historia de Usuario: Autenticar usuario.....	25
Tabla 4: Historia de Usuario: Página principal muestra un resumen	26
Tabla 5: Historia de Usuario: Visualizar inventario del plasma	26
Tabla 6: Estimación del tiempo de las Historias de Usuarios	27
Tabla 7: Plan de Iteraciones	28
Tabla 8: Plan de Entrega	29
Tabla 9: Tarjeta CRC (MINSAP).....	36
Tabla 10: Tarjeta CRC (PCR General)	36
Tabla 11: Tarjeta CRC (Aprobaciones Banco).....	36
Tabla 12: Lista de Rutas de la Funcionabilidad testDeleteTipoPlasma	44
Tabla 14: Caso de prueba unitaria de la primera ruta.....	45
Tabla 15: Caso de prueba unitaria de la segunda ruta	45
Tabla 15: Caso de prueba de interfaz de usuario: Autenticar usuarios.	48
Tabla 16: Resultados del Caso de prueba de interfaz de usuario: Autenticar usuarios.....	48
Tabla 16: Caso de prueba aceptación: Visualizar Página Principal.....	53
Tabla 18: Historia de Usuario: Visualización de la realización del proceso	59
Tabla 19: Historia de Usuario: Visualizar plasma sin muestreo	59
Tabla 20: Historia de Usuario: Generar informe de banco de sangre sin muestreo	59
Tabla 21: Historia de Usuario: Visualizar plasma sin PCR con muestreo.....	60
Tabla 22: Historia de Usuario: Generar Informe de Plasma sin PCR con Muestreo.....	60
Tabla 23: Historia de Usuario: Visualizar muestreo del plasma.....	61
Tabla 24: Historia de Usuario: Adicionar muestreo del plasma.	61

Índice de ilustraciones

Ilustración 1: Etapas de la Metodología Ágil XP.	14
Ilustración 2: Patrón de Diseño Modelo-Vista-Controlador.	31
Ilustración 3: Prototipo de interfaz de usuario del tipo Evolutivo: Autenticar Usuario.....	34
Ilustración 4: Prototipo de interfaz de usuario del tipo Evolutivo: Pantalla Principal	35
Ilustración 5: Prototipo de interfaz de usuario del tipo Evolutivo: Inventario del Plasma	35
Ilustración 6: Diagrama de Relación de Datos (Diseñador de phpMyAdmin)	37
Ilustración 7: Ejemplo de estilo lowerCamelCase25	40
Ilustración 8: Diagrama de Despliegue (Elaboración Propia).....	41
Ilustración 9: Gráfica de los resultados de las pruebas unitarias.	46
Ilustración 10: Gráfica de los resultados de las pruebas de interfaz de usuario	49
Ilustración 11: Gráfica del resultado del tiempo de general informes desde el modulo	51
Ilustración 12: Gráfica del resultado del tiempo de general informes desde el office	52
Ilustración 13: Prototipo de interfaz de usuario del tipo Evolutivo: Muestro del Plasma	62
Ilustración 14: Prototipo de interfaz de usuario del tipo Evolutivo: Adicionar Muestro del Plasma	62
Ilustración 15: Prototipo de interfaz de usuario del tipo Evolutivo: Revisiones de PCR	63
Ilustración 16: Prototipo de interfaz de usuario del tipo Evolutivo: Liberación Analítica.....	63

Introducción

La UEB SH+ “Sueros y Hemoderivados” es una de las 5 UEB que integran la empresa Laboratorios AICA; perteneciente a la O.S.D.E de la industria Biotecnológica Farmacéuticas (BioCubaFarma). La misma radica en Ave. 51 # 33235 entre 332 y 334 Arroyo Arenas, La Lisa, provincia La Habana. Su objetivo social es entregar salud y calidad de vida, a través de la producción de medicamentos genéricos de avanzada, en las formas farmacéuticas sueros y hemoderivados; con la excelencia, responsabilidad y sostenibilidad, que les consolida una imagen de confianza frente al mercado y la sociedad.

Está constituida por varios departamentos, entre estos se encuentran el departamento Aseguramiento y Mejoramiento de la Calidad. Este departamento en su conjunto maneja la mayor información referida a los muestreo y análisis para su posterior liberación de las materias primas e insumos para la producción, por lo que se decidió elaborar un sistema integral que permita gestionar toda la información generada en el proceso de la liberación del plasma.

La UEB por la complejidad de sus operaciones y de las materias primas con las que trabaja carece de un sistema de control informático que les permita reducir los errores humanos en el tratamiento de la información y las no conformidades encontradas por el CEDME organismo regulador de la industria biofarmacéutica cubana. El plasma materia prima fundamental en la producción de Hemoderivados de la Sangre, presenta complejidades específicas, en cuanto a su manipulación y almacenamiento. El proceso para llegar al producto final cuenta con un grupo subproceso diferentes, que comienza desde la información de los análisis de la sangre adquirida en los diferentes bancos del país la cual tiene diferentes categorías en dependencia de los resultados, la misma llega en la mayoría de los casos en una hoja escrita a mano. Cuando está arriba a la UEB pasa por el proceso de inspección y muestreo por parte de los inspectores de calidad, dándole un código de muestreo específico a las bolsas dependiendo del banco de sangre de donde provienen. Para luego ser almacenadas en las cámaras frías de (2 a 8°C) estas también llevan un control de la temperatura para controlar la cadena de frío del plasma. Para liberar la materia prima y comenzar la producción antes debe pasar por otro proceso de análisis en el laboratorio. Todo el proceso genera un volumen de registro e informaciones con difícil acceso y control, provocando de esta forma pérdidas de información valiosa y necesaria para todo el proceso productivo de la UEB.

La sociedad actual es parte indisoluble del constante y acelerado proceso de desarrollo científico y tecnológico apreciado en todos los campos de la vida diaria del hombre, desarrollo que brinda un mayor nivel de efectividad, eficiencia y competitividad a todos los procesos enmarcados en las distintas organizaciones. Las organizaciones se enfrentan al reto de aumentar su agilidad de innovación y respuesta a los constantes cambios en los negocios y a

los ambientes tecnológicos heterogéneos con la misión de reducir los gastos en sus Tecnologías de la Información (TI).

Como parte de todo este proceso de evolución se encuentra la industria informática, la cual proporciona nuevas soluciones que han logrado satisfacer las necesidades de muchas empresas mejorando su funcionamiento y productividad. En la actualidad, la mayoría de las empresas poseen sistemas informáticos que soportan distintos niveles de aplicación de la tecnología, destacando, entre otros, los procesos de gestión, comunicación y procesos para el apoyo a la toma de decisiones. (1)

Las Tecnologías de la Información y las Comunicaciones (TIC) se han constituido en uno de los recursos más importantes de la sociedad, trayendo como consecuencia una explosión exponencial en la transmisión e intercambio de datos, información y conocimientos, a los cuales se puede acceder sin tener en cuenta barreras geográficas o limitaciones del tiempo. Es por ello que en los últimos años casi todos los países del mundo han establecido e implementado proyectos, políticas y estrategias para promover el uso de las TIC y aprovechar los beneficios y los aportes que estas ofrecen. (2)

Hoy día es imposible concebir una empresa exitosa sin el apoyo de las TIC para administrar sus procesos de negocio. Las vías por las cuales las TIC ayudan a mejorar el desempeño en las empresas son cuatro: automatización, accesibilidad a la información, costos de transacción y procesos de aprendizaje. (3)

El continuo desarrollo de la ciencia y la técnica unido al crecimiento de la competitividad en el mercado ha obligado a las organizaciones a controlar su negocio en busca de un aumento en la eficiencia de todos sus procesos. Las organizaciones deben estar capacitadas para dar respuestas inmediatas a los constantes cambios en los negocios y a los ambientes tecnológicos heterogéneos teniendo control del impacto de gastos en sus TI.

Para ello necesitan contar con herramientas que les ayuden en la planificación, producción y control de sus productos, para cumplir con los estándares de calidad, teniendo la capacidad de recomendar acciones preventivas para evadir errores que pudiesen ocurrir y llevar un control sobre los recursos. A nivel mundial el sector farmacéutico realiza permanentes esfuerzos competitivos para introducir nuevos productos al mercado sobre la base de investigación y desarrollo (i+d) generando fuertes gastos de comercialización, control de la producción y publicidad. Una de las claves del logro de las empresas farmacéuticas en descubrir nuevos medicamentos es un programa intenso de investigación y desarrollo. La invención de nuevas medicinas es un proceso de gran complejidad, que depende altamente de un intenso programa de i+d. De hecho, la industria farmacéutica es uno de los sectores más dinámicos e innovadores de todas las áreas empresariales a nivel mundial. (4)

Esta industria es un mercado complejo y dinámico, que exige de sus participantes niveles muy altos de calidad y de exactitud. Para crecer de forma exitosa y optimizar la productividad, las compañías del sector necesitan elevar al máximo la eficiencia de todos sus procesos y mejorar la calidad de sus productos. Las empresas pequeñas, medianas o grandes en este campo que buscan potenciar y fortalecer sus negocios, realizan enormes esfuerzos por reducir al mínimo posible los errores y las equivocaciones, especialmente en todo lo vinculado al proceso de producción. (5)

Partiendo de esta situación problemática surge el siguiente **problema de investigación**: ¿Cómo controlar la trazabilidad de los diferentes tipos de bolsas de plasma? Se define como **objeto de estudio**: la gestión de los procesos de control de la calidad en la liberación del plasma, enmarcado en el **campo de acción**: los módulos de Control de la Calidad para la liberación del plasma. Por lo que se formula como **objetivo general** desarrollar un módulo de la aplicación Sis_Control que permita la gestión de la información generada durante el proceso de liberación del Plasma para el control de la calidad.

Para acometer el objetivo general se trazaron los siguientes **objetivos específicos**:

- ✓ Realizar el estudio teórico y el del arte sobre módulos de control de la Calidad.
- ✓ Diseñar la solución propuesta.
- ✓ Implementación y validación de la solución diseñada.

Para dar cumplimiento a estos objetivos específicos se identificaron las siguientes **tareas de investigación**:

- Análisis de los principales conceptos relacionados con la investigación.
- Investigación sobre los sistemas homólogos y las herramientas a utilizar para el desarrollo de la solución.
- Definición de los requisitos funcionales y no funcionales.
- Implementación de las funcionalidades de la herramienta.
- Aplicación de las pruebas a la herramienta desarrollada.

Entre los métodos de investigación científica utilizados se destacan:

Técnica:

La tormenta de idea fue el método empírico utilizado en el proceso de investigación para la determinación de las restricciones de *software* e implementación que se debe tener en cuenta en el proceso de desarrollo de la herramienta.

Métodos científicos:

El método **inductivo-deductivo** permite llegar a proposiciones generales a partir de hechos aislados que confirman la teoría o a partir de estas teorías arribar a conclusiones sobre casos particulares que se verifican en la práctica. (6) Este método fue usado para, a partir del análisis de las herramientas existentes que realizan la gestión de repositorios, seleccionar las características que va a poseer la herramienta a desarrollar de forma tal que realice correctamente la gestión de los repositorios.

El método **histórico-lógico** plantea que se debe estudiar la trayectoria real de los fenómenos y acontecimientos en el transcurso de una etapa o período analizando las leyes generales del funcionamiento y desarrollo del fenómeno, estudiando su esencia. (6) Fue empleado con el objetivo de verificar cómo evolucionan teóricamente las herramientas que realizan la gestión de productos farmacéuticos y de este modo poder realizar una selección de las técnicas y herramientas que se van a utilizar para desarrollar.

El método **analítico-sintético** consiste en la extracción de las partes de un todo, con el objetivo de estudiarlas y examinarlas por separado para luego sintetizarla y tomar los elementos más importantes. Este método permitió el estudio de diferentes fuentes bibliográficas para extraer los elementos más importantes que se relacionan con las herramientas para la administración de productos farmacéuticos.

El método **revisión bibliográfica** permite identificar las fuentes arbitradas¹, identificar y comprender las ideas principales (7) y sintetizar el argumento central de cada sección a través de la supresión y generalización de ideas principales. Dicho método fue utilizado para determinar las fuentes y referencias bibliografías óptimas y actualizadas para la elaboración de la investigación.

Capítulo 1: Fundamentación Teórica

Este capítulo centra su atención en el estudio del objeto y el flujo detallado de los procesos. Se analizaron sistemas informáticos utilizados en la gestión de procesos productivos. A su vez, se valoran las tecnologías y la metodología a utilizar en el desarrollo de la aplicación.

Capítulo 2: Análisis y Diseño de la Solución:

En este capítulo se determinan los servicios que brindará la herramienta, definiéndose las funcionalidades que debe cumplir, así como el diseño y se generan los artefactos que propone la metodología.

Capítulo 3: Implementación y Pruebas de la solución propuesta

Este capítulo abarca todo lo relacionado con la implementación de la herramienta y el proceso de pruebas utilizado. Se implementan todas las funcionalidades identificadas, logrando una herramienta que permita cumplir el objetivo general de los estudios que se presentan. Se detallan también, las pruebas que se le realizaron a la herramienta ya finalizada, con el objetivo de asegurar la eficiencia de la solución.

Capítulo 1: Fundamentación teórica

El presente capítulo centra su atención en el objeto de estudio y el flujo detallado de los procesos. Se analizaron sistemas informáticos utilizados en la gestión de procesos productivos y envasado de lotes de productos. Finalmente se valoran las tecnologías y la metodología a utilizar en el desarrollo de la aplicación.

1.1 Conceptos y definiciones asociados al dominio del problema

Plasma: El plasma es un líquido transparente y ligeramente amarillento que representa el 55 % del volumen total de sangre. En él se encuentran suspendidas las células sanguíneas: **glóbulos rojos, glóbulos blancos y plaquetas**. Está formado por agua (90%), sales minerales y una gran cantidad de proteínas que velan por el buen funcionamiento de nuestro organismo, entre las que destacan las siguientes: (8)

- Las **inmunoglobulinas**, defensas que nos protegen de las infecciones.
- Los **factores de coagulación**, que son responsables, junto con las plaquetas, de detener el sangrado cuando nos cortamos.
- La **albúmina**, una proteína que transporta hormonas y fármacos y los lleva hasta los tejidos.

De los 5 litros de sangre que tiene de media una persona de 70 kilos, **unos 3 litros son de plasma**. (8)

Tipos de Plasma: Plasma Normal Donación, Plasma Normal por Plasmaféresis, Plasma Anti-Tetánico, Plasma Anti-D, Plasma Anti-HB.

Hemoderivados: Los hemoderivados registrados constituyeron un grupo particular y diferenciado dentro de las especialidades farmacéuticas. Conceptualmente se entiende por hemoderivados aquellas especialidades farmacéuticas cuyo principio activo proviene del plasma de donantes humanos sanos a través de un proceso de fraccionamiento y purificación adecuado.

Gestión y Control:

Gestionar es un proceso social que implica responsabilidad por la planificación efectiva y económica y regulación de las operaciones de una empresa, en cumplimiento de un determinado propósito o tarea. Esa responsabilidad comprende: (a) juicio y decisión al determinar planes y al usar datos para controlar desempeño y el progreso con respecto a los

planes; y (b) la guía, integración, motivación y supervisión del personal que compone la empresa y lleva a cabo las operaciones. (9)

Gestionar es simplemente el proceso de toma de decisiones y control sobre las acciones de seres humanos con el expreso propósito de lograr metas predeterminadas. (10)

La gestión está caracterizada por una visión más amplia de las posibilidades reales de una organización para resolver determinada situación o arribar a un fin determinado. Puede asumirse, como la disposición y organización de los recursos de un individuo o grupo para obtener los resultados esperados. Pudiera generalizarse como una forma de alinear los esfuerzos y recursos para alcanzar un fin determinado.

Los sistemas de gestión han tenido que irse modificando para dar respuesta a la extraordinaria complejidad de los sistemas organizativos que se han ido adoptando, así como a la forma en que el comportamiento del entorno ha ido modificando la manera en que incide sobre las organizaciones.

Una de las características y ventaja de un sistema de gestión es la posibilidad de instaurar un control para vigilar el estado de una empresa, con el fin de elaborar acciones que garanticen un correcto funcionamiento de la misma. La mayoría de los autores definen el control como un proceso de vital importancia ya que establece medidas para corregir y evadir anomalías que impidan un correcto funcionamiento en la organización.

La Industria Farmacéutica:

La industria farmacéutica es un importante elemento de los sistemas de asistencia sanitaria de todo el mundo, está constituida por numerosas organizaciones públicas y privadas dedicadas al descubrimiento, desarrollo, fabricación y comercialización de medicamentos para la salud.

Su fundamento es la i+d de medicamentos para prevenir o tratar las diversas enfermedades y alteraciones. Los modernos avances científicos y tecnológicos aceleran el descubrimiento y desarrollo de productos farmacéuticos innovadores dotados de mejor actividad terapéutica y menos efectos secundarios. (11)

1.2 Objetivos estratégicos de la organización

La UEB SH+ define como parte de su **misión** “Entregamos salud y calidad de vida, a través de la producción de medicamentos genéricos de avanzada, en las formas farmacéuticas de sueros y hemoderivados; con la excelencia, responsabilidad y sostenibilidad, que nos consolida una imagen de confianza frente al mercado y la sociedad”.

Por otro lado, en la **visión** se expone que “Somos una UEB que cumple con las Buenas Prácticas de Fabricación, certificada, para la producción nacional y la exportación de

medicamentos genéricos biosimilares e innovadores en las formas farmacéuticas de sueros y hemoderivados”.

Como parte de la política de calidad de la Empresa Laboratorios AICA a la que pertenece la UEB está el producir medicamentos genéricos de avanzada para uso humano, en la forma farmacéutica sueros y hemoderivados teniendo como principio suministrar un producto que brinde salud y calidad de vida, cumpliendo consistentemente los requisitos de calidad, fiabilidad y seguridad para el mercado. A su vez, se hace uso del Sistema de Gestión de la Calidad, Certificado por la Norma de Calidad ISO 9001:2008, cumpliendo con las regulaciones internacionales y con las Buenas Prácticas de Fabricación de Productos Farmacéuticos y de Laboratorio, según lo establecido en el ámbito regulatorio para la producción de medicamentos estériles e implementando un Sistema de Gestión Integrado.

1.3 Estudio de sistemas homólogos.

Software de gestión para laboratorios farmacéuticos y visitantes médicos.

S-ERP Online Laboratorios farmacéuticos es un software de gestión enfocado a empresas de producción y distribución de medicamentos en España, especialmente diseñado para las necesidades concretas del sector farmacéutico. Este ERP Online permite la gestión de todas las áreas de negocio vinculadas al sector salud de una forma sencilla e intuitiva.

La aplicación se sirve de una manera totalmente online bajo el modelo Software como Servicio. Es escalable, permitiendo el pago de licencias por usuarios y mes/año, lo que evita la compra de producto y el mantenimiento del mismo. (12)

Funcionalidades incluidas en el ERP: (12)

1. Software CRM para la gestión de doctores, centros, visitantes médicos.
2. Emisión de talonarios de recetas automatizado.
3. Control de la producción de medicamentos con completa trazabilidad.
4. Control logístico completo: salidas de mercancía, rutas, albaranes, pedidos.
5. Facturación automatizada y emisión de remesas.
6. Informes completos del estado del negocio, panel de control.
7. Entorno colaborativo. Derivación de tareas, *workflows*, alertas.
8. Flujos de trabajo adaptados al proceso de negocio.
9. Gestión de incidencias según normas de calidad vigentes.

La herramienta permite su completa personalización y la realización de desarrollos sobre la misma. Creando un software a la medida del negocio.

Sistema de liberación de GNL por control de calidad (KROHNE L-QRS)

KROHNE L-QRS es un sistema software de medida de la calidad para la liberación en tiempo real de cargas de GNL¹; utiliza medidas en línea y cálculos estadísticos certificados, incluyendo cálculos de la calidad y de valores atípicos. Asegura la disponibilidad instantánea del certificado de calidad, evitando los altos costes ligados al manejo de muestras, pruebas de laboratorios y correcciones retrospectivas del conocimiento de embarque y los certificados de calidad.

El sistema está certificado por el NMI², el instituto independiente acreditado a nivel internacional, y es conforme a ISO 8943, GPA 2172, ASTM 4784 y GIIGNL. Combina las funciones de gestión de analizadores y adquisición de datos (AMADAS), utilizando técnicas de elaboración de gráficos de control y el control del proceso estadístico para determinar el rendimiento, la disponibilidad y la capacidad de mantenimiento de analizadores de procesos y otros equipos críticos, a fin de asegurar los mejores resultados y disponibilidad constante. (13)

Las características y opciones del producto dependen de la configuración del dispositivo: Póngase en contacto con nosotros para asegurarse de que sus funciones preferidas pueden combinarse en un solo dispositivo. (13)

- Asegura una disponibilidad instantánea del certificado de calidad y del conocimiento de embarque, evitando el manejo de muestras y los ensayos offline de laboratorio
- Conformidad indiscutible con los acuerdos GSPA (acuerdos de compra y venta de gas), estándares internacionales como ASTM D3764, ASTM D6299, OP 97-30425 y reglamentos locales
- Informes, evaluaciones estadísticas y cálculo del rendimiento / KPI
- Totalmente auditable y trazable, con informes multinivel, en todos los analizadores
- Detección de valores atípicos en el poder calorífico por unidad de masa según ASTM E-178.

¹ Gas Natural Licuado

² Instituto Holandés de Metrología

- Validación automatizada de equipos y monitorización del rendimiento real de equipos de medida de la calidad (QMI)
- Mantenimiento de analizadores altamente eficiente mediante técnicas avanzadas de elaboración de gráficos de control
- Control del proceso estadístico (SPC) con reglas de decisión integradas (ASTM D6299)
- Gestión de tendencia y alarmas
- Integración con sistemas de control y gestión de plantas (DCS, PI, LIMS, ERP, etc.)

MES (Manufacturing Execution System):

Este sistema de control de procesos por lotes está diseñado para las necesidades particulares de los procesos por lotes. Los módulos integrados *Plant Acquis iT* y *Plan Material iT* garantizan la adquisición de los datos de producción referentes a órdenes y lotes y la gestión de materiales relacionados con el proceso, con un coherente seguimiento de lotes. Dentro de sus ventajas se encuentran: (14)

- Sistema por lotes implantado en el sector real
- Gestión de materiales integrada con trazabilidad de lotes
- Registro electrónico de lotes.
- Gestión de órdenes integradas y de recetas.
- Sistema de control de procesos libremente escalable y extensible
- Transparencia del proceso hasta el nivel de control del proceso
- Sistema de control de procesos orientado a objetos, abierto y basado en PLC³.
- Interfaz de usuario intuitiva, diseñada según los últimos estándares de usabilidad.
- Librería estándar basada en gráficos de proceso vectoriales inteligentes
- Extensos módulos con librerías de clases de automatización predefinidas.
- Diseñador de pantallas de proceso para la creación de las mismas.

³ (Control Lógico Programable) es un equipo comúnmente utilizado por aquellas industrias que buscan dar un salto significativo en la automatización de todos sus procesos.

- Diseñador de informes de producción para la creación de los mismos.
- Basado en software estándar (ej. base de datos y sistemas operativos).

MISTRAL:

Es un Sistema especializado en Logística de almacenes, importaciones, Gestión de Materia Prima, Taller y Flota que es comercializado en Cuba por una entidad no cubana, está totalmente integrado al SISCONT5⁴ evitando la duplicidad de datos. Tecnomática brinda servicio de implementación y soporte técnico para este sistema. No cuenta con una versión web. (15)

Trabaja con diferentes módulos como son:

- Sistema Gestión de Usuarios
- Gestión de Flotas. FLEET
- Gestión de Almacenes (Insumos, Materias Primas, Producto Terminado)
- Gestión de Talleres
- Gestión de Contratos de Importaciones IMPORT

Como parte de la investigación realizada en el ámbito nacional no se encontró referencia documentada de sistemas web similares existentes. Si se encontró una referencia de la aplicación de escritorio Mistral, la cual es usada en la empresa **AICA+** y todas sus **UEB**⁵. Sin embargo, la aplicación no cumple con los requisitos específicos del proceso de la liberación del plasma.

Análisis de las principales aplicaciones para la gestión de producción

⁴ Es un software de Contabilidad y Finanzas, doble moneda, con Gestión de Tesorería, Gestión de Caja Chica, Gestión de Créditos y Cobranzas

⁵ Unidad Empresarial de Base

A continuación, se muestra una tabla donde se realiza una comparación entre las aplicaciones que realizan la gestión de producción y se tiene en cuenta como criterio de comparación sus características fundamentales.

Tabla 1: Análisis Sistemas Homólogos

Características	Mistral	MES	KROHNE L-QRS	ERP
Adquisición gratis	No permite	No permite	No permite	No permite
Capacitación del personal gratis	No permite	No permite	No permite	No permite
Especificaciones requeridas del proceso	No Cumple	No Cumple	No Cumple	No Cumple
Aplicación Web	No	Si	Si	Si
Mantenimiento y soporte gratis	No Cumple	No Cumple	No Cumple	No Cumple

Las aplicaciones analizadas en la tabla anterior corresponden a la investigación de los sistemas homólogos. Dicha tabla arrojó que las aplicaciones Mistral, MES, KROHNE L-QRS, ERP no permiten su adquisición y capacitación de personal gratuitas, no cumplen con las especificaciones requeridas para el proceso de liberación del plasma, ni presentan un mantenimiento y soporte gratuito. La aplicación Mistral no es una aplicación web. Por lo antes expuesto se propone implementar un módulo de la aplicación web Sis_Control capaz de cumplir con los requerimientos y características de la entidad, eliminando la gestión manual y centralizando la información de la liberación del Plasma.

1.4 Metodología de desarrollo de software

A diferencia de las metodologías tradicionales, las ágiles se enfocan en la obtención rápida de una aplicación funcional, esto garantiza una flexibilidad adaptativa ante los cambios que pudieran surgir durante el proceso de desarrollo de la aplicación. Además, tienen amplia relevancia puesto que las interacciones con el cliente son más frecuentes, dándole un sentido de implicación y colaboración.

El Manifiesto Ágil se considera el punto de partida y surgió de la necesidad de trazar una definición común para todas las metodologías que cumplieran esos principios. Específicamente, el desarrollo ágil se centra en la existencia de equipos de trabajo multifuncionales con autonomía de decisión y no en grandes divisiones agrupadas por jerarquía y funcionalidad. Además, se basa en iteraciones cortas, de manera que el cliente

ofrezca su opinión sobre la calidad de la aplicación y se sienta participe en el proceso de creación.

Todos los modelos que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.
2. El software que funciona, frente a la documentación exhaustiva.
3. La colaboración con el cliente, por encima de la negociación contractual.
4. La respuesta al cambio, por encima del seguimiento de un plan. Con la misma se desea minimizar el impacto de las tareas que no son totalmente imprescindibles para conseguir el objetivo del proyecto. Se pretende aumentar la eficiencia de las personas involucradas en el proyecto y, como resultado de ello, minimizar el coste.

Extreme Programming (XP)

Se utiliza en proyectos a corto plazo. Se basa en pruebas unitarias, que se realizan a los principales procesos y a posibles fallas que pudieran ocurrir, también en la reutilización de código para lo cual se crean patrones o modelos estándares. Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Es una metodología ágil 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 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. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (16)

Lo esencial de esta metodología es llevar a cabo una buena interacción entre usuarios y desarrolladores. La simplicidad, al desarrollar y codificar los módulos del sistema, así como la retroalimentación, concreta y frecuente, del equipo de desarrollo, el cliente y los usuarios finales.

Las características fundamentales:

- Desarrollo iterativo e incremental.
- Pruebas unitarias.
- Programación en parejas.

- Fuerte integración del equipo de desarrollo con el cliente.
- Corrección de todos los errores antes de añadir una nueva funcionalidad.
- Refactorización del código.
- Simplicidad en el código

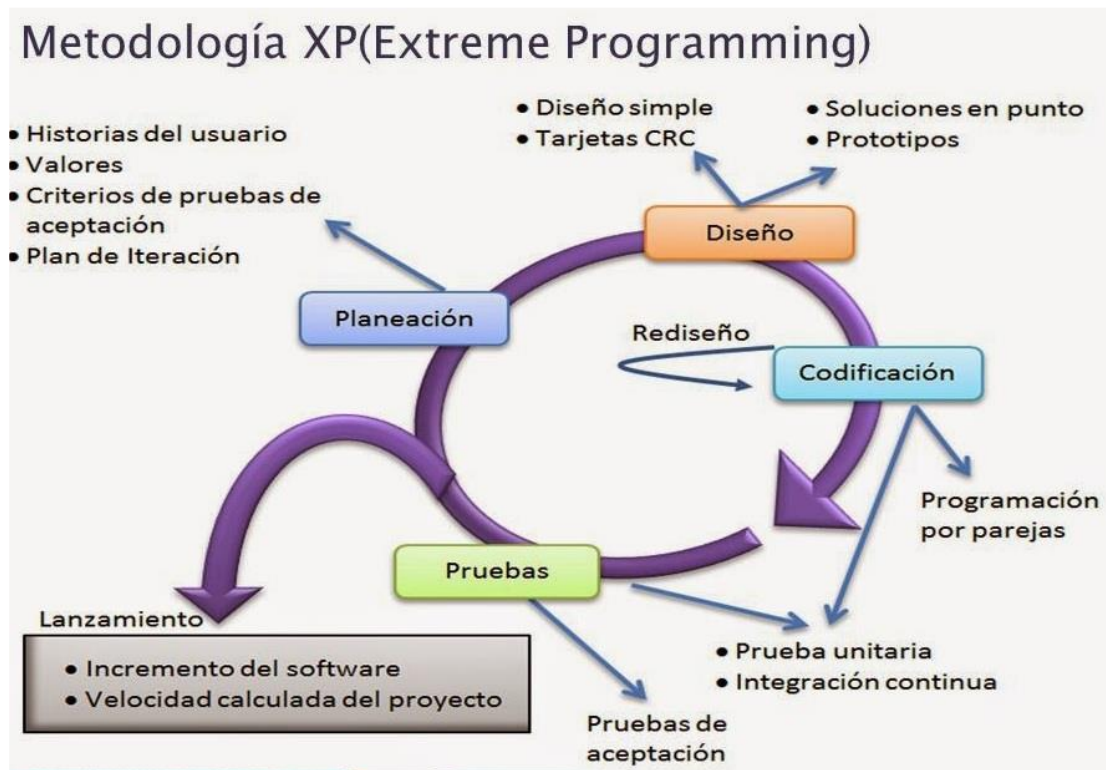


Ilustración 1: Etapas de la Metodología Ágil XP.

Se escoge la metodología ágil XP, porque es la que sigue el proyecto al cual se va integrar el módulo de **Control de la Calidad**

1.5 Herramientas y tecnológicas actuales

Las aplicaciones web sólo necesitan un navegador para ejecutarse, independientemente del sistema operativo, lo que las hace ligeras. Se alojan en un servidor al cual se accede de forma remota haciendo uso de los protocolos HTTP o HTTPS. Utilizando una arquitectura cliente-servidor, la información almacenada se encuentra en un solo lugar, lo que garantiza su

integridad. En este acápite se enumeran las características de las tecnologías y herramientas seleccionadas para el desarrollo del presente trabajo. Como el desarrollo de la solución es parte de la plataforma principal de trabajo de la UEB Laboratorio SH+ el perfil tecnológico de la propuesta de solución está sujeto a estos condicionamientos.

1.5.1 Lenguaje

Los siguientes lenguajes fueron seleccionados para el desarrollo, porque son los utilizados en el proyecto actual al que se va integrar el módulo Control de Calidad.

PHP

La principal causa que justifica la selección de PHP en su versión 5.6 como lenguaje de programación se basa en la necesidad de garantizar la inserción del módulo a la aplicación del Sistema integral que se está desarrollando con esta tecnología (17). Por otra parte, PHP permite la conexión a diferentes sistemas gestores de bases de datos, tales como: MySQL, PostgreSQL, Oracle, Microsoft SQL Server, etc. Este lenguaje cuenta con una amplia comunidad de desarrolladores y existe documentación disponible y gratuita. Además, presenta opciones útiles para el procesamiento de textos, como el análisis de documentos XML y JSON.

Entre las ventajas fundamentales del lenguaje de programación se encuentran:

- Tiene alta capacidad de conexión con la mayoría de los manejadores de base de datos empleados en la actualidad, especialmente con MySQL.
- Es de código abierto, lo cual hace que se presente como una alternativa de fácil acceso y empleo para todos.
- Permite el empleo de técnicas de Programación Orientada a Objetos.
- Es potente en el manejo de excepciones.
- Es multiplataforma, es decir que puede ser usado con Linux, el WIN32, o el Novel Netware.

HTML

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y

aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

JavaScript

JavaScript es lo que se conoce como lenguaje script, es decir: se trata de código de programación que se inserta dentro de un documento. JavaScript fue desarrollado por la empresa Netscape con la idea de potenciar la creación de páginas Web dinámicas para su navegador. (18)

La ventaja fundamental de JavaScript es que su aprendizaje y uso son muy sencillos y que permite realizar labores complejas en una página sin necesidad de aprender CGI.

La adopción de JavaScript para la aplicación web se debió a la facilidad de su uso y que permite dinamizar los contenidos mostrados.

Modelado.

El lenguaje de modelado *UML20⁶* en su versión 5, permite comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema, comprender mejor lo que se está construyendo y a la vez descubrir oportunidades de simplificación y reutilización. Se basa en el hecho de que un modelo es la simplificación de la realidad. (19)

Hojas de estilo en cascada (CSS).

Cascading Style Sheets (CSS) conocido en español como Hojas de Estilo en Cascada. Es un mecanismo simple que permite controlar la apariencia de una página web. Describe cómo se muestra un documento en la pantalla, cómo se va a imprimir o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos les permite a los desarrolladores web controlar el estilo y formato de múltiples páginas web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en el CSS afectará a todas las páginas vinculadas a ese CSS en las que aparezca ese elemento.

Tecnología AJAX

La necesidad de agilizar el desarrollo de la aplicación, con el objetivo de dinamizar el trabajo y lograr su disponibilidad en el menor tiempo posible y con la calidad requerida, hace que se

⁶ Lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

emplee AJAX (20). AJAX basa su funcionamiento en una comunicación asíncrona cliente - servidor. El lenguaje interpretado para ejecutar las funciones de llamada de AJAX es JavaScript, mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales; permitiendo que se procese los datos que deben ser actualizados en el cliente.

1.5.2 Marco de Trabajo

Framework de desarrollo

Un framework⁷ 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. (21)

Laravel 5.5

Se selecciona laravel 5.5 (22) como framework de trabajo puesto que es uno de los frameworks PHP de código abierto más conocidos dentro del mundo de la programación. Esto lo ha conseguido debido a su sencillez en la sintaxis de su código y facilidad para desarrollar proyectos de manera eficaz y rápida. Sus herramientas, aparte, la convierten en una de las mejores opciones para el desarrollo de las aplicaciones web. Es usado principalmente en el desarrollo backend o en la programación, pero también es uno de los principales frameworks PHP para las aplicaciones Full Stack. Esto quiere decir, que cada vez que se desarrollen funciones para que en una página los usuarios logren hacer login, o se programa cualquier otra capa de datos que no sea directamente visible para los usuarios, es probable que el programador web haya usado las bondadosas características de Laravel. Su popularidad radica en que este framework busca desarrollar código PHP de una manera más presentable y sencilla. Un desarrollador que use Laravel consigue descargar un paquete para añadir a una página funciones de autenticación de usuarios, acceso a bases de datos, validación de datos, entre otras cosas. Sus características para desarrolladores más resaltantes son: Su arquitectura es MVC (Modelo-Vista-Controlador), permite relacionar entre sí las partes de una

⁷ Es un framework de pruebas de JavaScript con múltiples funciones que se ejecuta en Node.js y en el navegador, haciendo que las pruebas asíncronas sean sencillas.

aplicación. Es una de las arquitecturas más habituales de los frameworks. Íntegras Laravel con plataformas de terceros o bibliotecas.

Bootstrap v5.3.1

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. (23) Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. (24)

Servidor Apache

Es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. (25) El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

El servidor HTTP Apache es un servidor web de código abierto compatible con sistemas operativos basados en Unix, Microsoft y MacOS. Este servidor tiene una licencia Apache que permite su distribución libre y la modificación del código fuente. Apache presenta una arquitectura modular, por lo tanto, se pueden instalar o desinstalar módulos dependiendo de las necesidades del usuario. Muchas aplicaciones web son diseñadas asumiendo que el ambiente donde serán desplegadas es un servidor Apache. (26)

Sistema de Gestión de Base de Datos

Un Sistema de Gestión de Base de Datos 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. (27)

1.5.3 Entorno de desarrollo (Herramientas).

PHPStorm

Es el IDE preferido por la mayoría de programadores. Al margen de todas las características habituales en los entornos de desarrollo. Es el único que ofrece una integración casi perfecta con Laravel gracias a su plugin para Laravel. (28)

A continuación, se resaltan otras de las características de esta herramienta:

- Compatibilidad.
- Permite la gestión de proyectos fácilmente.
- Proporciona un fácil autocompletado de código.
- Soporta el trabajo con PHP 5.5.
- Sintaxis abreviada.

Este IDE es uno de los entornos de programación más completos de la actualidad, permite editar código no solo del lenguaje de programación PHP como lo indica su nombre. Actualmente es compatible con Sistemas Operativos Windows, Linux y Mac OS X. Algo que destaca en PhpStorm es la ejecución del código en la misma interfaz del IDE. Así como también, la interpretación y visualización inmediata de código .php hasta en cinco de los navegadores web más populares. Con esto no habrá que cargar manualmente el navegador e ingresar la dirección url de la página.

MySql

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos

Para guardar la información persistente en la base de datos se selecciona MySQL, ampliamente utilizado y soportado por la mayoría de las plataformas. La selección de MySQL como SGBD está sujeto a que el módulo a desarrollar es parte de un sistema integral que se

está desarrollando y la base de datos debe ser modelada e implementada sobre este sistema gestor. MySQL es una solución viable para aplicaciones web donde la concurrencia es realmente baja, resultando mucho más frecuente la existencia de consultas para lectura de datos. A su vez, provee mayor velocidad al realizar las operaciones, lo cual hace que sea uno de los SGDB con mejor rendimiento, bajo costo en requerimientos para la elaboración de bases de datos y baja probabilidad de corromper datos, incluso si los errores no se producen en el propio SGDB, sino en la aplicación. Por otra parte, cuenta con una amplia comunidad de usuarios que le brindan soporte y proveen amplia documentación para la consulta de los diseñadores de las bases de datos en este SGBD. (29)

XAMPP Control Panel V3.3.0

XAMPP es un paquete de instalación de software libre, que consiste principalmente en el sistema de gestión de bases de datos como: MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. (30)

Ventajas de XAMPP:

- Es multiplataforma, es decir, está disponible para Windows, Linux y Mac.
- Viene con muchos otros módulos como OpenSSL, Media Wiki, Joomla, WordPress y muchos más.
- Está disponible en versiones estándar y completa.

Funciones de XAMPP:

XAMPP es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación), ejemplo en tu propia computadora sin necesidad de tener que acceder a internet.

Visual Paradigm en su versión 16.3, se selecciona como herramienta *CASE2*¹⁸ de modelado profesional (31), que utiliza *UML* para la completa representación de las etapas por las que transita un producto de *software*. Este permite la realización de una amplia gama de diagramas como: casos de uso, de actividades, de despliegue, entre otros, así como la generación de

⁸ Ingeniería de Software Asistida por Computadora (del inglés Computer Aided Software Engineering). Aplicaciones destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo del mismo en términos de tiempo y dinero

código fuente desde los mismos y la documentación asociada al proceso que esté siendo modelado.

1.6 Conclusiones del capítulo

Luego del estudio y análisis realizado del objeto de investigación del presente trabajo de diploma, apoyado en los métodos de la investigación científicos identificados, se concluye lo siguiente:

- Se construyó el marco conceptual que soporta la investigación y se adquirió el conocimiento necesario sobre los modelos de procesos de software, herramientas y lenguajes, seleccionándose los más adecuadas para el cumplimiento del objetivo general propuesto.
- Se identificó la metodología Extreme Programming (XP) como guía del proceso de desarrollo, pues provee un marco metodológico integrado de principios, prácticas y técnicas concretas que se ajustan al entorno y/o condiciones del proyecto y tienen el potencial de acelerar el tiempo de desarrollo.
- El perfil tecnológico propuesto de la propuesta está sujeto a (Apache como servidor web, MySQL como sistema gestor de bases de datos, lenguajes de programación PHP y el Framework Laravel).

Capítulo 2: Análisis y Diseño de la Solución:

En este capítulo se recogen las características de la propuesta de solución, se presentan los artefactos generados durante el proceso de desarrollo del *software*, como es el caso de las historias de usuario y tareas de ingeniería. Se muestra la arquitectura utilizada para el desarrollo, definiendo los patrones a utilizarse.

2.1 Fase: Exploración

En esta etapa el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

2.1.1 Propuesta de Solución al módulo Control de Calidad

Se implementará un módulo de la aplicación **Sis_Control** que permita la gestión de la información generada, durante el proceso de liberación del Plasma para el control de la Calidad.

2.1.2 Especificación de requisitos

Los requisitos son características requeridas por sistema que posibilitan cumplir con un objetivo o solucionar un problema determinado. Estos son necesarios para el cumplimiento de las especificaciones del cliente.

2.1.2.1 Requisitos Funcionales

Tabla 2: Requisitos Funcionales

RF 1	Autenticar usuarios.
RF 2	La página principal muestra un resumen de todo el proceso
RF 3	Visualización del inventario del plasma.
RF 4	Visualización de la realización del proceso.
RF 5	Visualizar plasma sin muestreo.
RF 6	Generar informe de banco de sangre sin muestreo. (en PDF)
RF 7	Visualizar plasma sin PCR con muestreo.
RF 8	Generar informe de plasma sin PCR con muestreo. (en PDF)

RF 9	Visualizar muestreo del plasma.
RF 10	Generar registro de muestreo. (en PDF)
RF 11	Adicionar muestreo de PCR
RF 12	Visualizar PCR del Plasma
RF 13	Generar registro de bolsa de plasma por PCR. (en PDF)
RF 14	Generar remisión de resultados reactivos. (en PDF)
RF 15	Adicionar registros de PCR
RF 16	Visualizar aprobaciones del plasma
RF 17	Generar registro de informe final de resultados de muestreo de plasma. (en PDF)
RF 18	Visualizar las bolsas positivas
RF 19	Visualizar los bancos con bolsas positivas en PCR
RF 20	Generar certificado de análisis

2.1.2.2 Requisitos no funcionales

Según *Somerville* los requisitos no funcionales o requerimientos no funcionales no son más que limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema. (32)

❖ **Requerimientos de confiabilidad:**

RNF 1: el servidor donde estará la aplicación de sistemas de control debe contar con un sistema de respaldo eléctrico.

❖ **Requerimientos de espacio:**

RNF 2: el servidor debe contar como mínimo de un espacio en disco de 1TB.

❖ **Requerimientos de rendimiento:**

RNF 3: el servidor debe contar como mínimo con un procesador *Intel Xeon E7330* y 16 gb de RAM.

RNF 4: el servidor debe constar con un servidor web con *Apache 2.4* y un servidor de base datos *MySql 5.0*

RNF 5: la aplicación debe ser escalable, permitiendo incorporarle nuevas funcionalidades sin afectar las existentes.

❖ **Requerimiento de Usabilidad:**

➤ **Protección contra errores de usuario:**

RNF6: Ante una excepción se notificará al usuario mediante un mensaje de error.

➤ **Estética de la interfaz de usuario:**

RNF7: El sitio web deberá tener una estructura clara, ordenando el contenido y las funciones de la aplicación en pestañas o menús que abarquen todas las funcionalidades disponibles, según el perfil de seguridad del usuario conectado.

RNF8: En los formularios de entrada, se valorará la inclusión de elementos de interacción asíncrona en la interfaz del cliente que mejoren la usabilidad de la aplicación. Por ejemplo, al rellenar los datos el usuario, debe facilitarse la selección de valores conocidos y filtrando automáticamente los valores aplicables conforme el usuario los teclea en el campo.

2.1.3 Validación de requisitos

La validez de los requisitos es muy importante antes de comenzar un desarrollo de *software* y tiene como objetivo comprobar que estos son correctos. Para ello debe de hacerse una comprobación de la correspondencia entre las descripciones iniciales y si el modelo es capaz de responder al planteamiento inicial. Para llevar a cabo esto, se suele realizar comprobando que el modelo obtenido responde de la misma forma deseada que la que el cliente pide o si otras respuestas del modelo convencen al cliente. En algunos casos será necesario construir prototipos con una funcionalidad similar muy reducida para que el cliente se haga una idea aproximada del resultado. Los parámetros a validar en los requisitos son:

- **Validez:** no basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos.
- **Consistencia:** no debe haber contradicciones entre unos requisitos y otros.
- **Compleitud:** deben estar todos los requisitos. Esto es imposible en un desarrollo iterativo, pero, al menos, deben estar disponibles todos los requisitos de la iteración en curso.
- **Realismo:** se pueden implementar con la tecnología actual.
- **Verificabilidad:** tiene que existir alguna forma de comprobar que cada requisito se cumple.

Se escogió para la validación de los requisitos de la herramienta la técnica prototipo de interfaz de usuario

2.2 Fase: Planificación.

En esta etapa se crean las historias de usuario que describen la salida necesaria de la aplicación, sus características y las funcionalidades del software que se van a elaborar. El cliente asigna un valor (es decir, una prioridad) a la historia con base en el valor general de la característica o función para el negocio. Después, los miembros del equipo XP evalúan cada historia y le asignan un costo, medido en semanas de desarrollo, las que no deben superar las tres semanas.

2.2.1 Historias de usuarios

Las historias de usuarios tienen una prioridad (Alta, Media, Baja) asociada, definida por el cliente y la importancia que tiene para el funcionamiento de la herramienta las funcionalidades antes mencionadas. El riesgo de desarrollo está dado por la ausencia del desarrollador por enfermedad o por pérdida de información imprescindible. El tiempo de cada funcionalidad será estimado por el desarrollador. La estimación del tiempo está dada de acuerdo al tiempo que toma la tarea en ser desarrollada. A continuación, se describen los ejemplos de historias de usuario (HU1, HU2, HU3) para el desarrollo del módulo de la aplicación. Las restantes historias se podrán visualizar en los anexos.

Tabla 3: Historia de Usuario: Autenticar usuario

Historia de Usuario	
Número: HU1	Usuario: Todos
Nombre de historia: Autenticar usuario	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 1	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: El usuario inserta su identificador (Ejemplo: ivan) y contraseña en el formulario de autenticación para acceder a las funcionalidades del sistema.	
Observaciones: Si el sistema no identifica al usuario le mostrará una notificación sobre este problema.	

Tabla 4: Historia de Usuario: Página principal muestra un resumen

Historia de Usuario	
Número: HU2	Usuario: Todos
Nombre de historia: La página principal muestra un resumen de todo el proceso	
Prioridad: Alta	Riesgo en desarrollo: Alta
Iteración: 1	Punto estimado: 0.5
Programador Responsable: Iván Lázaro Isa Regueiro	
Descripción: el usuario después de autenticarse, el sistema muestra una pantalla principal con tablas y gráficos de resumen de todo el proceso	
Observaciones:	

Tabla 5: Historia de Usuario: Visualizar inventario del plasma

Historia de Usuario	
Número: HU3	Usuario: Administrador
Nombre de historia: Visualizar inventario del plasma	
Prioridad: Alta	Riesgo en desarrollo: Alta
Iteración: 1	Punto estimado: 1
Programador Responsable: Iván Lázaro Isa Regueiro	
Descripción: El usuario debe dar clic en el menú desplegable del módulo <i>Control de Calidad</i> y luego en la opción <i>Inventario del Plasma</i> .	
Observaciones:	

2.2.2 Estimación del tiempo

En la siguiente tabla se muestra la estimación del tiempo de las historias de usuario, cuando se denota un punto de estimación con números enteros se refiere a semanas y cuando se realiza con números flotantes la estimación está dada en días o semanas y días.

Tabla 6: Estimación del tiempo de las Historias de Usuarios

Numero	Historia de Usuario	Estimación en Semanas
1	Autenticar usuarios.	0.5
2	La página principal muestra un resumen de todo el proceso	1
3	Visualización del inventario del plasma.	0.5
4	Visualización de la realización del proceso	0.5
5	Visualizar plasma sin muestreo.	0.5
6	Generar informe de banco de sangre sin muestreo	0.5
7	Visualizar plasma sin PCR con muestreo.	0.5
8	Generar informe de plasma sin PCR con muestreo.	1
9	Visualizar muestreo del plasma.	0.5
10	Generar registro de muestreo	0.5
11	Adicionar muestreo de PCR	0.5
12	Visualizar PCR del Plasma	0.5
13	Generar registro de bolsa de plasma por PCR	0.5
14	Generar remisión de resultados reactivos.	1
15	Adicionar registros de PCR	0.5
16	Visualizar aprobaciones del plasma	0.5
17	Generar registro de informe final de resultados de muestreo de plasma	1
18	Visualizar las bolsas positivas	0.5
19	Visualizar los bancos con bolsas positivas en PCR	0.5
20	Generar certificado de análisis	0.5

2.2.3 Plan de iteraciones

En la metodología XP, la duración ideal de una iteración está entre una y tres semanas. Para estas se establecen un conjunto de HU definidas que serán implementadas en cada iteración, superando las pruebas de aceptación las cuales son realizadas al final de cada ciclo. (33)

Tabla 7: Plan de Iteraciones

Iteraciones	Historia de Usuario	Duración en Semana	
Iteración 1	Autenticar usuarios.	0.5	3
	La página principal muestra un resumen de todo el proceso	1	
	Visualización del inventario del plasma.	0.5	
	Visualización de la realización del proceso	0.5	
	Visualizar plasma sin muestreo.	0.5	
Iteración 2	Generar informe de banco de sangre sin muestreo	0.5	3
	Visualizar plasma sin PCR con muestreo.	0.5	
	Generar informe de plasma sin PCR con muestreo.	1	
	Visualizar muestreo del plasma.	0.5	
	Generar registro de muestreo	0.5	
Iteración 3	Adicionar muestreo de PCR	0.5	3
	Visualizar PCR del Plasma	0.5	
	Generar registro de bolsa de plasma por PCR	0.5	
	Generar remisión de resultados reactivos.	1	
	Adicionar registros de PCR	0.5	
Iteración 4	Visualizar aprobaciones del plasma	0.5	3
	Generar registro de informe final de resultados de muestreo de plasma	1	
	Visualizar las bolsas positivas	0.5	
	Visualizar los bancos con bolsas positivas en PCR	0.5	
	Generar certificado de análisis	0.5	

2.2.5 Plan de entrega

El cronograma de entregas establece las HU que serán agrupadas para conformar una entrega y el orden de las mismas. El cliente es quien las agrupa según su prioridad. El cronograma de entregas se analiza sobre la base de las estimaciones de tiempos de desarrollo (33)

La siguiente tabla muestra la cantidad de historias de usuarios, así como las fechas de entrega.

Tabla 8: Plan de Entrega

Historias de Usuario por Iteración	Iteración 1	Iteración 2	Iteración 3	Iteración 4	Fecha de Entrega
5	Ver.1.0	Finalizado	Finalizado	Finalizado	24/07/2023
5	-	Ver.1.0	Finalizado	Finalizado	14/08/2023
5	-	-	Ver.1.0	Finalizado	04/09/2023
5	-	-	-	Ver.1.0	25/04/2023

2.3 Fase 3: Diseño

Arquitectura

En los últimos años la arquitectura de *software* ha tenido una gran atención en el campo de la ingeniería de *software*. Con el desarrollo de las tecnologías y el aumento en el campo económico, ha quedado en evidencia que un diseño previo y cuidadoso de la arquitectura de un producto puede reducir enormemente el número de fallas del mismo. Indicadores de calidad como la eficiencia, usabilidad, confiabilidad y seguridad pueden ser verificados y estimados con respecto al diseño de la arquitectura escogida, antes de que cualquier código sea escrito.

Las personas dedicadas al desarrollo y el mantenimiento de productos de software, constantemente, buscan optimizar las metodologías para obtener buenos resultados. Una de las soluciones para mejorar el proceso de desarrollo ha sido la arquitectura basada en “capas” que separan el código de un programa en función de sus responsabilidades. Esta separación también incluye a los responsables del desarrollo del software, tal es el caso de los diseñadores, quienes podrían no estar familiarizados con determinadas partes de un código que les competen a los programadores, por lo que necesitan separarse de ciertos módulos de programación.

Estos ejemplos, demuestran la necesidad de aplicar una arquitectura útil como es el Modelo Vista-Controlador (MVC), que requiere de la separación del código de un programa en capas atendiendo a sus responsabilidades.

Arquitectura cliente-servidor

Esta arquitectura de diseño significa que las "apps" en internet se implementan instalando en nuestro ordenador la aplicación "cliente"⁹ que conecta con el ordenador "servidor"¹⁰

Todo ordenador de internet puede ser "cliente" o "servidor" de algún servicio. Lo normal es habilitar servidores en las organizaciones y que el resto de ordenadores sean clientes que conectan con ellos. (34)

Algunos ejemplos:

- Outlook es un cliente de correo que sirve para leer los correos del servicio hotmail
- WhatsApp o Telegram son clientes de servicios de chat, vídeo, audio
- Firefox es un cliente web, Chrome también ("navegadores")
- Internet Information Server y Apache Web Server Apache Web Server, también conocido como Apache HTTP Server, es un servidor web de código abierto ampliamente utilizado en la entrega de contenido web. Fue desarrollado y es mantenido por la Apache Software Foundation. Apache es compatible con varios sistemas operativos, incluyendo Windows, Linux, macOS y Unix, y es conocido por su estabilidad, seguridad y flexibilidad. son servidores web

⁹ Ordenador o programa informático que solicita información.

¹⁰ Ordenador o programa informático que dispone de información para servir las peticiones de los clientes.

2.3.1 Patrón MVC

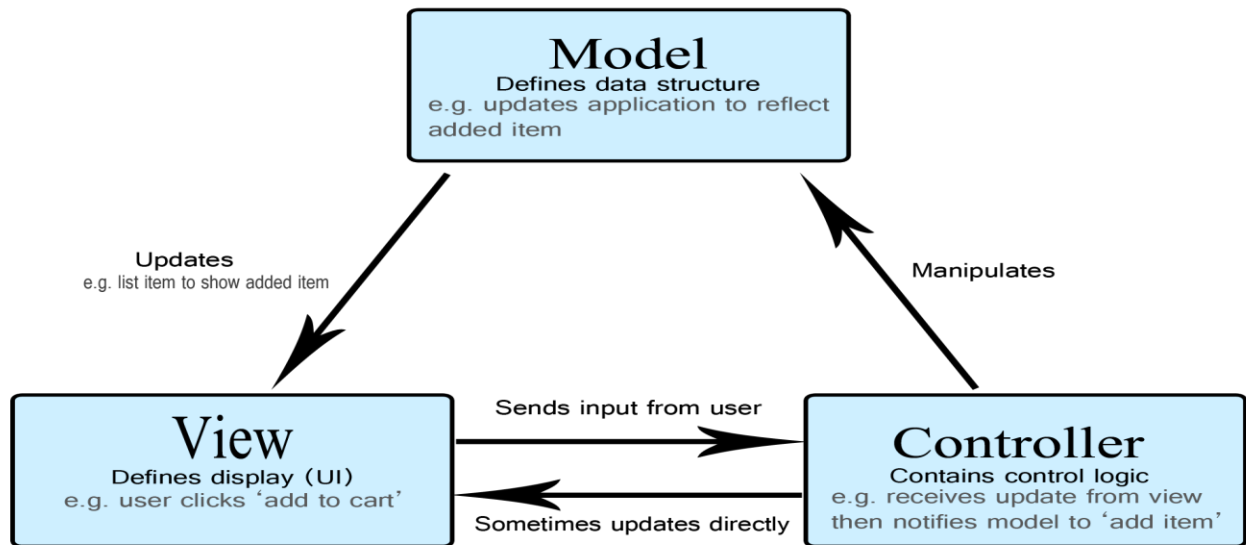


Ilustración 2: Patrón de Diseño Modelo-Vista-Controlador.

"El propósito de este patrón es simplificar la implementación de aplicaciones de acuerdo a las peticiones de los usuarios y los datos a desplegar".

El Model-View-Controller o Modelo-Vista-Controlador (MVC) es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista, o a cualquier parte del sistema puedan ser hechas con un mínimo impacto en el componente del modelo de datos o en los otros componentes del sistema. Este patrón cumple perfectamente el cometido de modularizar un sistema.

Algunos aspectos de la evolución del patrón MVC:

- HMVC (MVC Jerárquico)
- MVA (Modelo-Vista-Adaptador)
- MVP (Modelo-Vista-Presentador)
- MVVM (Modelo-Vista Vista-Modelo)
- ... y otros que han adaptado MVC a diferentes contextos.

Los tres principales componentes del patrón MVC son:

- **Modelo:** Representa los datos que el usuario está esperando ver, en algunos casos el Modelo consiste de Java Beans.
- **Vista:** Se encarga de transformar el modelo para que sea visualizada por el usuario, ya sea un archivo de texto normal o en una página Web (HTML o JSP) que el navegador pueda desplegar. El propósito de la Vista es convertir los datos para que el usuario le sea significativos y los pueda interpretar fácilmente; la Vista no debe trabajar directamente con los parámetros de request, debe delegar esta responsabilidad al controlador.
- **Controlador:** Es la parte lógica que es responsable de procesamiento y comportamiento de acuerdo a las peticiones (request) del usuario, construyendo un modelo apropiado, y pasándolo a la vista para su correcta visualización. En el caso de una aplicación Web Java en la mayoría de los casos el Controlador es implementado por un servlet.

Uso en aplicaciones Web

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón; estos se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero en el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier hiperenlace o formulario al controlador y después recibe de la vista una página completa y actualizada (u otro documento); tanto el modelo como el controlador (y buena parte de la vista) están completamente alojados en el servidor.

- Vista: la página HTML.
- Controlador: código que obtiene los datos dinámicamente y genera el contenido HTML.
- Modelo: la información almacenada en base de datos o en XML.

2.3.2 Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (35). Para la propuesta de solución se hizo uso de varios Patrones Generales de Software para Asignar Responsabilidad o por sus siglas en inglés (**GRASP**).

Experto: es el patrón más utilizado a la hora de asignar responsabilidades, principio básico del diseño orientado a objetos. Se fundamenta en que los objetos pueden realizar funciones dependiendo de la información que poseen (36). Las clases generadas poseen un grupo de funcionalidades que facilitan el acceso y la manipulación de los datos de las entidades persistentes en la base de datos.

Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. (36)

Bajo acoplamiento: soporta el diseño de clases más independientes, que reducen el impacto de los cambios, y también más reutilizables, que acrecientan la oportunidad de una mayor productividad. No puede considerarse en forma independiente de otros patrones como Experto o Alta Cohesión, sino que más bien ha de incluirse como uno de los principios del diseño que influyen en la decisión de asignar responsabilidades (36). El patrón se encarga de asignar una responsabilidad de modo que su colocación no aumente el acoplamiento produciendo resultados negativos del alto acoplamiento. Su aplicación se puede apreciar al no asociar las clases del modelo con las de la vista o el controlador, por lo que la dependencia entre las clases, se mantiene baja.

Alta cohesión: la cohesión es la medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una clase tiene responsabilidades moderadas en un área funcional y colabora con las otras para llevar a cabo las tareas. (36)

Controlador: en todos los casos, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada. Este patrón ofrece una guía para tomar decisiones apropiadas que generalmente se aceptan. (36)

DAO: Este patrón implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. El componente de negocio utiliza la interfaz más simple expuesta por el DAO para

sus clientes. Este patrón oculta completamente los detalles de implementación del origen de datos de sus clientes. Este modelo permite adaptarse a los diferentes esquemas de almacenamiento sin afectar a sus clientes o componentes de negocio. Esencialmente actúa como un adaptador entre el componente y la fuente de datos. Este patrón se evidencia en las clases *AbstractDAO* y similares a esta. (37)

2.3.3 Prototipo interfaz de usuario

El prototipo de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario de un sistema *software* que permite a clientes y usuarios entender más fácilmente la propuesta de los ingenieros de requisitos para resolver sus problemas de negocio. Los dos tipos principales de prototipos de interfaz de usuario son:

- **Desechables:** se utilizan sólo para la validación de los requisitos y posteriormente se desechan. Pueden ser prototipos en papel o en *software*.
- **Evolutivos:** una vez utilizados para la validación de los requisitos, se mejora su calidad y se convierten progresivamente en el producto final.

A continuación, se muestra las Figura 3, 4, 5 con el prototipo de interfaz de usuario del tipo Evolutivo:

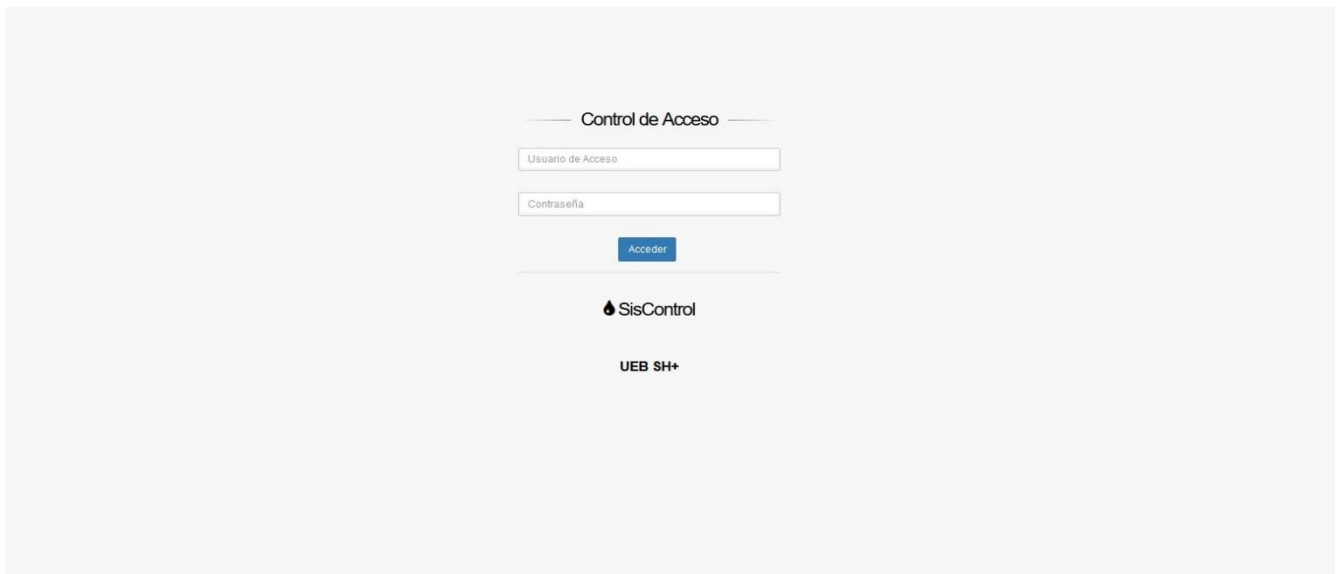


Ilustración 3: Prototipo de interfaz de usuario del tipo Evolutivo: Autenticar Usuario

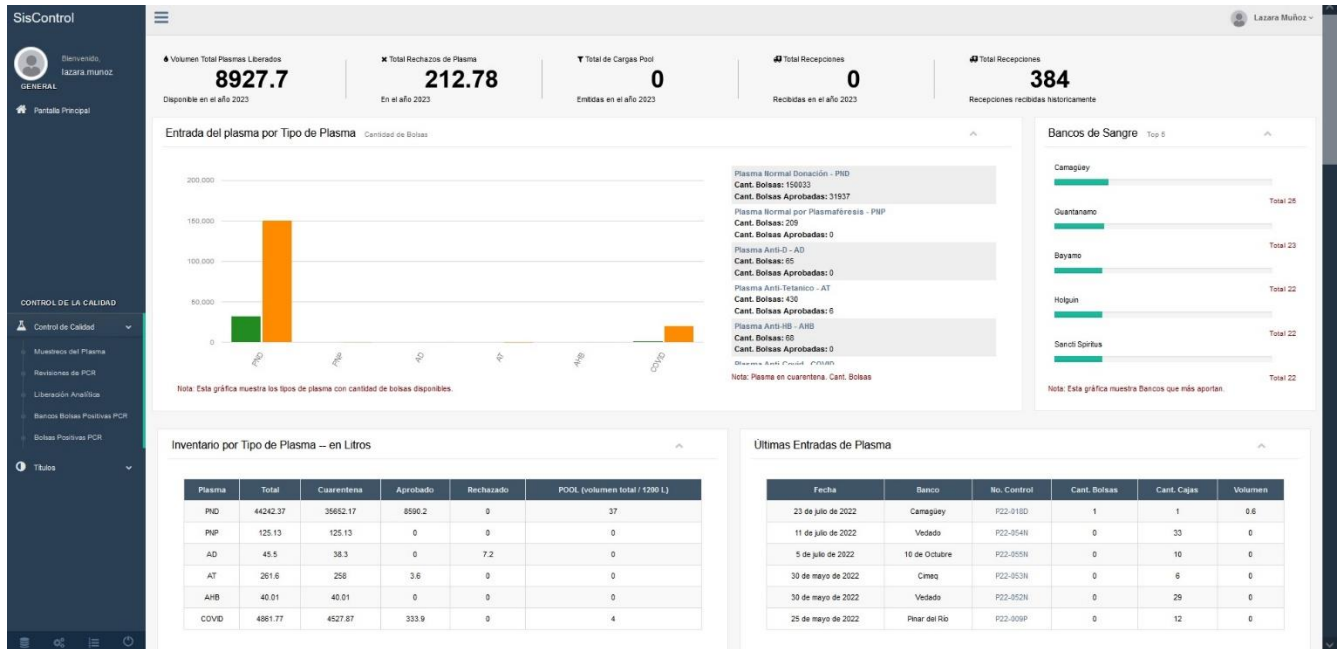


Ilustración 4: Prototipo de interfaz de usuario del tipo Evolutivo: Pantalla Principal

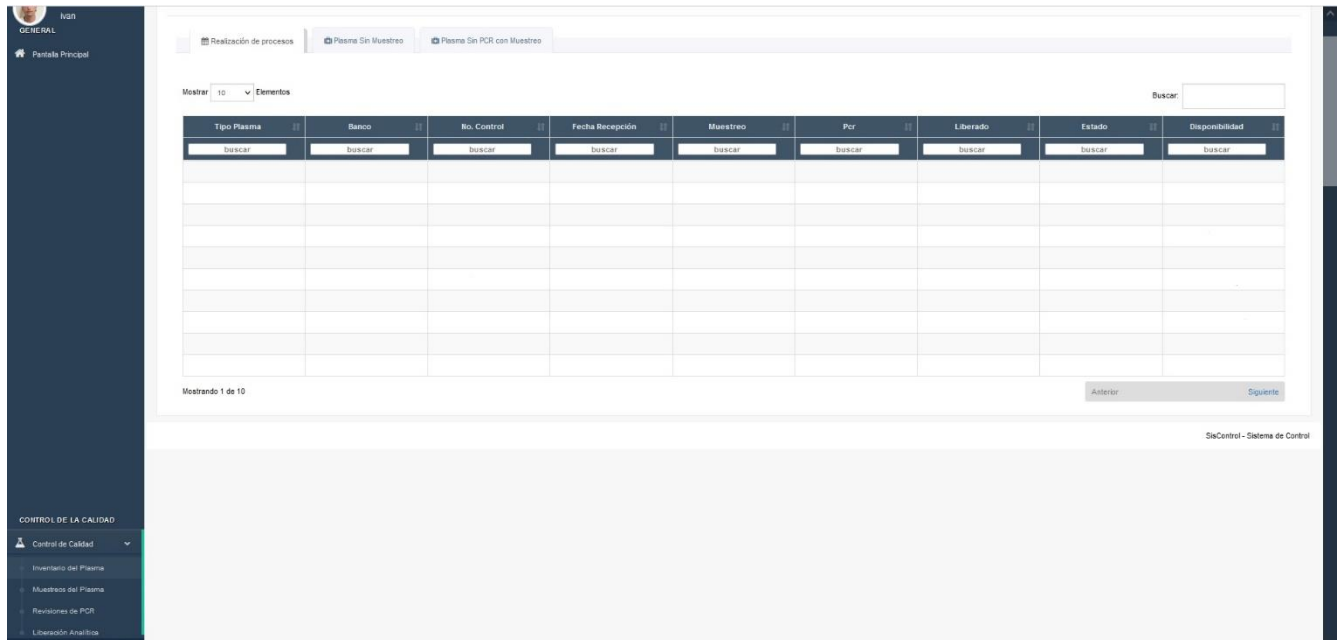


Ilustración 5: Prototipo de interfaz de usuario del tipo Evolutivo: Inventario del Plasma

2.3.4 Tarjetas CRC

Se utilizan para resumir el significado de una clase ya que mediante su uso se pueden representar las responsabilidades de las clases y las interacciones entre ellas. También sirven para simular escenarios que describen los requerimientos del sistema. Estos escenarios modelan el comportamiento del sistema que se está estudiando. (38).

Tabla 9: Tarjeta CRC (MINSAP)

Tarjeta CRC	
Reactivas Minsap	
Responsabilidad	Colaboración
+ Lista_información_análisis_minsap () + Adicionar_datos_minsap () + Actualizar_información_análisis_minsap ()	Aprobaciones_Banco Reactivas_Rechazadas

Tabla 10: Tarjeta CRC (PCR General)

Tarjeta CRC	
PCR General	
Responsabilidad	Colaboración
+ Listar_análisisPCR + Adicionar_datos_análisisPCR () + Actualizar_información_análisisPCR () + Exportar_PCR	Aprobaciones_Banco

Tabla 11: Tarjeta CRC (Aprobaciones Banco)

Tarjeta CRC	
Aprobaciones Banco	
Responsabilidad	Colaboración
+ Obtener_Análisis_PCR (id_PCR) + Obtener_Análisis_minsap (id_minsap) + Modificar_Análisis () + Exportar_informe ()	Reactivas_Minsap PCR_General Tipo_Plasma Banco_Sangre

2.3.5 Modelo de Datos

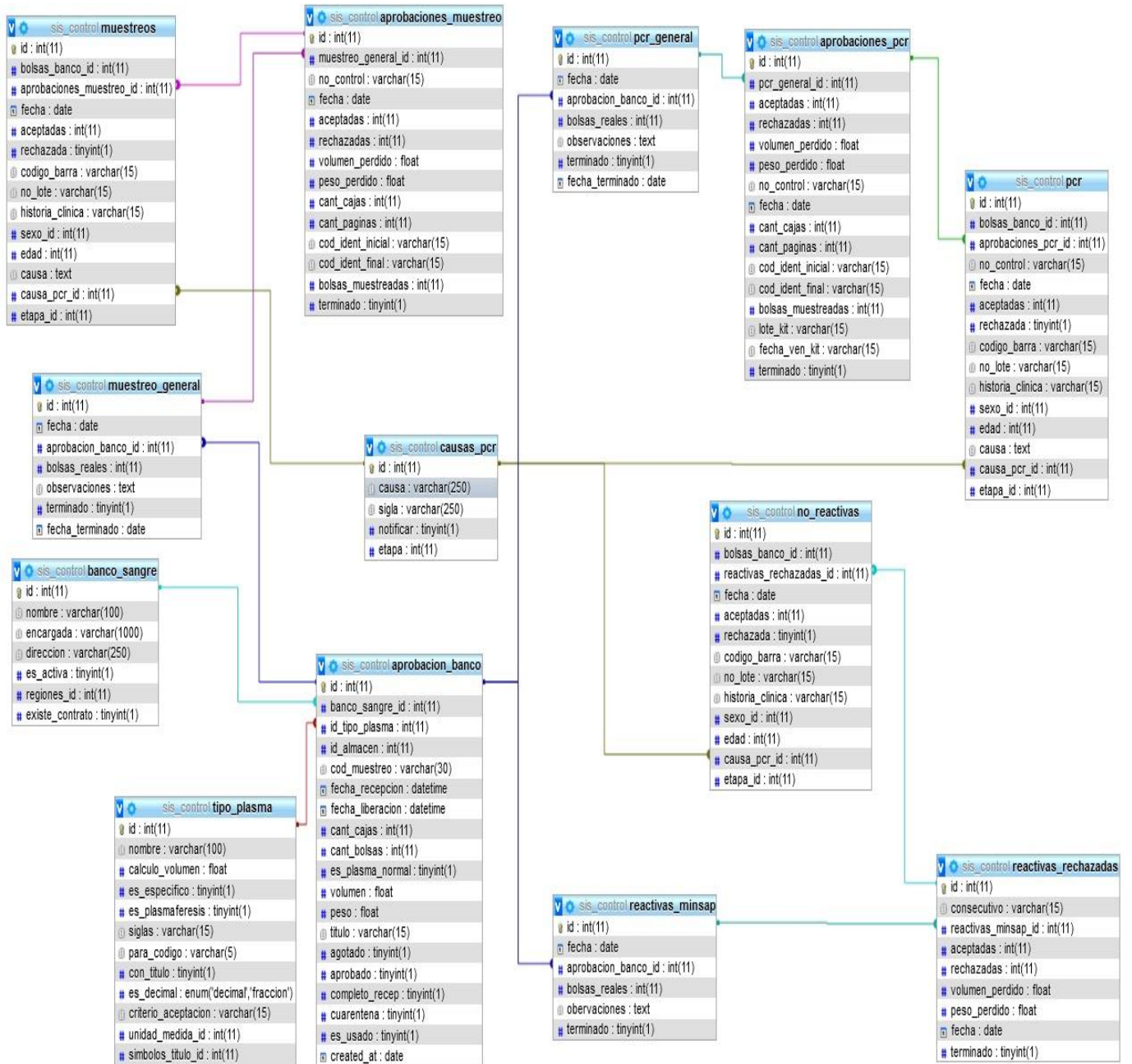


Ilustración 6: Diagrama de Relación de Datos (Diseñador de phpMyAdmin)

2.4 Conclusiones del capítulo

- Se definieron las HU que describen los aspectos principales a tener en cuenta para el desarrollo de la solución, contemplando una visión futura de las funcionalidades que debe cumplir el sistema.
- Asociado a estas HU se construyó el plan de entregas que enmarcó el tiempo de desarrollo en 12 semanas, determinando un cronograma que especifica las entregas que deben hacerse y conjuntamente se elaboró el prototipo de interfaz funcional de la solución y una tarjeta CRC que representa las funcionalidades a implementar definiendo las responsabilidades y los colaboradores.
- Los principios, prácticas y técnicas que propone la metodología XP aportó los artefactos necesarios que darán paso a la implementación de la solución.

Capítulo 3: Implementación y Pruebas de la Solución

Con el objetivo de materializar la solución propuesta de la presente investigación y cumplir con los requisitos obtenidos en el capítulo anterior, se lleva a cabo la fase de implementación y pruebas. Se continúa la fase de **ejecución** y se realiza la fase de **cierre** de la metodología escogida anteriormente, se recogen los estándares de codificación que se emplearán en el desarrollo de la herramienta, así como el diagrama de despliegue y finalmente se muestran las pruebas realizadas para la validación del correcto funcionamiento de la propuesta de solución.

3.1 Estándares de codificación

Entre las buenas prácticas a aplicar durante el proceso de desarrollo de un *software* se encuentran la refactorización del código y la propiedad compartida del mismo, de forma que todo el personal pueda corregir y entender cualquier parte del producto. Un estándar de codificación es necesario pues debe existir un estilo consistente a lo largo de todo el desarrollo de la herramienta y no el preferido por cada programador involucrado en este proceso. Esto posibilita que se pueden añadir nuevas funcionalidades, modificar ya existentes o depurar errores con gran facilidad y obtener un código con mayor legibilidad. A continuación, se define el estándar a utilizar en la implementación de la herramienta a desarrollar:

- Se empleará el estilo **lowerCamelCase25** para nombre de identificadores, donde todos deberán comenzar con una letra.
- Se hará uso de comentarios en aquellas funcionalidades de mayor complejidad.

```
* @property \App\Models\Acceso acceso
* @property \App\Models\Personas persona
* @property integer acceso_id
* @property integer persona_id
*/
class AccesosPersona extends Model
{
    public $table = 'accesos_personas';

    public $timestamps = false;

    public $fillable = [
        'acceso_id',
        'persona_id'
    ];

    /**
     * The attributes that should be casted to native types.
     *
     * @var array
     */
    protected $casts = [
        'id' => 'integer',
        'acceso_id' => 'integer',
        'persona_id' => 'integer'
    ];

    /**
     * Validation rules
     *
     * @var array
     */
    public static $rules = [
```

Ilustración 7: Ejemplo de estilo lowerCamelCase25

3.2 Diagrama de Despliegue

Es un tipo de diagrama de UML que muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red, conexiones TCP/IP y USB.

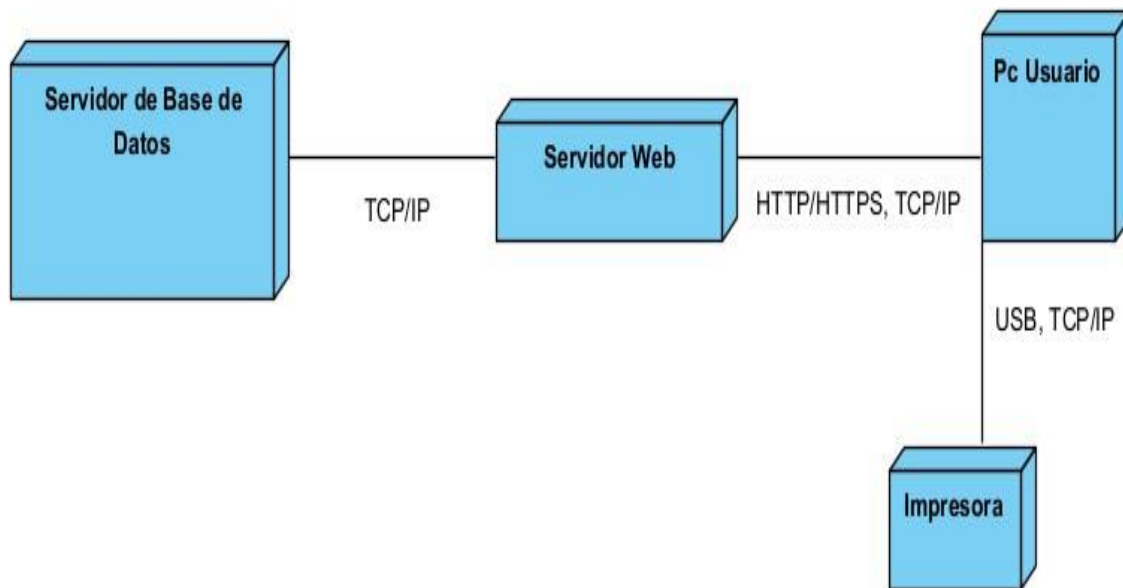


Ilustración 8: Diagrama de Despliegue (Elaboración Propia)

La comunicación entre la **PC usuario** y el **Servidor Web**, se realiza utilizando *HTTP* como protocolo para la transferencia de información y desde el **Servidor Web** hacia la **Base de Datos usuarios** la comunicación es por parte del protocolo *TCP/IP*, además la seguridad del sistema se incrementa al ser posible acceder a la **Base de Datos de los usuarios** solamente desde el **Servidor Web**

PC usuario: Estación de trabajo que necesita un navegador web para conectarse al **Servidor Web** utilizando el protocolo de comunicación *HTTP*.

Servidor Web: Estación de trabajo que hospeda el código fuente de la herramienta desarrollada.

Base de Datos de usuarios: Base de Datos responsable de almacenar los usuarios del sistema.

3.3 Pruebas de una Webapp.

Las pruebas de una aplicación web o como se conoce en inglés una *Webapp*, es una colección de actividades relacionadas con una sola meta: descubrir errores en el contenido, función, utilidad, navegabilidad, rendimiento, capacidad y seguridad de esa aplicación. Para lograr esto, se aplica una estrategia de pruebas que abarca tanto revisiones como pruebas ejecutables. (39)

Estrategia de pruebas propuestas

Una estrategia de prueba del *software* integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del *software*. Y lo que es más importante, una estrategia de prueba del *software* proporciona un mapa a seguir para el responsable del desarrollo del *software*, a la organización de control de calidad y al cliente: un mapa que describe los pasos que hay que llevar a cabo como parte de la prueba, cuándo se deben planificar y realizar esos pasos y cuánto esfuerzo, tiempo y recursos se van a requerir. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de las pruebas y la agrupación y evaluación de los datos resultantes. (39)

Estrategia de pruebas de software para Webapps.

La estrategia para probar *Webapps* adopta los principios básicos para todas las pruebas de *software* y aplica una estrategia y tácticas que se usan para sistemas orientados a objetos. Los siguientes pasos resumen el enfoque: (39)

1. El modelo de contenido para la *Webapp* se revisa para descubrir errores.
2. El modelo de interfaz se revisa para garantizar que todas las historias de usuarios puedan adecuarse a las necesidades del cliente.
3. El modelo de diseño para la *Webapp* se revisa para descubrir errores de navegación.
4. La interfaz de usuario se prueba para descubrir errores en los mecanismos de presentación y/o navegación.
5. A cada componente funcional se le aplica una prueba de unidad.
6. Se prueba la navegación a lo largo de toda la arquitectura.
7. La *Webapp* se implementa en varias configuraciones ambientales diferentes y se prueba en su compatibilidad con cada configuración.
8. Las pruebas de seguridad se realizan con la intención de explotar vulnerabilidades en la *Webapp* o dentro de su ambiente.

9. Se realizan pruebas de rendimiento.

10. La *Webapp* se prueba mediante una población de usuarios finales controlada y monitorizada. Los resultados de su interacción con el sistema se evalúan por errores de contenido y navegación, preocupaciones de facilidad de uso, preocupaciones de compatibilidad, así como confiabilidad y rendimiento de la *Webapp*.

3.4. Pruebas unitarias.

Las pruebas unitarias enfocan los esfuerzos de verificación en la unidad más pequeña del diseño de *software*: el componente o módulo de *software*. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Para garantizar la calidad del *software* en cuestión, el desarrollo de la herramienta se llevó a cabo utilizando el método de prueba de caja blanca y una práctica llamada: Desarrollo guiado por pruebas o TDD (del inglés *Test-Driven Development*), que presenta el siguiente ciclo de vida:

1. Confeccionar una prueba para que el código falle y así detectar vulnerabilidades.
2. Escribir el código fuente para que pase dicha prueba.
3. Llevar a cabo la optimización y refactorización del mismo, mientras que la prueba no falle.
4. Repetir este proceso hasta que el proyecto esté completado.

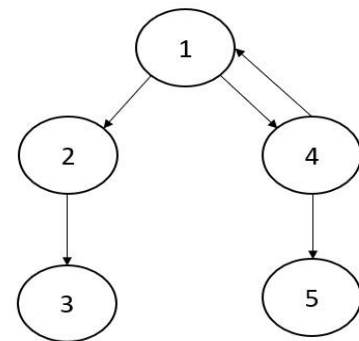
Este flujo de trabajo permite un desarrollo incremental evitando tener que emplear tiempo en encontrar un error en particular, pues el desarrollador es capaz de ir directamente y cambiar la parte de la aplicación que esté presentando problemas. De esta forma es posible saber si la implementación de nuevas funcionalidades o cambios en las ya implementadas están afectando el comportamiento del sistema, solamente con ejecutar dichas pruebas (27).

A continuación, se realiza la prueba unitaria con la técnica de camino básico a las funcionalidades **testDeleteTipoPlasma**

3.4.1. Resultados de las pruebas unitarias.

Luego de numerar las líneas de código, se diseña la gráfica del programa que describe el flujo de control lógico empleando nodos y aristas

- 1.tipoPlasma = \$this->makeTipoPlasma ()
- 2.this->json('DELETE', '/api/v1/tipoPlasmas/.\$tipoPlasma->id)
- 3.this->assertApiSuccess ()
- 4.this->json('GET', '/api/v1/tipoPlasmas/.\$tipoPlasma->id)
- 5.this->assertResponseStatus(404)



Luego de obtener el grafo de la funcionalidad testDeleteTipoPlasma se calcula la complejidad ciclomática $V(G)$, la cual constituye una métrica de *software* que proporciona una medida cuantitativa de la complejidad lógica del programa (28).

$$V(G) = (\text{cantidad_aristas} - \text{cantidad_nodos}) + 2$$

$$V(G) = (5 - 5) + 2 = 2$$

Una ruta independiente es cualquier ruta del programa que ingrese al menos un nuevo conjunto de instrucciones de procesamiento o una nueva condición. La cantidad de rutas independientes establecidas por la complejidad ciclomática antes calculada es de 2, a continuación, se presenta una tabla con las rutas y los caminos de ellas.

Tabla 12: Lista de Rutas de la Funcionabilidad testDeleteTipoPlasma

No. Rutas	Caminos
Ruta 1	1-2-3
Ruta 2	1-4-5

El valor de V(G) ofrece además un límite superior del número de pruebas que debe diseñarse y ejecutarse para garantizar la cobertura de todas las instrucciones. Por este motivo, se diseñan casos de pruebas para ser aplicados a cada ruta independiente de la funcionalidad antes mencionada.

Tabla 14: Caso de prueba unitaria de la primera ruta.

Caso de prueba unitaria	
No. Ruta: Ruta 1	Ruta: 1-2-3
Responsable de prueba: Iván Lázaro Isa Regueiro	
Descripción de la prueba: Finalizar la secuencia en el código que ejecuta el usuario.	
Entrada: Elimina el tipo de plasma por el usuario dado un id, y se finaliza la secuencia del código.	
Resultado esperado: La modulo muestra la interfaz tipo de plasma y una notificación "Se elimino <i>#nombre plasma</i> "	
Evaluación de la prueba: Satisfactoria.	

Tabla 15: Caso de prueba unitaria de la segunda ruta

Caso de prueba unitaria	
No. Ruta: Ruta 2	Ruta: 1-4-5
Responsable de prueba: Iván Lázaro Isa Regueiro	
Descripción de la prueba: Finalizar la secuencia en el código que ejecuta el usuario.	
Entrada: Obtener tipo de plasma dado un id entrado por el usuario, y se finaliza la secuencia del código.	
Resultado esperado: El módulo muestra en su interfaz tipo de plasma, el nombre del plasma seleccionado	
Evaluación de la prueba: Satisfactoria.	

Se realizaron cuatro iteraciones de pruebas unitarias. En la primera iteración se detectaron 7 no conformidades, de las cuales se corrigieron las 7; en la segunda, 3 no conformidades, 2 de ellas de la primera iteración que no cumplieron con lo esperado y una nueva no conformidad de las cuales se corrigieron las tres, en la tercera iteración se detectó una no conformidad de la segunda iteración que no cumplieron con lo esperado y se realizó una cuarta iteración la

cual no arrojo ninguna no conformidad. Las no conformidades detectadas estaban asociadas a errores de validación y errores de redacción dentro del código. En la gráfica siguiente se visualiza el resultado de dicha prueba.

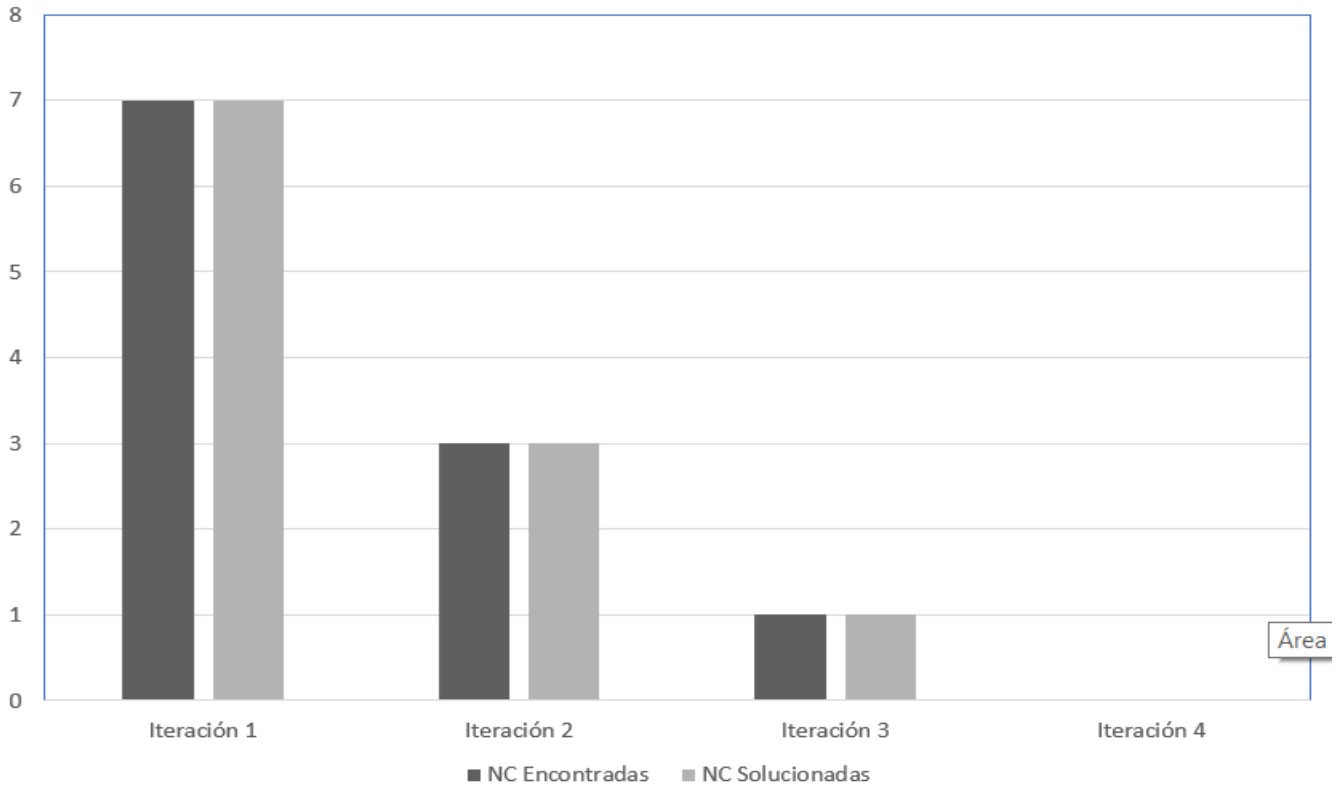


Ilustración 9: Gráfica de los resultados de las pruebas unitarias.

3.5. Pruebas de interfaz de usuario.

La verificación y validación de una interfaz de usuario de *Webapp* ocurre en tres puntos distintos. Durante el análisis de requerimientos, el modelo de interfaz se revisa para garantizar que se da conformidad a los requerimientos de los participantes y a otros elementos del modelo de requerimientos. Durante el diseño, se revisa el modelo de diseño de interfaz para garantizar que se logran los criterios de calidad genéricos establecidos para todas las interfaces de usuario y que los temas de diseño de interfaz específicos de la aplicación se abordaron de manera adecuada. Durante la prueba, la atención se centra en la ejecución de aspectos específicos de la aplicación de la interacción con el usuario, conforme se manifiesten por la sintaxis y la semántica de la interfaz. Además, la prueba proporciona una valoración final de la

usabilidad y se realizan mediante casos de prueba cuyo fin es validar que el *software* cumple con el nivel de calidad requerido.

En la herramienta se hace uso del método de prueba de caja negra a la interfaz de dicha herramienta para comprobar y corregir errores que se puedan presentar.

Método de prueba de caja negra.

Las pruebas de caja negra se centran en los requisitos funcionales del *software*, es decir, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra intenta encontrar errores de las siguientes categorías (40):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

Técnica de partición de equivalencia.

La partición de equivalencia es una técnica del método de caja negra que divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Un caso de prueba ideal descubre de primera mano una clase de errores (por ejemplo, procesamiento incorrecto de todos los datos carácter) que de otro modo podrían requerir la ejecución de muchos casos de prueba antes de observar el error general (39).

El diseño de casos de prueba para la partición de equivalencia se basa en una evaluación de las clases de equivalencia para una condición de entrada. Con los conceptos introducidos en la sección precedente, si un conjunto de objetos puede vincularse mediante relaciones que son simétricas, transitivas y reflexivas, se presenta una clase de equivalencia. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada (39).

Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana. Las clases de equivalencia pueden definirse de acuerdo con los siguientes lineamientos (39):

1. Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos inválidas.

2. Si una condición de entrada requiere un valor específico, se define una clase de equivalencia válida y dos inválidas.
3. Si una condición de entrada especifica un miembro de un conjunto, se define una clase de equivalencia válida y una inválida.
4. Si una condición de entrada es booleana, se define una clase válida y una inválida.

A diferencia de la prueba de caja blanca, que se lleva a cabo previamente en el proceso de prueba, las pruebas de caja negra tienden a aplicarse durante fases posteriores del proceso de pruebas. A continuación, se muestra el diseño de caso de prueba **Autenticar usuarios** por historias de usuarios de la herramienta.

Tabla 15: Caso de prueba de interfaz de usuario: Autenticar usuarios.

Caso de prueba de interfaz de usuario	
Número: CDPHU1.	Nombre: Autenticar usuarios.
Probador: Iván Lázaro Isa Regueiro	
Descripción: Prueba a la funcionalidad Autenticar usuarios.	
Condición de ejecución: El usuario debe estar autenticado.	
Entrada/Pasos de ejecución:	
1 el usuario accede a la interfaz de entrada de la herramienta.	
2 el usuario llena los campos usuario y contraseña del formulario Acceder .	
3 el usuario presiona el botón Acceder de la interfaz.	
4 la herramienta muestra la interfaz principal con el nombre Pantalla Principal	

Tabla 16: Resultados del Caso de prueba de interfaz de usuario: Autenticar usuarios.

Escenarios	Resultados esperados	Evaluación
EC1. El usuario introduce correctamente los datos.	La herramienta autentica al usuario y entra en la interfaz principal mostrando el nombre del usuario.	Satisfactoria.
EC2. El usuario introduce datos incorrectos.	La herramienta no autentica al usuario y muestra el siguiente mensaje: "Estas Credenciales son incorrectas o el usuario esta desactivado".	Satisfactoria.

EC3. El usuario deja campos vacíos.	La herramienta no autentica al usuario y muestra el siguiente mensaje: “Estas Credenciales son incorrectas o el usuario esta desactivado”.	Satisfactoria.
-------------------------------------	--	----------------

3.5.1. Resultados de las pruebas de interfaz de usuarios.

Para la validación de los requisitos funcionales por medio de las interfaces de los usuarios del módulo control de la calidad, se realizaron 3 iteraciones de pruebas de interfaz de usuarios a la misma, en las que se detectaron un total de 25 no conformidades, 20 en la primera iteración de las cuales fueron resueltas 20, en la segunda iteración se encontraron 8 no conformidades 3 de la primera iteración que no cumplieron como se esperaban y 5 nuevas las cuales fueron todas resueltas y se realizó una tercera iteración donde no se mostró ninguna no conformidad, lo cual arrojo como resultado final que la herramienta funciona correctamente. A continuación, se muestran gráficamente estos resultados.

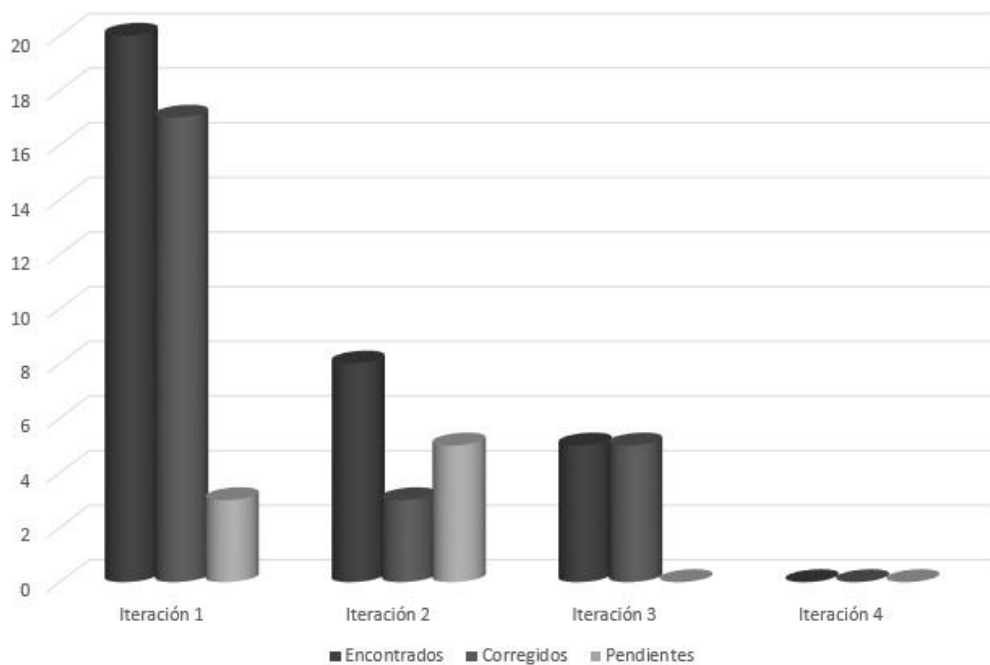


Ilustración 10: Gráfica de los resultados de las pruebas de interfaz de usuario

3.6. Pruebas de rendimiento.

Las pruebas de rendimiento son un conjunto de pruebas no funcionales que se realizan, para determinar la velocidad de ejecución de una tarea concreta en un sistema bajo condiciones particulares de trabajo.

Los objetivos de estas pruebas son:

- Validar y verificar atributos de la calidad del sistema: uso de los recursos, escalabilidad y fiabilidad.
- Comparación de sistemas para encontrar cuál de ellos funciona mejor.
- Determinar qué componentes del sistema provocan que el conjunto presente rendimientos bajos.

Tipos de Pruebas de Rendimiento

Prueba de Carga: prueba de rendimiento que se realiza para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada.

Objetivos:

- Mostrar los tiempos de respuesta de todas las transacciones importantes.
- Localizar los ‘cuellos de botella’ de una aplicación.

Pruebas de Estrés: Prueba de rendimiento que se realiza para observar el comportamiento de una aplicación bajo una cantidad de peticiones extrema.

Objetivos:

- “Romper” la aplicación.
- Determinar cómo rendirá la aplicación si la carga real supera a la carga esperada.

3.6.1. Resultados de las pruebas de rendimiento.

El módulo desarrollado fue sometido a varias pruebas de rendimiento para medir su funcionamiento. Estas pruebas estuvieron enfocadas en el análisis del comportamiento de los tiempos de respuesta de la funcionalidad: generar informes de análisis. Para ello se emplearon dos técnicas diferentes de generar informes de los análisis, para comprobar su impacto en dichos tiempos de respuesta:

Generar informes de análisis desde la aplicación: se genera desde la aplicación los informes seleccionados para ser generados.

Generar informes de análisis desde un pc con el Office: se genera desde un pc introduciendo los datos en el Office.

Las pruebas se llevaron a cabo en un ordenador con un microprocesador *Intel Core i3-2120* a 2.4GHz y 4 gb de RAM. Para probar el tiempo que la herramienta empleaba en generar 50 informe de análisis, hasta que se pudieran visualizar los informes seleccionados.

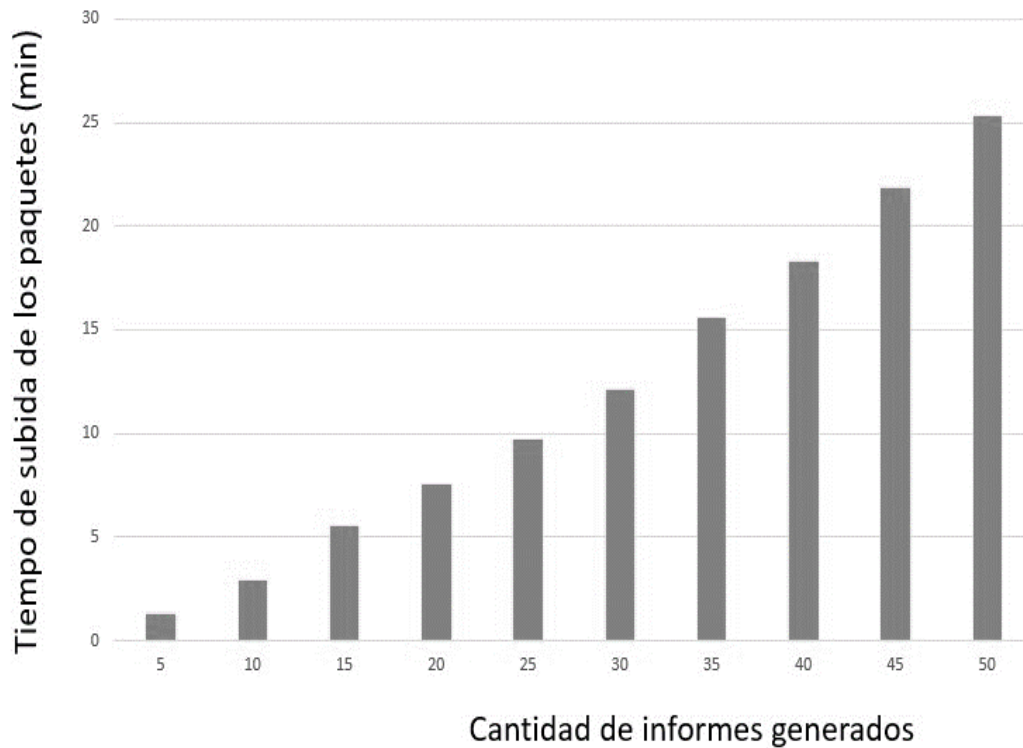


Ilustración 11: Gráfica del resultado del tiempo de general informes desde el modulo

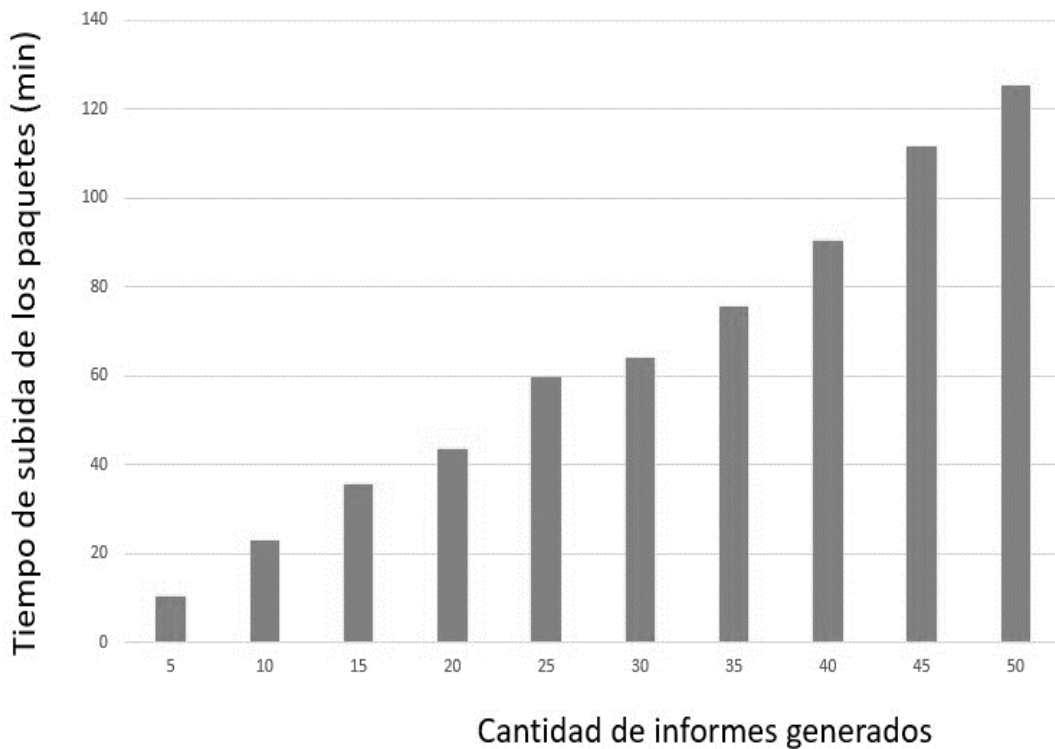


Ilustración 12: Gráfica del resultado del tiempo de general informes desde el office

3.7. Pruebas de aceptación.

Las pruebas de aceptación representan aquella fase del ciclo de vida de desarrollo de *software* en que el equipo de desarrollo y los usuarios tienen que garantizar que el sistema desarrollado se corresponda con los requerimientos definidos en la fase de análisis. Es fundamental que la calidad tanto del código como de la documentación aportada al usuario sea alta y se corresponda con los parámetros adecuados para el *software* que concretamente se esté desarrollando (41).

Las pruebas de aceptación se realizan a partir de las historias de usuarios, donde cada una se convierte en un caso de prueba en el cual el cliente especifica los puntos a probar. Además, una historia de usuario pudiera tener más de una prueba de aceptación, pues se deben realizar las necesarias para garantizar la calidad del producto final. A continuación, se muestra un caso de prueba de ejemplo para la funcionalidad **Visualizar Página Principal**.

Tabla 16: Caso de prueba aceptación: Visualizar Página Principal

Caso de prueba de aceptación	
Código: HU2CP2.	Historia de usuario: La página principal muestra un resumen de todo el proceso.
Responsable de prueba: Iván Lázaro Isa Regueiro	
Descripción: Prueba para comprobar el estado de visualización del resumen de todo el proceso.	
Condiciones de ejecución: El usuario tiene que autenticarse en el sistema.	
Entrada/Pasos de ejecución:	
1- El usuario accede autenticándose en el sistema.	
2- El Usuario navega por toda la página principal visualizando los datos.	
Resultado esperado: La página principal muestra un resumen en datos y graficas del todo el proceso de liberación de Plasma.	
Evaluación de la prueba: Prueba satisfactoria.	

Una vez terminado este proceso el cliente quedó satisfecho, pues todos los casos de prueba se ejecutaron de forma satisfactoria.

3.8 Conclusiones del capítulo

En el desarrollo del presente capítulo se realizó el diagrama de despliegue lo cual permitió conocer la distribución física del sistema sobre una arquitectura de *hardware*. Se especificó el uso de los estándares de codificación para lograr obtener claridad y organización en el código fuente de la solución. Se realizaron pruebas unitarias, de interfaz de usuario y de aceptación a la herramienta para comprobar tanto el funcionamiento del código como la interfaz de la misma y los errores encontrados fueron tratados y corregidos en su totalidad.

Conclusiones Finales

Con el desarrollo de un módulo para la gestión del Control de la Calidad en la liberación del Plasma en los procesos de producción de farmacéuticos se da cumplimiento al objetivo general planteado. Para llegar a este resultado se concluye lo siguiente:

- Se realizó un estudio de las principales aplicaciones que realizan la gestión del Control de la Calidad en la liberación del Plasma en los procesos de producción de farmacéuticos, con el cual se pudo identificar la herramienta más adecuada para darle solución al problema de la investigación.
- Se realizó el análisis, diseño e implementación del módulo para la gestión del Control de la Calidad en la liberación del Plasma en los procesos de producción de farmacéuticos que cumple con las necesidades del cliente y elimina las vulnerabilidades antes existentes.
- Se diseñaron y se realizaron las pruebas de *software*, permitiendo la correcta comprobación del funcionamiento de la solución, dando la posibilidad de realizar el proceso de entrega del *software* a consideración del cliente.

Recomendaciones

Luego de finalizada la implementación de la aplicación y cumplidos los objetivos declarados en el inicio, con el objetivo de implementar una futura versión de ésta, se recomienda:

1. Incorporar nuevas funcionalidades al sistema, según surjan nuevos requisitos, con el objetivo de obtener un producto de software robusto e integral.
2. Incluir nuevos reportes en función de las necesidades de los usuarios funcionales de **Control de la Calidad** y perfeccionar los existentes, incluyendo el uso de gráficas u otras alternativas para mostrar la información de forma dinámica y útil.
3. Confeccionar un manual de usuarios detallado, que ayude a los usuarios novatos en el uso del módulo.

Referencias Bibliográficas

1. Guevara Plaza, A.J., Abad Grau, María del Mar. *Informática aplicada a la gestión de la empresa*. España : s.n., 2004. ISBN: 978-84-368-1854-3 84-368-1854-7..
2. Díaz, V.P. Impacto de las tecnologías de la información y las comunicaciones en la educación y nuevos paradigmas del enfoque educativo. 2011.
3. *El uso de las tecnologías de información y comunicación TIC en las micro, pequeñas y medianas empresas (MIPyME)*. Saavedra García, M., Tapia Sánchez, L. Blanca. 2013, Venezolana de Información, Tecnología y Conocimiento. ISSN: 1690-7515..
4. *La Investigación y Desarrollo en la Industria Farmacéutica: Pasado, presente y futuro*. Luengo, Juan Ignacio. Madrid : s.n., 2012, Fundación de Ciencias y Salud, Vol. 37, pág. 5. ISSN: 2174-8292..
5. Montalván, I.Á.Y. *Sistema informático interoperable para la industria farmacéutica del Ecuador*. Facultad de Ciencias Matemáticas y Física. Guayaquil, Ecuador : s.n., 2014.
6. PERÉZ, GASTON, GARCÍA, G AND N, IRMA. *Metodología de la Investigación Educativa*. La Habana : Pueblo y Educación, 1996.
7. CREA, Centro de Recursos para la Escritura. *Instituto Tecnológico y de Estudios Superiores de Monterrey*. [En línea] 2012. [Citado el: 02 de 10 de 2023.] http://sitios.ruv.itesm.mx/portales/crea/buscar/que/6_lospasos.htm.
8. Banc de Sang. [En línea] [Citado el: 29 de 08 de 2023.] <https://donarsang.gencat.cat/es/donacion-plasma/que-es/>.
9. Brech, E.F.L. *The Principles And Practice Of Management*. s.l. : L.P. Group, 1975. ISBN: 058245039.
10. TR Jain, O.K., M L Grover, R K Singla. *Industrial Sociology, Economics & Management*. 2 V.K. Enterprises. 2006-2007. ISBN: 81-88597-77-5.
11. Keith, T, Industria Farmaceutica. [En línea] 2012. [Citado el: 22 de 08 de 2023.] <http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/EnciclopediaOIT/tomo3/79.pdf>.
12. ERP Laboratorios Farmacéuticos. *COMPAS*. [En línea] [Citado el: 02 de 09 de 2023.] <https://compas.net/erp-laboratorios-farmaceuticos.html>.
13. KROHNE. [En línea] [Citado el: 02 de 09 de 2023.] <https://mx.krohne.com/es/soluciones/soluciones-monitorizacion/software-supervision-validacion/sistema-liberacion-gnl-control-calidad>.
14. SIEMENS. Sistema de Ejecución de Manufactura (MES). [En línea] [Citado el: 02 de 09 de 2023.] <http://www.industry.siemens.com/automation/aan/es/manufacturing-execution-system-mes/pages/default.aspx..>

15. EcuRed. [En línea] [Citado el: 02 de 09 de 2023.] Disponible en "tecnomatica.minbas.cu". <https://www.ecured.cu/MISTRAL>.
16. Jose H. Canos, Maria del Carmen Penades. *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n., 2012.
17. PHP. [En línea] [Citado el: 25 de 09 de 2023.] https://www.php.net/releases/5_6_0.php.
18. Jorge, Sanchez. *DESARROLLO WEB EN ENTORNO CLIENTE CON JAVASCRIPT*. España : GARCETA GRUPO EDITORIAL, 2019. pág. 456. ISBN: 9788417289409.
19. EcuRed. [En línea] [Citado el: 29 de 10 de 2023.] <https://www.ecured.cu/AJAX>.
20. ASP.NET Ajax : Enhanced Interactivity and Responsiveness. [En línea] [Citado el: 29 de 10 de 2023.] [https://docs.microsoft.com/en-us/aspnet/ajax/..](https://docs.microsoft.com/en-us/aspnet/ajax/)
21. EcuRed. [En línea] <https://www.ecured.cu/Framework>.
22. Laravel. [En línea] [Citado el: 29 de 10 de 2023.] <https://laravel.com/docs/5.5>.
23. Getbootstrap. [En línea] [Citado el: 29 de 10 de 2023.] [https://getbootstrap.com/.](https://getbootstrap.com/)
24. SMiLE Comunicación. [En línea] [Citado el: 29 de 10 de 2023.] Guía Bootstrap 5. <https://smilecomunicacion.com>.
25. Apache Sitio Oficial. [En línea] [Citado el: 28 de 10 de 2023.] [http://www.apache.org/..](http://www.apache.org/)
26. Ecured. [En línea] [Citado el: 30 de 10 de 2023.] https://www.ecured.cu/Servidor_HTTP_Apache.
27. Gilfillan, Ian. *MySQL, La Biblia de*. España : Kluwer Academic, 2003.
28. <https://www.jetbrains.com/es-es/phpstorm/features/>. JetBrains. [En línea] 10 de 8 de 2023. [Citado el: 26 de 10 de 2023.] <https://www.jetbrains.com/es-es/phpstorm/features/>.
29. Sitio web oficial de MySQL. [En línea] [Citado el: 30 de 10 de 2023.] [https://www.mysql.com/..](https://www.mysql.com/)
30. Apache Friends. [En línea] [Citado el: 28 de 10 de 2023.] <https://www.apachefriends.org/es/index.html>.
31. Visual Paradigm. [En línea] [Citado el: 25 de 09 de 2023.] www.visual-paradigm.com.
32. Ian, Addison-Wesley. *Somerville, Ingeniería de software*. 2011. Vol. Novena Edición.
33. Bajo de Luque, M.J. . Metodologías ágiles. Agile Alliance. [En línea] 23 de 2 de 2011. [Citado el: 16 de 09 de 2023.] <http://metodologiasagiles.herobo.com/index.php/es/2011-12-05-16-09-55/metodologia-xp>.
34. Adrformacion, Soluciones eLearning. [En línea] [Citado el: 29 de 10 de 2023.] https://www.google.com/amp/s/www.adrformacion.com/knowledge/ingenieria-y-proyectos/_que_es_una_arquitectura_cliente_servidor_.amp.html.

35. TABARES., RICARDO BOTERO. Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos. 2011, pág. 161 a 163. El presente artículo plantea una estrategia didáctica para introducir el uso de los patrones GRASP (General Responsibility).
36. Craig, L. *UML y patrones*. 2da Edición. Vancouver : s.n., 2004. ISBN:84-205-3438-2.
37. Alur, D., John, Crupi y Malks, Dan. Patterns Best Practices and Design Strategies. [En línea] [Citado el: 02 de 10 de 2023.] <ftp://ftp.hxu.edu.cn/pub/Develop/J2EE/Core%20J2EE%20Patterns.pdf>.
38. 2014, *Chilvisa Pallo*.
39. Pressman, Roger S. 2011, Ingeniería de Software. Un enfoque practico, editorial: MCGraw Hill, 7ma edicion, 2011. *Pressman*. 2011.
40. Ingeniería de Software Educativo, de Galvis [en línea]. Disponible. Bibliotecas UPLAP. [En línea] [Citado el: 19 de 05 de 2017.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo4.pdf.
41. José Ponce González, Francisco José Domínguez Mayo, Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental. *Documant*. [En línea] [Citado el: 01 de 09 de 2023.] <https://documat.unirioja.es/servlet/articulo?codigo=5413986>.
42. *Comunicación, Revista internacional de Información y*. 4, 2010, Vol. 19.

Anexos

Anexos Historias de Usuarios

Tabla 18: Historia de Usuario: Visualización de la realización del proceso

Historia de Usuario	
Número: HU4	Usuario: Administrador
Nombre de historia: Visualización de la realización del proceso	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 1	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: El usuario navega hacia el módulo <i>Control de la Calidad</i> y marca en el menú desplegable la opción <i>Inventario de Plasma</i>	
Observaciones:	

Tabla 19: Historia de Usuario: Visualizar plasma sin muestreo

Historia de Usuario	
Número: HU5	Usuario: Administrador
Nombre de historia: Visualizar plasma sin muestreo	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 1	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: En el menú <i>Inventario del Plasma</i> , se habilita un botón con nombre <i>Plasma Sin Muestreo</i> con acceso a una tabla que muestra la información	
Observaciones:	

Tabla 20: Historia de Usuario: Generar informe de banco de sangre sin muestreo

Historia de Usuario	
Número: HU6	Usuario: Administrador
Nombre de historia: Generar informe de banco de sangre sin muestreo	
Prioridad: Alta	Riesgo en desarrollo: Alta

Iteración: 2	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: Se muestra una tabla con la información del Plasma, y botón con la opción de <i>Generar Reporte</i>	
Observaciones: El usuario presiona en el botón <i>Generar Reporte</i> de color azul con el logotipo del Adobe Reader	

Tabla 21: Historia de Usuario: Visualizar plasma sin PCR con muestreo

Historia de Usuario	
Número: HU7	Usuario: Administrador
Nombre de historia: Visualizar plasma sin PCR con muestreo	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 2	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: En el menú <i>Inventario del Plasma</i> , se habilita un botón con nombre <i>Plasma Sin PCR con Muestreo</i> con acceso a una tabla que muestra la información	
Observaciones:	

Tabla 22: Historia de Usuario: Generar Informe de Plasma sin PCR con Muestreo

Historia de Usuario	
Número: HU8	Usuario: Administrador
Nombre de historia: Generar informe plasma sin PCR con muestreo	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 2	Punto estimado: 1
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: Se muestra una tabla con la información del Plasma, y botón con la opción de <i>Generar Reporte</i>	
Observaciones: El usuario presiona en el botón <i>Generar Reporte</i> de color azul con el logotipo del Adobe Reader	

Tabla 23: Historia de Usuario: Visualizar muestreo del plasma.

Historia de Usuario	
Número: HU9	Usuario: Control de la Calidad
Nombre de historia: Visualizar muestreo del plasma.	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 2	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: En el menú <i>Muestro del Plasma</i> , se visualiza una tabla con los diferentes lotes de plasma que están muestreados.	
Observaciones:	

Tabla 24: Historia de Usuario: Adicionar muestreo del plasma.

Historia de Usuario	
Número: HU11	Usuario: Control de la Calidad
Nombre de historia: Adicionar muestreo del plasma.	
Prioridad: Media	Riesgo en desarrollo: Alta
Iteración: 3	Punto estimado: 0.5
Programador responsable: Iván Lázaro Isa Regueiro	
Descripción: En el menú <i>Muestro del Plasma</i> , se visualiza una tabla con los diferentes lotes de plasma que están muestreados, además de un botón con nombre <i>Adicionar</i>	
Observaciones: El usuario presiona en el botón <i>Adicionar</i> y es direccionado por el sistema a otra pantalla, donde incluirá los datos necesarios para el nuevo muestreo, luego presiona en el botón <i>adicionar</i> y por último en el botón <i>guardar</i> .	

Anexos Prototipos de Interfaz de Usuario

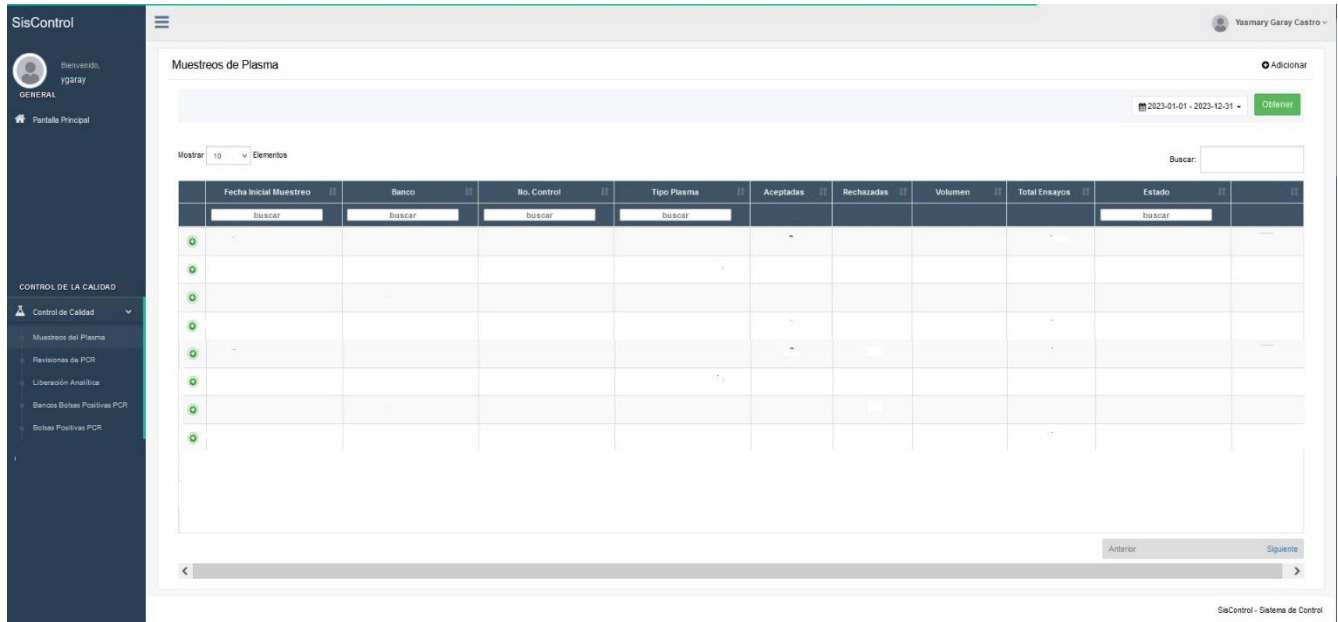


Ilustración 13: Prototipo de interfaz de usuario del tipo Evolutivo: Muestro del Plasma

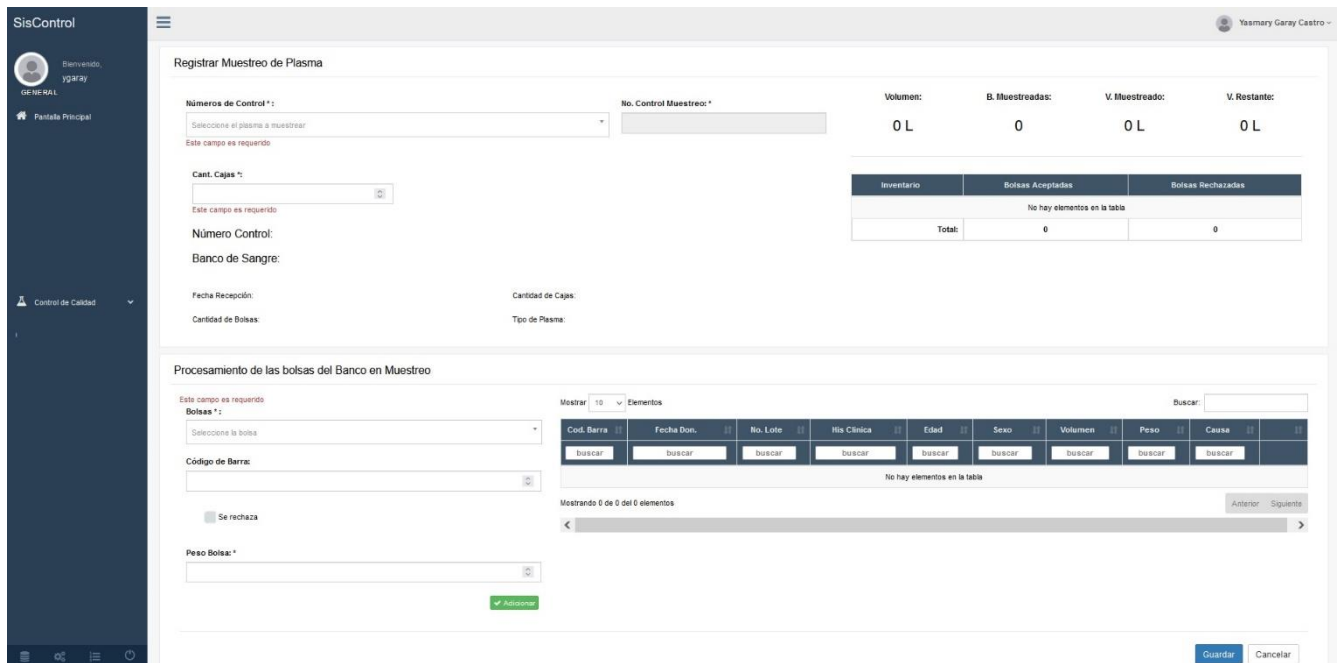


Ilustración 14: Prototipo de interfaz de usuario del tipo Evolutivo: Adicionar Muestro del Plasma

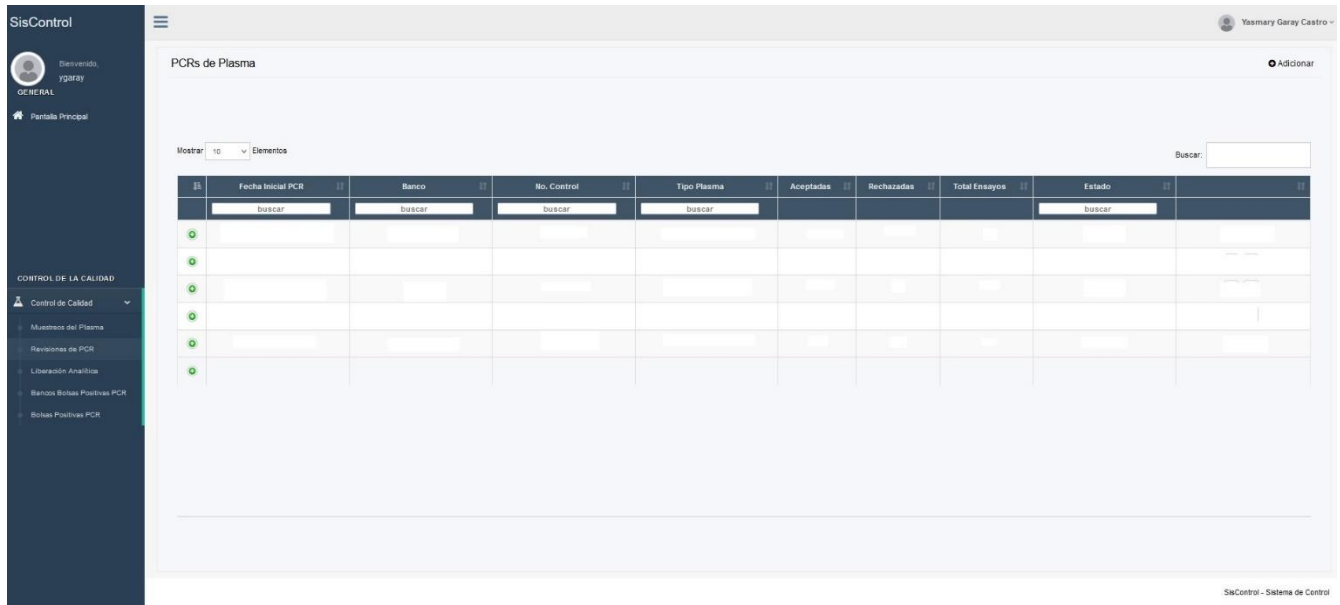


Ilustración 15: Prototipo de interfaz de usuario del tipo Evolutivo: Revisiones de PCR

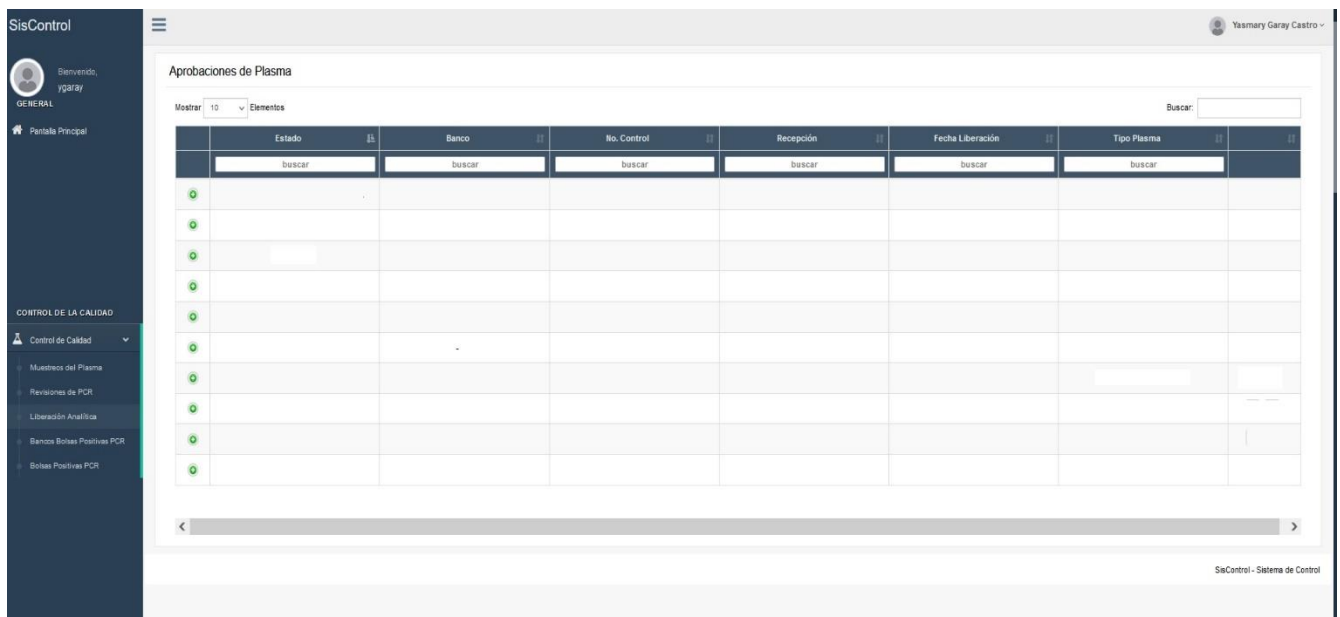


Ilustración 16: Prototipo de interfaz de usuario del tipo Evolutivo: Liberación Analítica