

UNIVERSIDAD DE MATANZAS

Facultad de Ciencias Técnicas

Departamento de Informática



Tesis en opción al título de ingeniero en Informática

Título:

" Aplicación web para la solicitud de artículos para los clientes de LaBiofam en Matanzas".

Autor: Andy Padrón González

Tutor: Enma Rizo

Matanzas, Cuba

julio, 2019

Declaración de autoría

**Declaro ser el único autor de este trabajo y autorizo al Departamento de Informática de la
Universidad de Matanzas a hacer uso del mismo.**

Para que así conste firmo la presente a los _____ de 2019

Firma del autor

Andy Padrón González

Dedicatoria

A mi mamá, por ser la mejor persona del mundo, por todo el amor y la dedicación que me
brindas.

Agradecimiento

- A mi mamá, por su amor y comprensión, por apoyarme siempre y por ser la mejor persona del mundo.
- A mis abuelos Esther y Roberto por ser los mejores abuelos, por ser un ejemplo a seguir y por todo el amor y cariño que me han brindado.
- A mi hermano, que es el mejor hermano y el que cualquier persona quisiera tener.
- A mi novia Dianelys, por estar siempre a mi lado, apoyarme en todo momento y por todo el amor y cariño que me brinda.
- A mi tía Belkis, por todo el empeño y dedicación hacia mí para que hoy pueda ser lo que soy.
- A mis tíos Marlen y Reinol, por estar siempre a mi lado y apoyarme en todo momento
- A Noel, Isabel, A yneé, Noelito, por apoyarme en todo momento y estar a mi lado.
- A Adalberto, Marialys, Orlando y Mery, por apoyarme en todo momento y confiar siempre en mí.
- A mis amigos de la universidad, en especial a: Orelbis y Alejandro por toda la ayuda que me brindaron durante la realización de este trabajo.
- A mi tutora Enma, por la ayuda brindada durante la realización de mi tesis.
- A todas aquellas personas que de una forma u otra contribuyeron a mi formación moral y profesional, a todos muchas gracias.

Resumen

Este Trabajo de Diploma propone una Aplicación web para mejorar la gestión de la solicitud de los productos para los clientes de LaBiofam en Matanzas. Para ello se empleó la metodología de desarrollo de software Scrum, se utilizó como lenguaje de programación para el lado del cliente TypeScript, y el framework Angular, el cual es extremadamente modular, ligero y fácil de aprender y como sistema de base de datos MongoDB. Se presenta la metodología de investigación empleada, las definiciones asociadas al objeto de estudio y al campo de acción. Se documentaron adecuadamente todas las etapas del ciclo de vida del software y se termina con las Pruebas de Aceptación que validan la funcionalidad del software, obteniéndose una aplicación web funcional que permite la solicitud de los artículos de LaBiofam.

Summary

This Diploma Work proposes a Web Application to improve the management of the product request for LaBiofam clients in Matanzas. For this, the software development methodology Scrum was used, it was used as a programming language for the TypeScript client side, and the Angular framework, which is extremely modular, lightweight and easy to learn and as a MongoDB database system. The research methodology used, the definitions associated with the object of study and the field of action are presented. All the stages of the software life cycle were properly documented and the Acceptance Tests validating the functionality of the software are finished, obtaining a functional web application that allows the request of the LaBiofam articles.

Contenido

Introducción	1
1. Capítulo I: MARCO TEÓRICO REFERENCIAL	5
1.1 1.1. Introducción	5
1.2 1.2 Objeto de estudio	5
1.3 1.3 Caracterización de LABIOFAM	5
1.4 1.4 Flujo actual de trabajo	6
1.5 1.5 Antecedentes del trabajo	6
1.6 1.6 Métodos de la investigación	6
1.7 1.7 Herramientas, tecnologías y metodologías de desarrollo	7
1.7.1 Arquitectura cliente-servidor:	8
1.7.2 N-capas:	8
1.7.3 Software libre	8
1.7.4 Patrones de diseño:	9
1.7.5 Patrón de arquitectónico:	9
1.8 Metodologías de desarrollo	10
1.9 Tecnologías y Herramientas	17
CCS	18
NodeJS	18
Herramientas utilizadas	20
CONCLUSIONES PARCIALES	21
CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA	22
2.1 Introducción:	22
2.2 Modelación del negocio	22
2.2.1 Diagrama de Casos de Uso del Negocio	22
2.2.2-Descripción de Actores del Negocio (elaboración propia):	23
2.2.3 -Descripción de Trabajadores del Negocio (elaboración propia):	23
2.2.4-Procesos del Negocio	24

Proceso: "autenticarse"	24
Proceso: "emitir queja" (elaboración propia):	28
Proceso: "emitir sugerencia" (elaboración propia):	30
2.2.5-Reglas del Negocio	31
2.3 Descripción de la solución	31
2.4 Roles de la metodología	32
2.5 Etapa de planificación	32
2.7 Definición de Sprint Backlog (elaboración propia):	36
2.8 Plan de entregas (elaboración propia):	39
2.9 Análisis de los costos	40
CONCLUSIONES PARCIALES	44
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	45
3.1 Introducción	45
3.2 Pruebas al software	45
3.2.1 Plan de pruebas	47
3.2.2 Pruebas de aceptación	49
3.2.3 Clases de equivalencia por cada atributo (elaboración propia):	49
3.2.4 -Tabla de Pruebas para crear empresa (elaboración propia):	53
3.3 Pruebas de sistemas web	57
3.4 Pruebas de Caja Blanca	58
3.5 Interfaces de la aplicación	61
Conclusiones Parciales	63
Conclusiones Generales	64
Recomendaciones	65
Bibliografía	66
Anexos	68

Introducción

La computación surgió para mejorar todos los procesos de la vida cotidiana, la acumulación de datos, los cálculos excesivamente grandes, y para contribuir en gran medida a la gestión y transmisión de la información, y tiene como propósito facilitar los procesos que requieren transformar, conservar o transmitir gran cantidad de información. Con el desarrollo de la ciencia y la tecnología ha aumentado la informatización de los medios socioeconómicos, evidenciándose en gran medida en la inmensa cantidad de sistemas informáticos que existen a nivel mundial, los cuales tienen disímiles funciones como el entretenimiento, en el caso de los juegos, la enseñanza de materias escolares y no escolares, en los procesos industriales a pequeña, mediana y gran escala, en los procesos militares, espaciales, en la economía y en la medicina, por así mencionar alguno de ellos.

De ahí la importancia de la informatización de los sistemas como una vía que facilite a los responsables de los proyectos la toma de decisiones.

Desde finales del siglo xx con el surgimiento y puesta en práctica del sistema operativo Windows se comenzó a emplear la informática utilizando paquetes como los de Microsoft Office, que brindan un grupo de facilidades entre las que se puede encontrar los procesadores de texto, las hojas de cálculo electrónicas y gestores de mensajería.

En Cuba a partir de inicios del siglo XXI, a la par del desarrollo mundial, se empiezan a utilizar las nuevas Tecnologías de la Informática y las Comunicaciones (TIC), trazándose una política en el país encaminada a la informatización de la sociedad, y el Comandante en Jefe Fidel Castro funda la Universidad de las Ciencias Informáticas (UCI) con el objetivo de formar profesionales de este sector y convertirse en núcleo de la futura industria del software cubana, produciéndose en dicho centro un grupo de aplicaciones informáticas para aspectos de la esfera académica, pero sobre todo son para la instrucción.

Con el avance de las Tecnologías de la Información y las Comunicaciones en el mundo se ha visto un auge en el desarrollo de aplicaciones informáticas que automatizan procesos relacionados con distintas actividades, facilitando la organización, ejecución y control en la gestión de estas. Un ejemplo de estas son las llamadas tiendas en línea o tiendas virtuales que se refiere a un tipo de comercio que usa como medio principal para realizar transacciones en un sitio web o una aplicación conectada a una red. Los vendedores de productos ponen a disposición de sus clientes una página web (o aplicación informática) en la que se pueden observar imágenes de los productos, leer sus especificaciones, su precio entre otras características y finalmente, adquirirlos. Este servicio le da al cliente rapidez en la compra, la posibilidad de hacerlo desde cualquier lugar y a cualquier hora del día.

En Cuba con el desarrollo de las tecnologías informáticas se han puesto en marcha diferentes sitios web para la gestión de artículos, para que así sean más asequible para los clientes, ejemplo de esto es la aplicación del Cimex DondeHay, que su principal uso es para que los clientes sepan dónde encontrar los artículos que necesitan y puedan conocer su precio con anterioridad, antes de dirigirse al centro de venta.

LABIOFAM SA se estableció el 27 de septiembre de 1993 a solicitud del entonces Grupo de Laboratorios Biológicos y Farmacéuticos LABIOFAM. Desde 1996, ha estado comercializando las producciones y servicios del Grupo Empresarial LABIOFAM en el extranjero a través de diferentes canales, de modo que actualmente mantiene vínculos comerciales con más de 95 países del mundo. Su misión es proporcionar servicios de importación y exportación que permitan aumentar la capacidad productiva, con un destino de salud animal, vegetal y humana, con una capacidad laboral constituida principalmente por científicos e ingenieros en laboratorios y plantas de producción.

Comercializa productos importados, así como productos naturales, suplementos dietéticos, biotecnológicos, farmacéuticos, agroquímicos, químicos y plásticos, producidos por las entidades del Grupo LABIOFAM, y proporciona servicios de saneamiento ambiental.

Para solicitud de artículos en la empresa LaBiofam en la provincia de Matanzas, los clientes tienen que dirigirse personalmente hacia las sucursales más cercanas y realizar el pedido de lo que necesitan, incurriendo en gastos materiales en cada gestión que se realice y lo convierte en un proceso lento y engorroso por todos los trámites que se necesitan para la compra de artículos por parte de clientes mayoristas.

Todo lo cual deriva en el siguiente **problema científico** a resolver, ¿cómo mejorar el proceso de gestión de los planes de producción de la empresa LaBiofam en la provincia de Matanzas?

Gestión de los planes de producción de LaBiofam es el **objeto de estudio**.

Teniendo en cuenta el problema se propone trabajar en la siguiente **hipótesis**:

Si se desarrolla una aplicación web para la gestión de los planes de producción de la empresa LaBiofam en la provincia de Matanzas entonces disminuye el tiempo de gestión de los procesos de los planes de producción de la empresa LaBiofam en la provincia de Matanzas

El **objetivo general** de la investigación es desarrollar una aplicación web para facilitar la gestión de los planes de solicitud de los clientes de LaBiofam en la provincia de Matanzas.

Los **objetivos específicos** son:

- Desarrollar un prototipo de aplicación funcional para gestión de los planes de producción de LaBiofam .
- Utilizar una metodología para el desarrollo del software .
- Validar la propuesta implementada con la pertinencia del software .
- Analizar y seleccionar las herramientas para el desarrollo del software .
- Disminuir el tiempo de gestión en la solicitud de productos .

Para el desarrollo del software se utilizaron diversos métodos y técnicas tales como:

Dentro de los métodos teóricos:

- Método de análisis histórico - lógico .
- Método de análisis y síntesis .
- Método inductivo - deductivo .

Como métodos empíricos, utilizados por medio de las siguientes técnicas:

- Observación .
- Entrevistas .
- Análisis de documentos .

La investigación se estructuró en 3 capítulos, como se indica a continuación

Capítulo 1. Marco Teórico - Referencial: Se plantean las definiciones fundamentales asociadas al campo de acción. Se realiza un estudio sobre el estado de la planificación de la empresa, las tendencias en las herramientas, tecnologías y metodologías actuales que se escogieron para el desarrollo de la solución y se exponen sus principales características y su utilidad para la solución propuesta.

Capítulo 2. Análisis, diseño y construcción de la solución propuesta: Se argumenta la solución que se propone, mostrando una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software SCRUM. Se desarrolla la propuesta para darle solución a la situación problemática, presentando una planificación por iteraciones.

Capítulo 3. Validación de la Solución Propuesta: Se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio de los clientes y los propios de la metodología de software. Se realiza además un estudio de los beneficios tangibles e intangibles de la realización del software.

Finalmente, se presentan las Conclusiones y Recomendaciones de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

De igual manera, quedan recogidas las Bibliografías y Anexos empleados que fueron necesarios para el desarrollo del trabajo y un mejor entendimiento del mismo

1. Capítulo I: MARCO TEÓRICO REFERENCIAL

1.1 1.1. Introducción

En el desarrollo de este capítulo se abordan los antecedentes de este trabajo y se presentan las bases teóricas que sustentan la investigación. Son mencionados aspectos tecnológicos y científicos que permiten enfocar la atención en los conocimientos necesarios que se tuvieron en cuenta. Se realiza un análisis del estado del negocio, tomándose en consideración diferentes herramientas y sistemas existentes con una finalidad similar, para proceder a su estudio y analizar posibles mecanismos de implementación.

1.2 1.2 Objeto de estudio

El objeto de estudio es la gestión de solicitud de productos a LABIOFAM, por tanto, surge la necesidad de hacer una descripción más detallada de estos elementos que se pretenden transformar y perfeccionar.

1.3 1.3 Caracterización de LABIOFAM

LABIOFAM SA se estableció el 27 de septiembre de 1993 a solicitud del entonces Grupo de Laboratorios Biológicos y Farmacéuticos LABIOFAM. Desde 1996, ha estado comercializando las producciones y servicios del Grupo Empresarial LABIOFAM en el extranjero a través de diferentes canales, de modo que actualmente mantiene vínculos comerciales con más de 95 países del mundo. Su misión es proporcionar servicios de importación y exportación que permitan aumentar la capacidad productiva, con un destino de salud animal, vegetal y humana, con una capacidad laboral constituida principalmente por científicos e ingenieros en laboratorios y plantas de producción.

Comercializa productos importados, así como productos naturales, suplementos dietéticos, biotecnológicos, farmacéuticos, agroquímicos, químicos y plásticos, producidos por las entidades del Grupo LABIOFAM, y proporciona servicios de saneamiento ambiental.

LABIOFAM SA ofrece una amplia gama de productos exportables a sus clientes y socios comerciales, siendo Vidatox 30CH, Biorat y Productos Naturales los que tienen el mayor potencial, incluidos servicios como programas de control de vectores, ya implementados en varias regiones del mundo con excelentes resultados.

1.4 1.4 Flujo actual de trabajo.

La gestión de los planes de producción en la empresa se realiza de forma manual por el encargado, el cual tiene que revisar todas las peticiones de los clientes y compararlas con la información que obtiene desde el almacén y así poder dar respuesta a la necesidad de los clientes. Una de las tareas más importantes es poder gestionar de manera rápida los planes de solicitud y mostrarle al cliente la información que necesita inmediatamente de su petición y poder presentarle un material agradable a la vista y que sea interactivo.

Analizando el estado actual del negocio se pudo comprobar que hasta el momento se realizan los procesos de gestión de solicitud de artículos de forma manual cuyas herramientas básicas eran los libros de registro, contratos y el libro de planes de solicitud de artículos y venta, dado que el cliente tiene que dirigirse hasta la institución varias veces para cerrar un contrato, se incurre en gastos de combustible y tiempo, lo que ha traído consigo un constante manejo de toda la información necesaria y a su vez les ha proporcionado demora en la venta de los productos.

1.5 1.5 Antecedentes del trabajo.

Previo a la elaboración de este trabajo se realizó una investigación con el fin de conocer acerca de la existencia de sistemas o aplicaciones informáticas similares al propuesto.

A pesar de la existencia de aplicaciones *web* relacionados con este proceso, la Empresa ha decidido desarrollar una aplicación personalizada, de acuerdo con sus intereses, que capture la información necesaria para ser almacenada en una base de datos. Con esto se logra el acceso rápido a la misma, y registrar todas las operaciones requeridas para generar reportes de salidas con formatos específicos y datos imprescindibles, a fin de realizar consultas de interés y contribuir a la toma de decisiones y agilización de los procesos de gestión.

1.6 1.6 Métodos de la investigación

Para el logro de los objetivos de este trabajo se hace necesaria la investigación a través de diferentes métodos tanto teóricos como empíricos que permitan el desarrollo de este proceso.

Los métodos teóricos fundamentales utilizados fueron:

- Método Inductivo Deductivo.
- Método Histórico.
- Analítico-sintético

Y los empíricos fueron:

- Observación.
- Entrevistas.
- Análisis de documentos.

El método Inductivo y Deductivo fue utilizado mediante el razonamiento, que va de lo particular a lo general. Se profundizó y valoró hasta donde podía ser posible la construcción de la aplicación, cuáles eran las salidas de las informaciones, y cómo se respondía al cumplimiento de los objetivos de la investigación.

El método analítico-sintético se utilizó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas. Hipotético-deductivo: Desempeñó un papel esencial en el proceso de verificación

El método Histórico: se estudió la evolución y desarrollo del objeto o fenómeno de la investigación, y se analizó la trayectoria del mismo a lo largo de la historia.

La observación científica ha acompañado la investigación desde sus primeros momentos, principalmente en su desarrollo e implementación. Se ha empleado tanto la observación directa como la indirecta de aspectos importantes de la práctica diaria, propios de la actividad.

Como método para recolectar datos es muy similar al análisis de contenido, es una forma de observación del contenido de comunicaciones.

La entrevista refuerza la necesidad de introducir los resultados de este trabajo de investigación, esencialmente a partir de los criterios aportados por el cliente y los tutores.

El método de análisis de documentos se empleó en la revisión de documentos importantes aportados por los tutores y los presentes en internet, que impulsaron el desarrollo de la presente propuesta.

1.7 1.7 Herramientas, tecnologías y metodologías de desarrollo

Para desarrollar la investigación, se hace necesario el estudio de las herramientas, tecnologías y metodologías de desarrollo para darle cumplimiento al objetivo general. Seguidamente se mencionan aspectos a tener en cuenta.

1.7.1 Arquitectura cliente-servidor:

Aquí se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de cliente. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un Sistema Gestor de Base de Datos (SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red (Peña, 2007).

1.7.2 N-capas:

Según (Castro, Llorente, & Press, 2010) el objetivo de la arquitectura n-capas es separar física y lógicamente los componentes mínimos de una aplicación. Las principales ventajas son:

- Clara separación de las funciones del control de la interfaz y presentación de datos con la lógica de la aplicación.
- Reusabilidad de componentes.
- Independencia de la interfaz del cliente y la arquitectura de datos.
- Mejores posibilidades de balancear la carga.
- Uso de protocolos abiertos.

En la Figura 1 se muestra la estructura de la arquitectura n-capas. En la capa de interfaz de usuario se realiza la presentación de interfaces de usuario, la captura de datos y la validación de datos del usuario. En la capa de negocio el modelo del negocio, la invocación a la capa de modelo de acceso a datos y la recepción de las solicitudes de la capa de interfaz de usuario. La capa de datos es la encargada del almacenamiento y recuperación de datos, así como la interacción con la capa de negocio.

1.7.3 Software libre

El término software libre se refiere a un programa de ordenador con libertad para su utilización, distribución, modificación y estudio. Desde el punto de vista técnico-legal, se considera software libre a los programas licenciados en términos que garantizan a sus usuarios el derecho de ejecutarlos, copiarlos, distribuirlos, estudiarlos, cambiarlos y mejorarlos. (Zanotti, 2016).

1.7.4 Patrones de diseño:

Los patrones de diseño son un conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Estos identifican y especifican abstracciones que van más allá del simple ámbito de clases e instancias, o componentes. Para que una solución sea considerada un patrón debe poseer ciertas características como son efectividad habiendo resuelto problemas similares en ocasiones anteriores y reusabilidad permitiendo su aplicación a diferentes problemas de diseño en distintas circunstancias. (García, 2016).

1.7.5 Patrón de arquitectónico:

1.7.5.1 Modelo Vista Controlador

El modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica del negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Según (Fernández, 2012) el Modelo Vista Controlador es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido. MVC consiste en dividir el código de una aplicación en capas:

Modelo: es todo acceso a datos y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo.

Vista: en una aplicación Web, es el HTML y lo necesario para convertir datos en HTML. O sea, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.

Controlador: es lo que une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través del modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen, gestiona las entradas del usuario. Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que "hace algo" está perfectamente separado del código dedicado a crear HTML.

1.8 Metodologías de desarrollo

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos que deben seguirse para llevar a cabo el desarrollo de software con calidad. Estas brindan un conjunto de detalles organizativos añadiendo rigurosidad y normas permitiendo que los integrantes de un equipo de desarrollo puedan seguir un criterio común a la hora de realizar las tareas durante el desarrollo de un software. La constante innovación tecnológica hace que cada vez sea más necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos. (RocketTheme, 2017)

Una metodología de desarrollo es un conjunto de métodos que especifican quién debe hacer qué cosa, cuándo debe hacerse, estableciendo un conjunto de roles (el quién), actividades (la cosa) y un ciclo de vida que establece las diferentes fases. (RocketTheme, 2017).

Según (Pressman, 2010) la ingeniería de software ágil combina una filosofía con un conjunto de lineamientos de desarrollo:

- La filosofía enfatiza en: La satisfacción del cliente y en la entrega rápida de software incremental, los equipos pequeños y muy motivados para efectuar el proyecto, los métodos informales, los productos del trabajo con mínima ingeniería de software y la sencillez general en el desarrollo.
- Los lineamientos de desarrollo enfatizan en: La entrega sobre el análisis y el diseño y la comunicación activa y continua entre desarrolladores y clientes.

A través de los años han surgido diferentes metodologías. Algunas hacen más énfasis en el control del proceso, estableciendo rigurosamente las actividades y los artefactos que se deben producir, las herramientas y notaciones a utilizar, las cuales se denominan metodologías tradicionales, que son utilizadas para grandes proyectos; y otras denominadas metodologías ágiles se centran más en las personas que en el proceso, dan mayor valor al individuo, al equipo de desarrollo y a su

colaboración con el cliente; proponiendo un desarrollo incremental del software con iteraciones muy cortas y la documentación necesaria. (André Ampuero & Muñoz Castillo, 2008).

Las metodologías ágiles de desarrollo de software han despertado interés gracias a que proponen simplicidad y velocidad para crear sistemas.

Las metodologías tradicionales no se adaptan a las nuevas necesidades o expectativas que tienen los usuarios, en parte porque los métodos usados no son flexibles ante la posibilidad de la exigencia de nuevos requerimientos. (Highsmith & C., 2017) (Cao & otros, 2016) .

Estos cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto que se esté llevando; en contraparte, los métodos ágiles permiten un desarrollo iterativo y adaptable que permite la integración de nuevas funcionalidades a lo largo del desarrollo del proyecto; para que tanto el cliente como el desarrollador queden satisfechos porque el producto final tiene una calidad adecuada. (Giraldo, 2016) .

El enfoque tradicional de desarrollo de aplicaciones heredado de otras ramas de la ingeniería, ha sido el de hacer un estudio exhaustivo del problema, establecer un plan y llevar a cabo la construcción. Es la metodología conocida como desarrollo en cascada, con las fases consecutivas de Análisis, Diseño, Codificación, Pruebas e Implementación.

El objetivo de las metodologías de desarrollo ágil de software es la organización de un trabajo creativo, que suele ser bastante caótico. Se intenta dar prioridad a la ejecución sobre la planificación. A medida que se profundiza en el conocimiento de un problema se cambian los planes. Cuando el cliente vea nuestras propuestas, se le ocurrirán nuevas ideas que cambiarán los planes. Cuando profundicemos en el conocimiento de nuevas tecnologías haremos descubrimientos que cambiarán de nuevo los planes.

Ágil quiere decir, adaptable. Desde esta premisa, los cambios son bienvenidos, derivan de un mejor entendimiento del problema y son una oportunidad para mejorar el software. (Blanch, 2010).

Surgen a mediados de los años 90, ante la necesidad de una nueva forma de desarrollar software diferente a las que hasta el momento se habían utilizado (metodologías tradicionales). Se derivan de la lista de los principios que se encuentran en el "Manifiesto Ágil" que están basados en un desarrollo iterativo que se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones de diferente longitud, cada una de ellas generando un producto completo y entregable; donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto, las interacciones individuales deben producir alguna característica completamente funcional o mejorada, su principal objetivo es reducir el tiempo de desarrollo. ("Manifiesto por el Desarrollo Ágil de Software," s.f).

El manifiesto ágil plantea que es necesario valorar más a los individuos y su interacción que a los procesos y las herramientas. Las herramientas mejoran la eficiencia, pero sin personas con conocimiento técnico y actitud adecuada, no producen resultado. (Letelier et al., 2015).

También afirma que en la mayoría de las ocasiones se crea el entorno y luego se espera a que el equipo se adapte a él, lo cual constituye un error que debe ser erradicado, ya que debe ser el equipo y no de forma contraria, el que en base a sus necesidades sea capaz de crear el entorno. Destaca la importancia de poder ver anticipadamente cómo se comportan las funcionalidades esperadas sobre prototipos o sobre las partes ya elaboradas del sistema final. El manifiesto ágil define al cliente como un miembro más del equipo, que se integra y colabora en el grupo de trabajo. (José H, 2015).

Según (José H, 2015) los principales valores de la gestión ágil son la anticipación y la adaptación; diferentes a los de la gestión de proyectos ortodoxa: planificación y control para evitar desviaciones sobre el plan, los principios son:

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

En cuanto a la superioridad de las metodologías no ágiles con respecto a las que lo son se considera que para lograr hacer frente a cualquier proyecto y se obtengan resultados favorables no se puede establecer una metodología única. Éstas deben ser adaptadas al contexto es decir los recursos humanos y técnicos, el tiempo de desarrollo, el tipo de sistema, es decir depende de lo que se desee lograr es necesario hacer un estudio para aplicar la metodología correcta.

Las metodologías ágiles de desarrollo de software proponen una estrecha relación entre el cliente y el equipo de desarrollo, hacer entregas funcionales del software continuamente, se aplican a proyectos con requerimientos cambiantes o imprecisos, y están dirigidas fundamentalmente para equipos pequeños.

Diversas han sido las metodologías ágiles que han surgido en los últimos años, y como es de esperar algunas más populares que otras. En una encuesta realizada en el 2006 por CM Crossroads en Estados Unidos se obtuvo como resultado que la metodología más utilizada era XP, seguida por FDD, Scrum, Crystal y DSDM. (CM Crossroads, 2006).

Tabla 1: Comparación entre metodologías ágiles y tradicionales. (Sánchez, 2003).

Metodología Ágil	Metodología Tradicional
Pocos artefactos.	Más artefactos.
Pocos roles, más genéricos y flexibles.	Más roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Se hace énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Se hace énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.

producción de código.	
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios

Tras haber realizado una comparación de ambas Metodologías de realización del software, se observa que las Ágiles se ajustan más para conducir el desarrollo del software propuesto, debido a que este presenta pocos artefactos y roles, hay poco tiempo de desarrollo y es mucho más flexible con los cambios en el software.

Entre las metodologías ágiles más conocidas se encuentran: Scrum, Crystal Metodologies y Extreme Programming (XP). Se selecciona entre las metodologías ágiles Scrum para desarrollar e implementar la solución planteada.

Con esta metodología se van a crear manuales básicos, que permite presentar y aclarar todos los procesos y procedimientos necesarios a seguir, y en los que el equipo SCRUM tiene que regirse para terminar con éxito esta solución en los tiempos estimados.

Justificación para su uso en el proyecto:

Lo ágil de esta metodología, permite ir presentando avances para revisar, probar, lo que no se definió al inicio y confirmar los controles con los usuarios finales para que queden aprobados; esto evita retrasos, de otra forma recién al final del desarrollo de la solución planteada se empieza a revisar y probar a ver cuáles son los problemas y lo que no se definió al inicio.

¿Por qué utilizar Scrum?

Porque es un enfoque de gestión ágil que facilita la administración de proyectos, programas y portafolios de cualquier tamaño y complejidad, facilitando el flujo de información, la comunicación entre el equipo de trabajo y la entrega de valor con oportunidad a los interesados de la organización.

Algunas de las principales ventajas de utilizar SCRUM son:

Transparencia: Todas las fuentes de información del proyecto tales como el Scrumboard y el Sprint Burndown Chart son compartidos, lo que lleva a un ambiente de trabajo abierto.

Adaptabilidad: El control empírico de los procesos y las entregas continuas hacen que los proyectos sean adaptables y abiertos a la incorporación del cambio.

Mejora Continua: Los entregables se mejoran progresivamente Sprint por Sprint a través de un

proceso de definición y priorización de entregables y sus características (Groom Prioritized Product Backlog).

Retroalimentación Continua: A través de reuniones de seguimiento diario y reuniones para mostrar y validar los entregables, se proporciona retroalimentación continua al cliente. (Conduct Daily Standup y Demonstrate and Validate Sprint).

Ritmo de trabajo sostenible: Los procesos Scrum están diseñados de tal manera que las personas involucradas pueden trabajar a un ritmo de trabajo cómodo, (Sustainable Pace), y ayudan a que este ritmo de trabajo continúe indefinidamente.

Entrega Anticipada de Alto Valor: Scrum cuenta con procesos que aseguran que los requisitos de mayor valor al cliente sean los primeros en ser satisfechos. Este proceso se llama (Create Prioritized Product Backlog)

Proceso de Desarrollo Eficiente: El definir periodos específicos de trabajo y reducir al mínimo de trabajo que no es esencial conduce a mayores niveles de eficiencia (Time Boxing).

Según (Kniberg, 2017) las ventajas del uso de Scrum son:

- Scrum puede ser adoptado para manejar las prácticas de ingeniería que se utilizan en una organización, ya que no requiere ninguna práctica de ingeniería específica.
- Scrum puede mejorar la comunicación entre todos los equipos.
- Los problemas pueden ser transparentes es decir se hace fácil la detección de errores gracias a la entrega de los nuevos módulos o complementos entregados en cada sprint.
- Durante el sprint el equipo de Scrum se mantiene centrado y creativo sin interferencias. Lo cual es bueno para la productividad, ya que en este proceso se desarrolla lo que se ha tomado en cuenta en reuniones previas y se busca satisfacer las necesidades del cliente.

A modo de resumen Scrum es un marco de trabajo que da libertad en cómo implementarlo.

Empezar a usar Scrum es fácil y aplicarlo en el desarrollo de proyectos aporta beneficios no sólo a la empresa que fabrica el producto sino al cliente y al equipo de desarrollo. El impacto que producen los cambios en el desarrollo es menor para la empresa, el cliente puede empezar a usar

las partes más valiosas del producto antes empezando así a recuperar parte de la inversión. Ver y usar parte del producto ayuda al cliente a darse cuenta más rápido de los cambios necesarios, y al ser detectados antes tendrán menor impacto y coste en el desarrollo. El equipo se auto gestiona y cada persona puede crecer profesionalmente adquiriendo conocimiento y aportando valor en múltiples áreas del proyecto. (<http://www.enzyme.es/es/blog/7-razones-para-usar-scrum-en-el-desarrollo-de-proyectos>)

Según (Casco, 2017) los principios de SCRUM son:

- Eliminar el desperdicio: No perder tiempo haciendo cosas sin valor para el cliente.
- Construir la calidad con el producto: Inyectar la calidad directamente desde el principio.
- Crear conocimiento: No se puede tener conocimiento desde antes de empezar el proyecto.
- Diferir decisiones: Dejar las decisiones para el momento que se necesiten tomarlas, puesto que se puede tener más información en ese momento.
- Entregar rápido: Da competencia.
- Respetar a las personas: Respetar a todo el equipo, permite motivar.
- Optimizar el todo: Optimizar cada proceso para mejorar continuamente.

Según (Requerimientos, Técnicas de Especificación y metodologías Ágiles, 2017) los artefactos del SCRUM son:

- Product Backlog: es la lista maestra que contiene toda la funcionalidad deseada en el producto. La característica más importante es que la funcionalidad se encuentra ordenada por un orden de prioridad.
- Sprint Backlog: es la lista que contiene toda la funcionalidad que el equipo se comprometió a desarrollar durante un Sprint determinado.
- Burndown Chart: muestra un acumulativo del trabajo hecho, día-a-día.

1.9 Tecnologías y Herramientas

Lenguajes de programación web:

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. (Domínguez, 2014).

También se utiliza principalmente para crear páginas *web* dinámicas. Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con *JavaScript* se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, legalmente JavaScript es una marca registrada de la empresa Sun Microsystems. (Hightower, 2016) (Pérez, 2016)

TypeScript

HTML5

El HTML5 (Hyper Text Markup Language, versión 5) es la quinta revisión del lenguaje de programación de la World Wide Web, el HTML. Esta nueva versión reemplaza al actual (X)HTML, corrige los problemas que los desarrolladores web encuentran, así como rediseñan el código, la presente versión se actualiza en función de las nuevas necesidades que demanda la web en la actualidad. (Cantón, 2017).

Según (Olan, 2016) es una actualización de HTML que es el lenguaje en el que se crea la *web*. HTML5 además es un término de *marketing* para agrupar las nuevas tecnologías de desarrollo de aplicaciones *web*: HTML5, CSS3 y nuevas capacidades de *JavaScript*.

Puede ser considerado como piedra angular de la Web Semántica. Presenta una serie de ventajas frente al HTML tradicional, como la capacidad de ordenar semánticamente el contenido del documento con etiquetas como *nav*, *header*, *section*, *footer*; incrustar directamente elementos multimedia como audios, videos y canvas 2d y 3d; un nuevo grupo de tipos de entrada de datos para formularios con validación sin JavaScript; soporte de etiquetas para manejo de grandes cantidades de datos (*Datagrid*, *Details*, *Menu* y *Command*) que permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en el cliente, (en dependencia del nivel de implementación de HTML5 del navegador utilizado) (Consortium, 2014).

C C S

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o 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 ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los *Estilos* definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

N o d e J S

Es un intérprete *JavaScript* del lado del servidor que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir código que maneje decenas de miles de conexiones simultáneas en sólo una máquina física. (Foundation, 2016) .

NodeJS proporciona mecanismos para escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas), integración con motores de renderización (generación de imágenes) de "vistas" para generar respuestas mediante la introducción de datos en plantillas, establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta. (Colaboradores, 2018) .

Frameworks:

Un framework constituye una estructura de soporte definida (o plataforma de desarrollo enfocada al entorno web) en la cual otro proyecto de software puede ser organizado y desarrollado. (Alcalde, 2013) .

E x p r e s s

Proporciona mecanismos para escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas), integración con motores de renderización (generación de imágenes) de "vistas" para generar respuestas mediante la introducción de datos en plantillas, establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta. (Colaboradores, 2018)

A n g u l a r

Una de las ventajas del desarrollo en Angular está dada por el hecho de que cuando un nuevo programador retome un proyecto de Angular se siente como en casa de forma rápida, ya que si

conoce Angular conoce la manera en la que se hacen las cosas con este marco de trabajo. Esto facilita también el intercambio de programadores entre proyectos. (MVP, 2017).

Bootstrap

Según (Risueño, 2015) Bootstrap es un framework que permite montar una estructura responsive fácilmente, se dice fácilmente porque muchas clases y funciones están desarrolladas. Además, incorpora muchas librerías como normalice, jQuery, less, por lo que se puede hacer uso de muchos efectos y funciones sin tener que programar de cero. También permite conseguir un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones. Características principales de Bootstrap

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Incorpora un sistema de grid basado en 12 columnas fluidas para poder acoplarse en función del dispositivo desde el que se acceda.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo, jQuery.
- Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTMLShim para que reconozca los tags HTML5.

Sistemas de Gestión de Base de Datos (SGBD):

Los Sistemas Gestores de Bases de Datos (SGBD, en inglés DBMS: DataBase Management System) son software que permiten 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 (Marcano, 2009).

MongoDB

Como gestor de base de datos se usó MongoDB (del inglés humongous, "enorme"). Este es un sistema de base de datos NOSQL, orientado a documentos de código abierto. Las bases de datos no relacionales surgen de la necesidad de tener sistemas de bases de datos **altamente disponibles, distribuidos y escalables horizontalmente**, es decir si se necesita un campo más en un registro en específico, no es necesario remodelar toda la tabla, los datos guardados con MongoDB son de fácil lectura para los usuarios, menor tiempo de recuperación ante fallas y permite hacer consultas más rápidas. (Viviana, 2019).

Herramientas utilizadas

Microsoft Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto,¹ aunque la descarga oficial está bajo software propietario.

Herramienta para el modelado.

Como herramienta para el modelado de la solución se ha seleccionado Visual Paradigm for UML (VP-UML v8.0), una de las líderes del mercado de las llamadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, según sus siglas en inglés).

VP-UML v8.0 soporta los principales estándares de la industria tales como el Lenguaje de Modelado Unificado (UML, según sus siglas en inglés), SysML, BPMN, XMI, entre otros. Entre las bondades de VP destacan: ofrecer un conjunto completo de herramientas; ofreciendo a los equipos de desarrollo de software todo lo necesario para la captura de requisitos, planificación de software, planificación de controles, modelado de clases y modelado de datos, así como brindar interoperabilidad entre diagramas ya que es capaz de exportar los diagramas de un modelo a otro con mucha facilidad, ahorrando de esta manera tiempo, lo cual es crucial para el desarrollo.

CONCLUSIONES PARCIALES

En este capítulo se abordaron temas útiles para emprender el desarrollo de la propuesta. Se realizó un análisis de la importancia de contar con un medio que permita realizar la gestión de los planes de producción más rápidamente. Para ello se llevó a cabo un estudio de cómo funcionan en la actualidad estos procesos. Durante la preparación de este capítulo ha sido expuesta la metodología ágil Scrum para guiar el desarrollo del software y las tecnologías a emplear en el sistema.

Dentro de las herramientas de desarrollo e implementación se utilizaron TypeScript y JavaScript como lenguaje de programación, Angular como frameworks para facilitar el desarrollo de las aplicaciones, NodeJs como entorno en tiempo de ejecución para la capa del servidor, Visual Studio Code como entorno de desarrollo, MongoDB como gestor de Base de Datos y Node/Express como servidor Web.

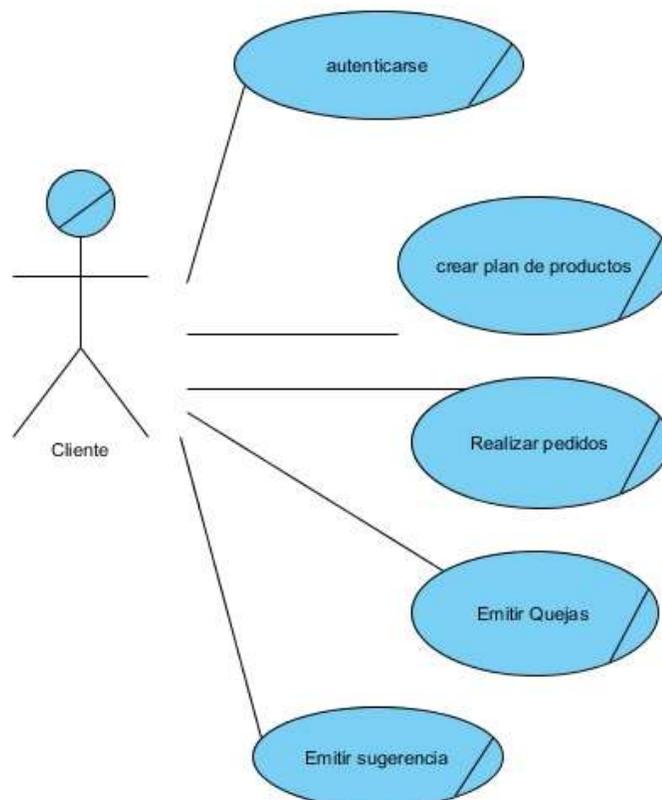
CAPÍTULO 2: ANÁLISIS, DISEÑO Y DESARROLLO DE LA SOLUCIÓN PROPUESTA

2.1 Introducción:

En el presente capítulo se representa la solución de la propuesta, regida por la metodología ágil de desarrollo de software SCRUM; que involucra al cliente en el proceso de desarrollo, mejorando la comunicación del mismo con el equipo y retroalimenta el código desarrollado; tiene como principal objetivo la satisfacción del cliente. Todo proyecto de software combina diferentes etapas y la primera es el levantamiento de requisitos. Teniendo en cuenta la utilización de SCRUM como metodología, se han empleado las Historias de Usuario (HU) para extender las necesidades descritas por el cliente en la Pila del Producto. Además, se realiza la planificación inicial del proyecto junto a un estudio de factibilidad, analizando para ello los costos y beneficios, para establecer si resulta factible o no el desarrollo de la aplicación, también se realiza la modelación del negocio utilizando UML.

2.2 Modelación del negocio

2.2.1 Diagrama de Casos de Uso del Negocio



2.2.2-Descripción de Actores del Negocio (elaboración propia):

Actor del negocio	Descripción
<i>Cliente</i>	El cliente es aquella persona que inicializa el negocio (creando el plan de solicitudes, realizando una solicitud o emitiendo una queja o sugerencia), pidiéndole al recepcionista los modelos correspondientes en cada caso (lista de productos, listado de planes anuales o libro de quejas y sugerencias).

2.2.3 -Descripción de Trabajadores del Negocio (elaboración propia):

Trabajador del negocio	Descripción
<i>Recepcionista</i>	El (la) Recepcionista es aquel trabajador del negocio que se encarga de recibir al cliente, gestionar toda la información referente al mismo, y solicitar los servicios necesarios para satisfacer las necesidades del mismo.

Trabajador del negocio	Descripción
Gestor de quejas y sugerencias	Es el encargado de analizar la solicitud de queja o sugerencia y tomar una decisión de acuerdo con lo analizado.

Trabajador del negocio	Descripción
<i>Director</i>	Es el director de la entidad, es el encargado de dirigir y participar en todos los procesos que se llevan a cabo en la entidad. Es también quien verifica y aprueba las soluciones que se le dan a las quejas y sugerencias.

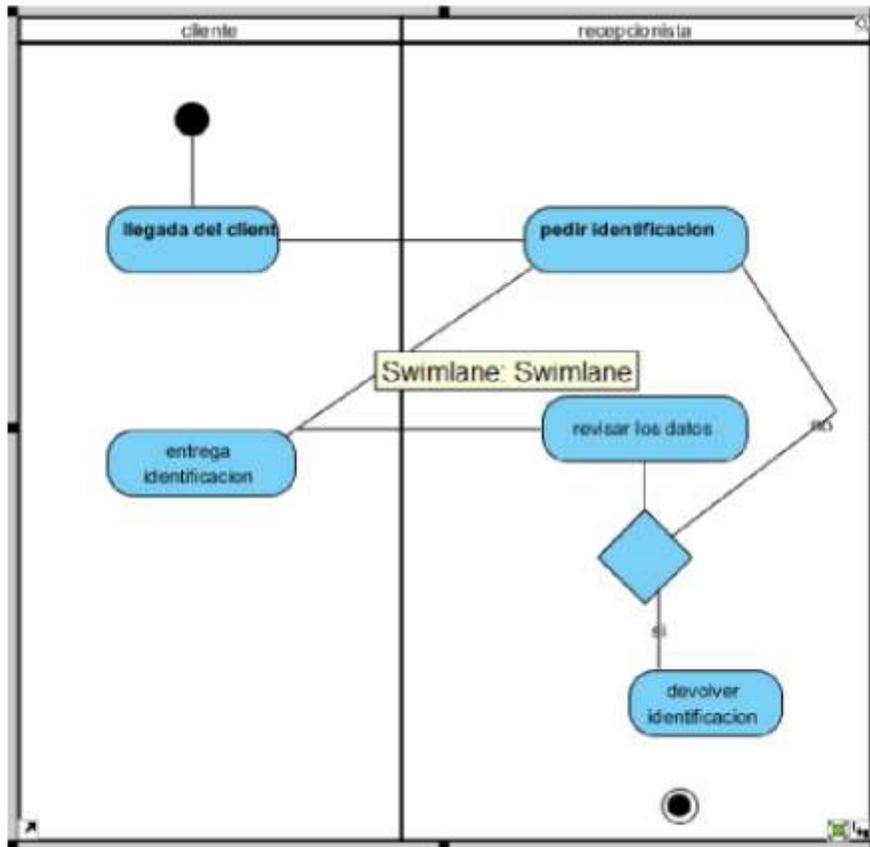
2.2.4-Procesos del Negocio

Proceso: "autenticarse"

Descripción del Caso de Uso (elaboración propia):

Caso de uso del negocio	autenticarse
Actores	<i>C l i e n t e</i>
Resumen	El caso de uso se inicia cuando el cliente llega a la tienda y este es atendido por el recepcionista. El caso de uso culmina cuando el cliente se identifica
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<p>1.Llega a la recepción de la tienda</p> <p>3. entrega su identificación</p>	<p>2- El Recepcionista recibe al cliente y le pide su identificación.</p> <p>4- El Recepcionista revisa si los datos son correctos.</p> <p>5-Si son correctos se archivan los datos para su posterior procesamiento.</p> <p>6-Si no son correctos le devuelve la identificación al cliente y se repite el proceso.</p>
Otras secciones	
Mejoras propuestas	

Diagrama de Actividades:



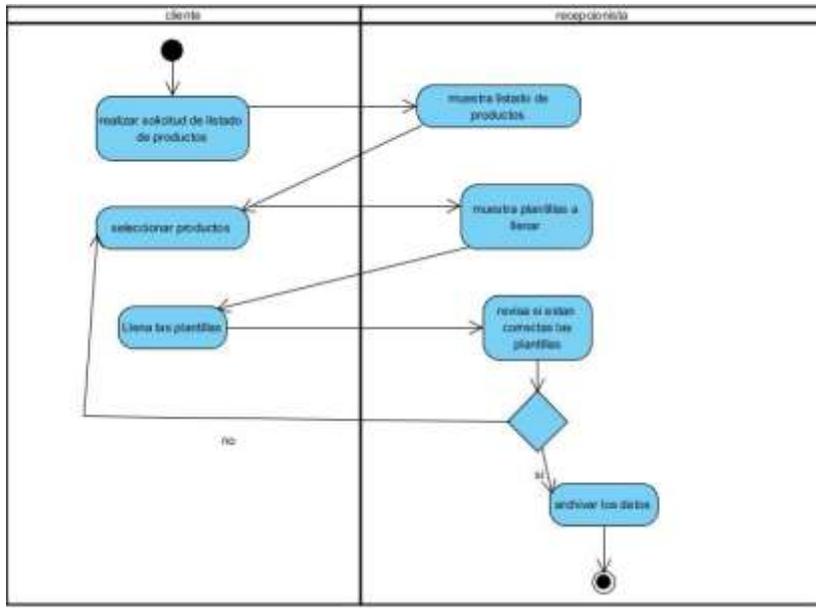
Proceso: "realizar pedidos"

Descripción del Caso de Uso (elaboración propia):

Caso de uso del negocio	<i>realizar pedidos</i>
Actores	<i>Cliente</i>
Resumen	El caso de uso se inicia cuando el cliente hace una solicitud. Este es atendido por la recepcionista que le da los listados de los productos que se encuentran a la venta. El caso de uso termina el cliente realiza el pedido del producto
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio

<p>1.El cliente realiza una solicitud.</p> <p>3. selecciona los productos que desea reservar, llena las planillas correspondientes y las entrega al recepcionista.</p>	<p>2- El Recepcionista muestra el listado de los productos a la venta y las planillas a llenar en caso que seleccione alguno.</p> <p>4- El Recepcionista revisa si los datos de las planillas son correctos.</p> <p>5-Si son correctos se archivan los datos para su posterior procesamiento.</p> <p>6-Si no son correctos le devuelve las planillas al cliente y se repite el proceso</p>
<p>Otras secciones</p>	
<p>Mejoras propuestas</p>	

Diagrama de Actividades:

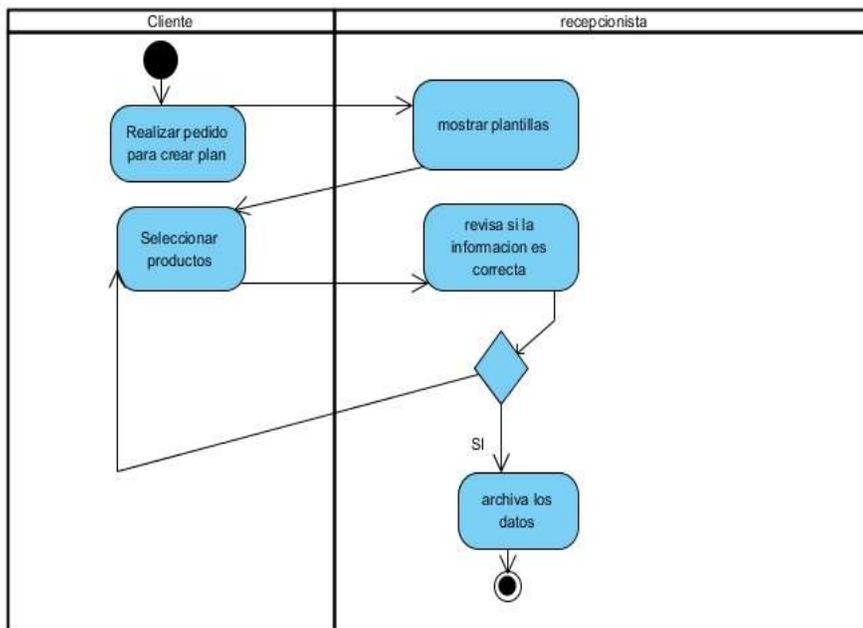


Proceso: "Crear plan"

Descripción del Caso de Uso (elaboración propia):

Caso de uso del negocio	Crear plan
Actores	Cliente
Resumen	El caso de uso se inicia cuando el cliente se le solicita crear el plan de solicitudes. El caso de uso termina cuando el cliente deja por escrito los productos que necesita.
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio

Diagrama de Actividades:

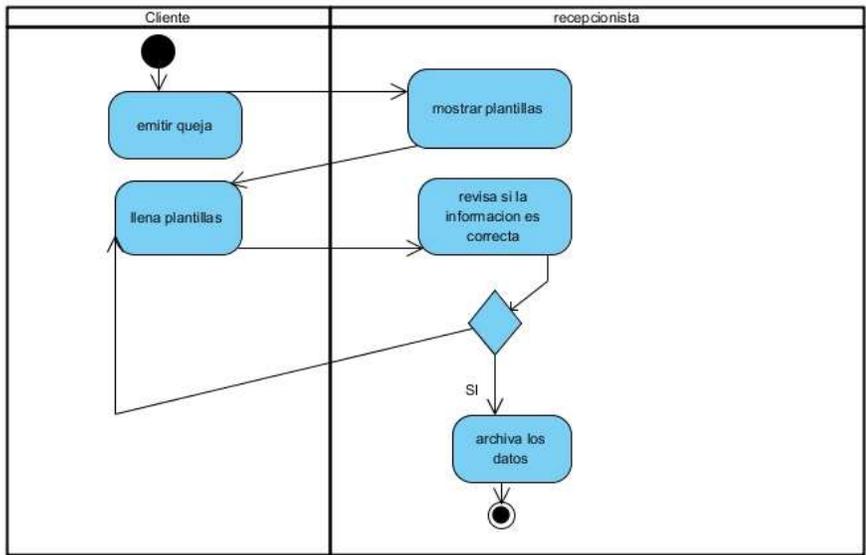


Proceso: “emitir queja” (elaboración propia):

Caso de uso del negocio	<i>Em itir que jas</i>
Actores	<i>C lie nte</i>
Resumen	El caso de uso se inicia cuando el cliente decide emitir una queja a la empresa .El caso de uso termina cuando el cliente deja por escrito la inconformidad .
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio

<p>1. Emitir queja</p> <p>3. Llena la planilla con los datos correspondientes</p>	<p>2- El Recepcionista le muestra la plantilla de las quejas</p> <p>4-Si son correctos se archivan los datos para su posterior procesamiento.</p> <p>5-Si no son correctos le devuelve las planillas al cliente y se repite el proceso</p>
<p>Otras secciones</p>	
<p>Mejoras propuestas</p>	

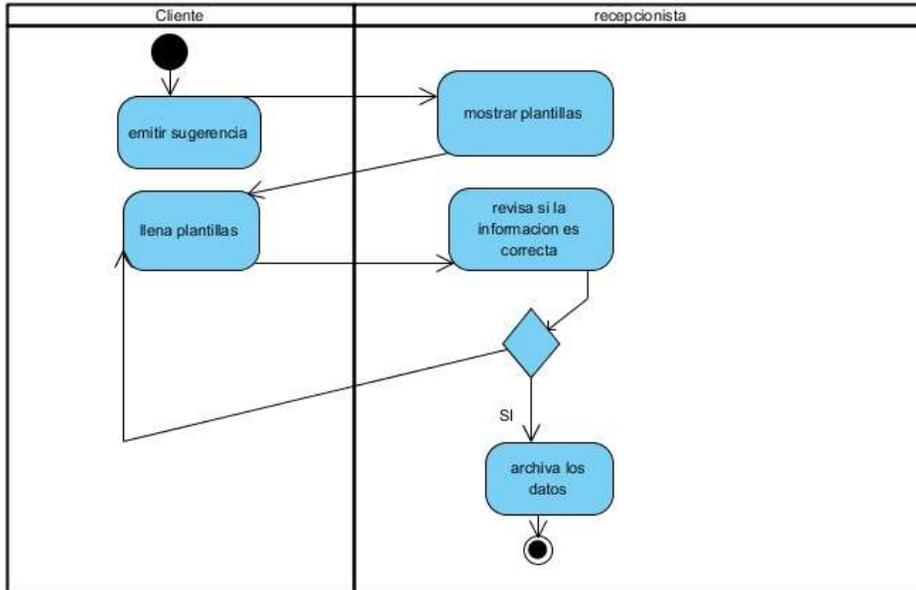
Diagrama de Actividades:



Proceso: “em itir sugerencia” (elaboración propia):

Caso de uso del negocio	<i>Em itir sugerencia</i>
Actores	<i>Cliente</i>
Resumen	El caso de uso se inicia cuando el cliente decide emitir una sugerencia a la empresa .El caso de uso termina cuando el cliente deja por escrito la inconformidad .
Casos de uso asociados	
Acción del actor	Respuesta del proceso de negocio
<p><i>1.Em itir sugerencia</i></p> <p><i>3.Llena la planilla con los datos correspondientes</i></p>	<p>2- El Recepcionista le muestra la plantilla de las sugerencias</p> <p>4-Si son correctos se archivan los datos para su posterior procesamiento.</p> <p>5-Si no son correctos le devuelve las planillas al cliente y se repite el proceso</p>
O tras secciones	
Mejoras propuestas	

Diagrama de Actividades:



2.2.5-Reglas del Negocio

- a) Primeramente, cuando se hace alguna solicitud se debe verificar que el cliente se encuentra registrado, si no lo esta se debe registrar en el formulario de clientes.
- b) Cuando va a realizar un plan de solicitudes debe quedar aprobado por el director de la empresa
- c) En el caso de presentar una queja, se comprueba la validez de la identificación que el cliente presenta y con ello se prosigue con resto de los pasos.

2.3 Descripción de la solución

Se propone el desarrollo de una aplicación web que les permita a los trabajadores de LaBiofam mejorar la gestión de la solicitud de productos. Además, se almacena y se gestiona el plan mensual de cada cliente, así como la planificación de los siguientes meses si así lo desean. De esta manera facilitará el acceso a la información, la seguridad, la rapidez y eficiencia de los datos.

Los usuarios que interactúan con la aplicación tendrán los siguientes roles:

Administrador del sistema: Tiene la tarea de asignar permisos, crear los usuarios y configurar la aplicación.

Usuario registrado: Trabaja directamente con el sistema, es el cliente que creara sus planes de producción y luego reservará sus productos cuando lo desee.

Usuario no registrado: Trabaja con el sistema, pero solo puede verificar los productos existentes.

Para garantizar la seguridad y la confiabilidad en la información que se procesa es de gran importancia implementar una aplicación web que permita la autenticación para identificar los usuarios que utilizan el programa.

2.4 Roles de la metodología

La metodología SCRUM define tres roles que rigen el proceso de desarrollo, ellos son:

- Cliente o Dueño del Producto (Product Owner)
- Facilitador (SCRUM Máster o SCRUM Manager)
- Equipo (Team)

Para la designación de los roles en el presente trabajo se adaptaron las definiciones expuestas en el capítulo anterior, en vista de que ha sido desarrollado por una única persona que asumirá los roles de Facilitador y Equipo. Esto trae como consecuencia la pérdida de la relación entre estos, que se considera importante para el desarrollo del software; pero a diferencia de otras metodologías, SCRUM permitió transformar el flujo de trabajo y adaptando las condiciones del entorno a su metodología.

2.5 Etapa de planificación

La planificación se realiza con el objetivo de lograr eficiencia en la realización del prototipo inicial que se está realizando, y así lograr un buen comienzo. Con este objetivo y según las ideas del cliente sobre el software se desarrollará la Pila del Producto, mediante la cual se obtendrá un punto de partida para el resto de la planificación del proyecto. Igualmente se realizará un estimado de cada una de las entregas del proyecto y del tiempo, basándose en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto, que esto se desglosaría en los Sprint y sus descripciones.

La pila de producto es una lista priorizada de requisitos, o historias, o funcionalidades, cosas que el cliente quiere, descritas usando la terminología del cliente. A esto se le llama historias, o simplemente elementos de la Pila. (Kniberg, 2017).

El Product Backlog es la pila que contiene todas las funcionalidades deseadas del producto. Al iniciarse un proyecto el Product Owner generalmente no determina el esfuerzo que demanda el equipo para desarrollar sus requisitos, por lo que en la Sprint Planning Meeting (Reunión de

planificación del Sprint) presenta al equipo los puntos que definió en el Product Backlog y le asigna una prioridad. El equipo establece las funcionalidades que estarán presentes en el siguiente Sprint y las pasa al Sprint Backlog, debe comenzarse por los requisitos de más prioridad en la pila del producto.

Las historias incluyen los siguientes campos:

- ID – un identificador único, simplemente un número auto-incremental. Esto nos permite no perder la pista a las historias cuando cambiamos su nombre o el atributo que las identifica.
- Nombre – una descripción corta de la historia, suficientemente claro como para se comprenda a de qué se está hablando y para distinguirla de las otras historias. Normalmente, 2 a 10 palabras.
- Importancia – importancia que posee la historia. Por ejemplo, 10 o 150. Más alto es igual a más importante
- Estimación inicial – la valoración inicial (en semanas) del equipo acerca de cuanto trabajo es necesario para implementar la historia, comparada con otras historias.
- Como probarlo – descripción a alto nivel de cómo se demostrará esta historia en la Demo al final del Sprint. Se trata, esencialmente, de una simple especificación de un test: "Haz esto, entonces haz lo otro, y entonces debería ocurrir aquello".

Product Backlog o Pila del producto (elaboración propia):

Id	Nombre	Importancia	Estimación	Como Probarlo	Notas	No del Sprint
1	Diseño y creación de la base de datos	15	1	Se verifica la introducción de datos en la base de datos y se le realiza consultas a la misma	Realizada a partir de un diagrama de entidad relación.	2
2	Diseño y creación de la interfaz de usuario	20	1	Verificar si la interfaz se acopla con las necesidades del cliente	Se debe ajustar a las necesidades del cliente	3

3	Gestionar nomencladores: -Provincia -Categorías de productos -Categoría de quejas -Categoría de sugerencias	10	1	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		4
4	Gestionar Usuarios	20	2	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.	Los usuarios deben pertenece solamente a una empresa	
5	Gestionar Empresas	15	2	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.	Las empresas no pueden tener más de un usuario	
6	Gestionar Artículos	20	3	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		

7	Gestionar Quejas	10	1	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		
8	Gestionar Sugerencias	10	1	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		
9	Gestionar Plan de producción	20	5	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		
10	Gestionar Sistema de autenticación	20	4	Verificar el acceso de cada rol al sistema.	Solo los usuarios autenticados pueden realizar pedidos en el sistema	
11	Implementar Trazas	15	1	Verificar que se inserten en la BD las acciones de	Se obtienen los movimientos del usuario	

				los usuarios	en la aplicación	
12	Pedir artículos	20	3	Verificar si son insertados, modificados o eliminados correctamente los datos en la BD.		

2.7 Definición de Sprint Backlog (elaboración propia):

Para la definición de la planeación se utilizan los métodos empíricos de la metodología Scrum, basándose en las técnicas de selección por, "ojo de buen cubero", la cuales viable para proyectos pequeños, con pocas iteraciones, donde un error en la planeación no repercute en el producto final, debido a que, si no se cumple el sprint, se pasa la tarea para el siguiente sprint. (Kniberg, 2017).

No.1 Nombre: Entrevista con los trabajadores de LaBiofam			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
07 enero 2019	11 enero 2019	1:40pm -4:30pm	15 min
Descripción: Durante la reunión con el cliente se realizaron preguntas a modo de entrevistas para determinar que necesita el cliente y los requisitos funcionales y no funcionales			

No.2 Nombre: Diseño y creación de la base de datos e interfaz de usuario			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
14 enero 2019	25 enero 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - Se diseña y crea la base de datos usando el gestor MongoDB. - Se diseña y crea la interfaz de usuario utilizando el framework de css Bootstrap			

No.3 Nombre: Nomencladores, quejas y sugerencias:			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
04 marzo 2019	11 marzo 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - - Se implementa: <ul style="list-style-type: none"> • Crear, modificar y eliminar nomencladores • Crear, modificar y eliminar quejas • Crear, modificar y eliminar sugerencias 			

No.4 Nombre: Gestión de usuarios:			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
12 marzo 2019	22 marzo 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - Se implementa: <ul style="list-style-type: none"> • Crear, modificar y eliminar usuarios 			

No.5 Nombre: Empresas y artículos:			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
25 marzo 2019	05 abril 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - Se implementa:			
<ul style="list-style-type: none"> • Crear, modificar y eliminar empresas • Crear, modificar y eliminar artículos 			

No.6 Nombre: Plan de producción y trazas:			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
08 abril 2019	19 abril 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - Se implementa:			
<ul style="list-style-type: none"> • Crear, modificar y eliminar plan de producción • Sistema de trazas 			

No.7 Nombre: Fase de pruebas:			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
22 abril 2019	03 mayo 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - - Se realizan pruebas de software.			

No.8 Nombre: Implementación del software			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
04 mayo 2019	22 mayo 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - Se implanta el software en la empresa. - Se imparten cursos de superación para el manejo de los usuarios con la aplicación.			

No.9 Nombre: Pedir artículos			
Encargado :Andy Padrón González			
Fecha Inicial	Fecha Final	Tiempo de Trabajo	Tiempo de Descanso
22 mayo 2019	22 mayo 2019	8:00am -4:30pm	30 min cada 3:00h
Descripción: - se piden los artículos del mes correspondiente			

2.8 Plan de entregas (elaboración propia):

Sprint	Fecha de Inicio	Fecha de Fin
Entrevista con los trabajadores de LaBiofam	07 enero 2019	11 enero 2019
Diseño y creación de la base de datos e interfaz de usuario	14 enero 2019	25 enero 2019
Nomencladores, quejas y sugerencias	04 marzo 2019	11 marzo 2019
Gestión de usuarios	12 marzo 2019	22 marzo 2019
Empresas y artículos	25 marzo 2019	05 abril 2019
Plan de producción y trazas	08 abril 2019	19 abril 2019
Fase de pruebas	22 abril 2019	03 mayo 2019

Pedir artículos	04 mayo 2019	22 mayo 2019
-----------------	--------------	--------------

2.9 Análisis de los costos

Principales Conceptos en el Método por Puntos de Función

- EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier interno.
- EO: Procesos en los que se envía datos al exterior de la aplicación.
- EQ: Procesos consistentes en la combinación de una entrada y una salida en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.
- ILF: Grupos de datos relacionados entre sí, internos al sistema.
- EIF: Grupos de datos que se mantienen externamente.
- PFTe: Total Puntos de Función para las entradas del sistema.
- PFTo: Total Puntos de Función para las salidas del sistema.
- PFTq: Total Puntos de Función para las consultas del sistema.
- PFTif: Total Puntos de Función para los archivos internos del sistema.
- PFTef: Total Puntos de Función para los archivos externos del sistema

Tablas de Valores

EI- EQ

CLASIFICACION DE SALIDAS	1-5 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 o 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 3 ficheros accedidos	MEDIA 5	ALTA 7	ALTA 7

Tabla 2

EO

CLASIFICACION DE ENTRADAS Y CONSULTAS	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

Tabla 1

ILF

FICHEROS LÓGICOS INTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 5 Entidades o registros lógicos	MEDIA 10	ALTA 15	ALTA 15

Tabla 3

E I F

FICHEROS LÓGICOS EXTERNOS	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro lógico	BAJA 5	BAJA 5	MEDIA 7
2 - 5 Entidades o registros lógicos	BAJA 5	MEDIA 7	ALTA 10
Más de 5 Entidades o registros lógicos	MEDIA 7	ALTA 10	ALTA 10

Tabla 4

La información que resulte de estas tablas se pondera en una tabla general la cual describe el componente y el nivel en el que se encuentra. Determinando el peso de cada componente

Componente	Bajo	Medio	Alto	Total
E I	$E_b * 3 = _$	$4 * 4 = _$	$5 * 6 = _30$	$PFT_e = 46$
E O	$O_b * 4 = 16$	$O_m * 5 = _$	$O_a * 7 = _$	$PFT_o = 16$
E Q	$10 * 3 = 3$	$2 * 4 = _$	$Q_a * 6 = _$	$PFT_q = 38$
I L F	$IF_b * 7 = _$	$IF_m * 10 = _$	$IF_a * 15 = _$	$PFT_{if} = 15$
E I F	$E F_b * 5 = _$	$E F_m * 7 = _$	$E F_a * 10 = _$	$PFT_{ef} = 0$
				$PFS_A = 115$

$$PFS_A = PFT_e + PFT_o + PFT_q + PFT_{if} + PFT_{ef}$$

$$PFS_A = 46 + 16 + 38 + 15 + 0$$

$$PFS_A = 115$$

Luego de obtener los puntos de función sin ajustar, debemos calificar cada uno de los factores de valor de ajuste, utilizando una escala del 0 al 5 con el siguiente desglose:

0 → sin influencia

1 → influencia incidental

2 → influencia moderada

3 → influencia media

4 → influencia significativa

5 → fuerte influencia en toda la aplicación

Calificamos cada uno de los 14 Ítem y sumamos los grados de influencia (TDI) para obtener el factor de complejidad técnica (FCT):

Ítem	TDI
1. Comunicación de datos	5
2. Proceso distribuido de datos	2
3. Desempeño	4
4. Configuración	3
5. Volumen de transacciones	4
6. Captura de datos en Línea	5
7. Eficiencia al usuario final	5
8. Actualización de Datos en Línea	4
9. Complejidad	4
10. Reusabilidad	3
11. Facilidad de instalación	4
12. Facilidad de operación	4
13. Instalación Múltiple	2
14. Facilidad de cambio	2
FCT	51

Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0,65 + (0,01 * FCT)]$$

$$PFA = 115 * [0,65 + (0,01 * 51)]$$

$$PFA = 115 * [0,65 + 0,51]$$

$$PFA = 115 * 1,16$$

$$PFA = 133,4$$

Líneas de código (LC)

$$LC = PFA * (\text{Líneas} * PF)$$

$$LC = 133.4 * 100$$

$$LC = 13340$$

Esfuerzo hora/persona

$$E = PFA / (1/8 \text{ persona/hora})$$

$$E = PFA / (1/5) \quad 1 \text{ persona trabaja } 5 \text{ h}$$

$$E = 133.4 / 0.2$$

$$E = 667 \text{ horas/persona}$$

Tomando 24 días laborables en el mes y 8 horas productivas al día, obtenemos 192 horas laborables al mes.

Duración del proyecto en meses

$$667 \text{ horas/persona} / 1 \text{ persona} = 667 \text{ horas}$$

$$DM = 667 \text{ horas} / 192 \text{ horas/mes}$$

DM = 4,75 aproximadamente 5 meses + 1 mes y 1 semana por el margen de error de la estimación por puntos de función y 1 mes para la fase de prueba

$$DM = \text{aproximadamente } 6 \text{ meses}$$

Costo total del proyecto

$$CT = \text{suelo de } 1 \text{ persona} / \text{cant de personas} * DM$$

$$CT = (500/1) * 6$$

$$CT = 3000$$

CONCLUSIONES PARCIALES

- La solución propuesta responde a los requerimientos de los procesos de gestión de la planificación de los planes de producción y la solicitud de artículos.
- En la pila de producto se listan 11 funcionalidades agrupadas en 12 sprint.

En este capítulo se da una muestra de cómo fue el desarrollo del software mediante la ingeniería de este. Con las diferentes tablas se fue describiendo el proceso, para un mejor entendimiento de las funcionalidades del mismo. La estimación realizada establece que es posible realizar el desarrollo de la solución dentro del periodo enmarcado

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En este capítulo se realizan las pruebas al software que permiten demostrar la calidad del producto, lo que constituye uno de los pasos más importantes en el desarrollo de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con el fin de encontrar, si existieran, algunos errores no descubiertos durante el desarrollo del software.

3.2 Pruebas al software

Según (González, 2017) el proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

La prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Una estrategia de pruebas debe incluir tanto pruebas de alto como de bajo nivel. Son parte de la Verificación y Validación incluidas en el aseguramiento de la calidad del software.

Verificación: Comprobar que el software está de acuerdo con su especificación, donde se debe comprobar que satisface tanto los requerimientos funcionales como los no funcionales.

Validación: El objetivo es asegurar que el software satisface las expectativas del cliente.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

Prueba de Unidad: verifican que el componente funciona correctamente a partir del ingreso de diferentes casos de prueba. Los errores más comunes son detectados en estas pruebas.

- Se examinan las estructuras de datos locales
- Se prueban condiciones límite.
- Se ejercitan todos los caminos independientes. Se utiliza un controlador independiente para cada caso. Este es un programa que recibe las pruebas, las envía al módulo y muestra el resultado.

Prueba de Integración: verifican que los componentes trabajan correctamente en forma conjunta.

pág. 46

- Se toman los componentes que han pasado las pruebas de unidad y se los combina.
- Estos test sirven ya que los datos podrían perderse en alguna interfaz
- . La combinación de los mismos podría traer efectos que no son los esperados.

La integración puede ser:

- Descendente: Inician por el programa principal. En profundidad: Integra todos los módulos de un camino de control principal de la estructura. En anchura: Incorpora todos los módulos directamente subordinados a cada nivel.
- Ascendente: Se empieza la prueba con los módulos atómicos. Datos que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible, pero no así los conductores.

Prueba de Sistema: verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

Pruebas de recuperación: Se controla que el software se recupere ante fallas.

Generalmente se fuerza el fallo.

Pruebas de seguridad: Se comprueban los mecanismos de protección integrados.

Pruebas de resistencia: Se diseñan para enfrentar a los programas a situaciones anormales.

Pruebas de rendimiento: Se prueba el sistema en tiempo de ejecución. A veces va emparejada con la prueba de resistencia.

Prueba de Aceptación: Proporcionan una seguridad final de que el software satisface los requerimientos.

Revisión de la configuración: Asegurar que todos los elementos de la configuración del software se hallan desarrollado apropiadamente.

Pruebas de aceptación (ALFA y BETA): Las realiza el usuario final en lugar del responsable del desarrollo.

o Pruebas ALFA: Desarrolladores con clientes antes de liberar el producto. o

Pruebas BETA: Seleccionando los clientes que efectuarán la prueba. El desarrollador no se encuentra presente.

Prueba de Seguridad: verifica que el software, no presente ninguna vulnerabilidad

Pruebas de Autenticación: Autenticar un objeto puede significar confirmar su procedencia, mientras que autenticar a una persona consiste a menudo en verificar su identidad.

3.2.1 Plan de pruebas

Según (Loaiza, 2010) el plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales.

Resumen las pruebas que se realizan a la aplicación web (elaboración propia):

No	Pila del producto	Pruebas a realizar
1	Diseño y creación de la base de datos	Test de base de datos

2	Diseño y creación de la interfaz de usuario	Test diseño de la interfaz de usuario
3	Gestionar nomencladores (Provincias, categoría de productos, categoría de quejas o sugerencias)	-Test de Insertar nomencladores. - Test de Modificar nomencladores. -Test de Eliminar nomencladores. -Test de Listar nomencladores.
4	Gestionar Plan de producción	-Test de Insertar Plan de producción. - Test de Modificar Plan de producción. -Test de Eliminar Plan de producción -Test de Listar Plan de producción
5	Gestionar quejas	-Test de Insertar quejas. - Test de Modificar quejas. -Test de Eliminar quejas. -Test de Listar quejas.
6	Gestionar sugerencias	-Test de Insertar sugerencias. - Test de Modificar sugerencias. -Test de Eliminar sugerencias. -Test de Listar sugerencias.
7	Gestionar Usuarios	-Test de Insertar Usuarios. - Test de Modificar Usuarios. -Test de Eliminar Usuarios. -Test de Listar Usuarios.
8	Gestionar Empresas	-Test de Insertar Empresas. - Test de Modificar Empresas. -Test de Eliminar Empresas. -Test de Listar Empresas.
9	Gestionar Artículos	-Test de Insertar Artículos. - Test de Modificar Artículos. -Test de Eliminar Artículos. -Test de Listar Artículos.
10	Autenticarse	-Test de Insertar Autenticarse - Test de Modificar Autenticarse -Test de Eliminar Autenticarse -Test de Listar Autenticarse
11	Implementar Trazas	-Test listar traza.

		- Test buscar traza .
12	Pedir artículos	-Test de pedir artículos

3.2.2 Pruebas de aceptación

A continuación, se muestran las pruebas de aceptación realizadas a la aplicación web:

En la Tabla se muestran las clases de equivalencia de las pruebas de aceptación (PA) al sistema de Test Crear Plan.

3.2.3 Clases de equivalencia por cada atributo (elaboración propia):

Atributo	Valida	Representante	Invalida	Representante
Nombre	1-Cualquier nombre de un mes	enero	2-Que este vacío	NULL
mes	3- Seleccionado de la lista desplegable	febrero	4- Que este vacío	NULL
código	5-Cualquier combinación de números y letras	2544 12A45	6-Que este vacío	NULL
			7-que contenga caracteres extraños	12/*7
Nombre de artículo	8-Selccionado de la lista desplegable	Vidatox	9- Que este vacío	NULL
Cantidad de artículos	10- cualquier combinación de números	200	11- Que este vacío	NULL
			12-Que contenga letras o caracteres extraños	2ki5*/

No	Clase de equivalencia	Nombre	Mes	Código	Nombre del artículo	Cantidad de artículos	Resultado esperado
1	1,3,5,8,10	enero	enero	1455	Vidatox	478	Plan guardado correctamente
2	1,3,5,8,12	Enero	enero	1455	Vidatox	2ki5*/	Campo requerido
3	1,3,5,8,11	Enero	enero	1455	Vidatox	NULL	Campo requerido
4	1,3,5,9,10	Enero	enero	1455	NULL	200	Campo requerido
5	1,3,5,9,11	Enero	enero	1455	NULL	NULL	Campo requerido
6	1,3,5,9,12	Enero	enero	1455	NULL	2ki5*/	Campo requerido
7	1,3,6,8,10	enero	enero	NULL	Vidatox	200	Campo requerido
8	1,3,6,8,12	Enero	enero	NULL	Vidatox	2ki5*/	Campo requerido
9	1,3,7,8,10	Enero	enero	12/*7	Vidatox	200	Campo requerido
10	1,3,7,8,12	Enero	enero	12/*7	Vidatox	2ki5*/	Campo requerido
11	1,3,7,9,10	Enero	enero	12/*7	NULL	200	Campo requerido
12	1,3,7,9,12	Enero	enero	12/*7	NULL	2ki5*/	Campo requerido
12	1,4,5,8,10	Enero	NULL	1455	Vidatox	200	Campo

							requerido
14	1,4,5,8,11	Enero	NULL	1455	Vidatox	NULL	Campo requerido
15	1,4,5,8,12	Enero	NULL	1455	Vidatox	2ki5*/	Campo requerido
16	1,4,6,8,10	Enero	NULL	NULL	Vidatox	200	Campo requerido
17	1,4,6,8,11	Enero	NULL	NULL	Vidatox	NULL	Campo requerido
18	1,4,6,8,12	Enero	NULL	NULL	Vidatox	2ki5*/	Campo requerido
19	1,3,6,9,11	Enero	enero	NULL	NULL	NULL	Campo requerido
20	1,3,6,9,12	Enero	enero	NULL	NULL	2ki5*/	Campo requerido
21	1,3,5,9,11	Enero	enero	1455	NULL	NULL	Campo requerido
22	1,3,5,9,12	Enero	enero	1455	NULL	2ki5*/	<u>Campo</u> <u>requerido</u>
23	1,3,5,8,11	Enero	enero	1455	Vidatox	NULL	<u>Campo</u> <u>requerido</u>
24	1,3,5,8,12	Enero	enero	1455	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
25	2,3,5,8,10	NULL	enero	1455	Vidatox	200	<u>Campo</u> <u>requerido</u>
26	2,3,5,8,12	NULL	enero	1455	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
27	2,3,5,8,11	NULL	enero	1455	Vidatox	NULL	<u>Campo</u> <u>requerido</u>
28	2,3,5,9,10	NULL	enero	1455	NULL	200	<u>Campo</u> <u>requerido</u>
29	2,3,5,9,11	NULL	enero	1455	NULL	NULL	<u>Campo</u> <u>requerido</u>
30	2,3,5,9,12	NULL	enero	1455	NULL	2ki5*/	<u>Campo</u> <u>requerido</u>

31	2,3,6,8,10	NULL	enero	NULL	Vidatox	200	<u>Campo</u> <u>requerido</u>
32	2,3,6,8,12	NULL	enero	NULL	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
33	2,3,7,8,10	NULL	enero	12/*7	Vidatox	200	<u>Campo</u> <u>requerido</u>
34	2,3,7,8,12	NULL	enero	12/*7	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
35	2,3,7,9,10	NULL	enero	12/*7	NULL	200	<u>Campo</u> <u>requerido</u>
36	2,3,7,9,12	NULL	enero	12/*7	NULL	2ki5*/	<u>Campo</u> <u>requerido</u>
37	2,4,5,8,10	NULL	NULL	1455	Vidatox	200	<u>Campo</u> <u>requerido</u>
38	2,4,5,8,11	NULL	NULL	1455	Vidatox	NULL	<u>Campo</u> <u>requerido</u>
39	2,4,5,8,12	NULL	NULL	1455	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
40	2,4,6,8,10	NULL	NULL	NULL	Vidatox	200	<u>Campo</u> <u>requerido</u>
41	2,4,6,8,11	NULL	NULL	NULL	Vidatox	NULL	<u>Campo</u> <u>requerido</u>
42	2,4,6,8,12	NULL	NULL	NULL	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
43	2,3,6,9,11	NULL	enero	NULL	NULL	NULL	<u>Campo</u> <u>requerido</u>
44	2,3,6,9,12	NULL	enero	NULL	NULL	2ki5*/	<u>Campo</u> <u>requerido</u>
45	2,3,5,9,11	NULL	enero	1455	NULL	NULL	<u>Campo</u> <u>requerido</u>
46	2,3,5,9,12	NULL	enero	1455	NULL	2ki5*/	<u>Campo</u> <u>requerido</u>
47	2,3,5,8,11	NULL	enero	1455	Vidatox	NULL	<u>Campo</u> <u>requerido</u>

48	2,3,5,8,12	NULL	enero	1455	Vidatox	2ki5*/	<u>Campo</u> <u>requerido</u>
----	------------	------	-------	------	---------	--------	----------------------------------

3.2.4 -Tabla de Pruebas para crear empresa (elaboración propia):

Tabla de Pruebas	
No	1
Requerimiento	Estar conectado a la base de datos
Objetivo	Probar la acción de crear un plan en el sistema (Para cubrir las clases válidas 1,3,5,8,10)
Tipo de prueba	Funcional
Hardware	Sistema de cómputo con un Procesador Intel CORE i5 - Disco duro de 1TB - Memoria RAM de 8 GB
Software	Sistema Operativo Windows 10 - Base de datos MongoDB - Navegador de Internet Mozilla Firefox 61.0, Google Chrome 64.0.3 e Internet Explorer 11.15.16.
Personal	Equipo Scrum
Caso de prueba	1
Datos de entrada	Nombre: enero Mes: enero Código:1455 Artículo: Vidatox Cantidad:478
Resultado esperado	Plan creado con éxito
Resultado obtenido	Si(x) No()

En [las figuras 2,3 y 4](#) se muestran las imágenes de los resultados de la prueba anterior. (elaboración propia)

Tabla de Pruebas	
No	2
Requerimiento	Estar conectado a la base de datos
Objetivo	Probar la acción de crear un plan en el sistema (Para cubrir las clases válidas 2,4,6,8,11)
Tipo de prueba	Funcional
Hardware	Sistema de cómputo con un Procesador Intel CORE i5 - Disco duro de 1TB - Memoria RAM de 8 GB
Software	Sistema Operativo Windows 10 - Base de datos MongoDB - Navegador de Internet Mozilla Firefox 61.0, Google Chrome 64.0.3 e Internet Explorer 11.15.16.
Personal	Equipo Scrum
Caso de prueba	42
Datos de entrada	Nombre: NULL Mes: NULL Código: NULL Artículo: Vidatox Cantidad: NULL
Resultado esperado	Campos requeridos
Resultado obtenido	Si(x) No()

En la [figura 5](#) se muestran los resultados de la prueba anterior. (elaboración propia)

Tabla de Pruebas	
No	3
Requerimiento	Estar conectado a la base de datos
Objetivo	Probar la acción de crear un plan en el sistema (Para cubrir las clases válidas 2,3,5,8,10)
Tipo de prueba	Funcional
Hardware	Sistema de cómputo con un Procesador Intel CORE i5 - Disco duro de 1TB - Memoria RAM de 8 GB
Software	Sistema Operativo Windows 10 - Base de datos MongoDB - Navegador de Internet Mozilla Firefox 61.0, Google Chrome 64.0.3 e Internet Explorer 11.15.16.
Personal	Equipo Scrum
Caso de prueba	25
Datos de entrada	Nombre: NULL Mes: enero Código: 1455 Artículo: Vidatox Cantidad:200
Resultado esperado	Campos requeridos
Resultado obtenido	Si(x) No()

En la [figura 6](#) se muestran los resultados de la prueba anterior. (elaboración propia)

Tabla de Pruebas	
No	4
Requerimiento	Estar conectado a la base de datos
Objetivo	Probar la acción de crear un plan en el sistema (Para cubrir las clases válidas 2,3,5,8,10)
Tipo de prueba	Funcional
Hardware	Sistema de cómputo con un Procesador Intel CORE i5 - Disco duro de 1TB - Memoria RAM de 8 GB
Software	Sistema Operativo Windows 10 - Base de datos MongoDB - Navegador de Internet Mozilla Firefox 61.0, Google Chrome 64.0.3 e Internet Explorer 11.15.16.
Personal	Equipo Scrum
Caso de prueba	19
Datos de entrada	Nombre: enero Mes: enero Código: NULL Artículo: NULL Cantidad: NULL
Resultado esperado	Campos requeridos
Resultado obtenido	Si(x) No()

En la [figura 7](#) se muestran los resultados de la prueba anterior.

3.3 Pruebas de sistemas web

Esta prueba se realizó con el fin de medir requerimientos inherentes a las aplicaciones web y sus características, como son el rendimiento, tiempo de carga, entre otras. Para esta prueba se utilizó la herramienta Lighthouse, una herramienta de medición desarrollada por Google específicamente para medir aplicaciones web de acuerdo con métricas estandarizadas y otras determinadas por ellos a partir de probar de sus propios productos.

La figura 8 muestra las métricas utilizadas por la herramienta Lighthouse para medir la calidad de una aplicación web, estas son el rendimiento, que se trata de los tiempos que toma la aplicación antes de que se renderice algún contenido o está lista para que el usuario interactúe con ella. La accesibilidad mide mayormente la integración con los screen readers o lectores de pantallas, estos son software utilizados por las personas con discapacidades visuales o que le impiden interactuar con los dispositivos como el resto de las personas, para una correcta integración se utilizan etiquetas especiales que permiten a los softwares leer al usuario la página y de esta forma este pueda entender la disposición y el objetivo de los componentes y como interactuar con estos. Las Best Practices o buenas prácticas son soluciones a problemas comunes que están probadas son las mejores posibles hasta el momento, ya sea que aumentan el rendimiento, la usabilidad o el mantenimiento del código, esta métrica se encarga de verificar la implantación de dichas prácticas en el sitio. El SEO u optimización para motores de búsqueda (Search Engine Optimization), mide el uso de las etiquetas de keywords o palabras claves existentes en el sitio web que ayudan a los web crawlers o indexadores web que dichos motores emplean para analizar el contenido de la página y así determinar el objetivo y poder posicionarlo mejor en los resultados de las búsquedas. Por último, el Progressive Web App o aplicación web progresiva son sitios web que utilizan la tecnología de Service Workers para indexar y almacenar en el dispositivo del usuario todos los archivos necesarios para el funcionamiento del sitio completamente desde el dispositivo, lo que permite que solo se necesiten hacer las peticiones necesarias para obtener los datos e incluso que la aplicación funcione sin estar conectada a internet. Esta métrica mide el nivel de implementación de dichas técnicas.

3.4 Pruebas de Caja Blanca

En las pruebas de caja blanca se analiza el código del software asegurando que se ejecute por lo menos una vez todos los caminos independientes de cada módulo, que se ejecute cada variante lógica, todos los ciclos y todas las estructuras internas de datos para asegurar su validez. Un tipo de prueba de caja blanca es la llamada prueba de camino básico que permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para definir un conjunto básico de caminos de ejecución. Los casos de pruebas obtenidos del conjunto básico garantizan que durante la prueba se ejecuten por lo menos una vez cada sentencia del programa.

También es importante que se defina la complejidad ciclomática como una métrica que proporciona una medida cuantitativa de la complejidad lógica de un programa. Este valor se reduce al número de caminos independientes del camino básico de dicho programa y se calcula a partir de un diagrama o un grafo de flujo (Pressman, 2010).

Ejemplo de la aplicación de esta técnica en el código de la aplicación perteneciente a la función para salvar los datos

```
articuloscliente > articuloscliente-detallees > ts articuloscliente-detallees.component.ts > ArticulosclienteDetalleesComponent > guardarPlan > plan.articulos.to
}
guardarPlan() {
  let cantidad: { nombre: string, cantidad, fecha }[] = [];
  const articulospedidos = this.listArticulos.map(art => {
    console.log(art);

    const { articulo, control } = art;
    return {
      ...articulo,
      cantidad: parseInt(control.value),
    };
  });
  console.log(articulospedidos);
  articulospedidos.forEach((articulos, i) => {
    let cantPlan = 0;
    if (this.plan.articulospedidos.find(d => d.nombre === articulos.nombre)) {
      let articulosPedidos = this.any articulospedidos.filter(d => d.nombre === articulos.nombre);
      cantPlan = articulosPedidos.reverse()[0].cantidad;
    }
    let cant = articulos.cantidad + cantPlan;
    cantidad.push({ nombre: articulos.nombre, cantidad: cant, fecha: this.date.transform(Date.now(), 'shortDate') });
    console.log(cantidad);
  });
  let error = false;
  this.plan.articulos.forEach(articulo => {
    if (cantidad.find(d => d.nombre === articulo.nombre))
    if (cantidad.find(d => d.nombre === articulo.nombre).cantidad > articulo.cantidad) {
      alert('Error');
      error = true;
    }
  });
  let nuevoPedido = this.plan.articulospedidos;

  if (this.plan.articulospedidos.length > 0) {
    cantidad.forEach(pedido => {
      let entre = false;
      this.plan.articulospedidos.forEach((artPedido, i) => {
        if (pedido.nombre === artPedido.nombre && pedido.fecha === artPedido.fecha && !error) {

```

1

2

3

4

```

nuevoPedido[i].cantidad = pedido.cantidad;

entre = true;
} else if (pedido.nombre === artPedido.nombre && i === this.plan.articulospedidos.length - 1) {
nuevoPedido.push(pedido);

entre = true;
}
} else if(i === this.plan.articulospedidos.length - 1 && !error && !entre){
nuevoPedido.push(pedido);

entre = false;
}else{

}
}
});
console.log(nuevoPedido);
this.plan.articulospedidos = nuevoPedido;
this.plan.id = this.plan._id;
if (!error)
this.planesService.update(this.plan).subscribe();
} else {

this.plan.articulospedidos = cantidad;
this.plan.id = this.plan._id;
if (!error)
this.planesService.update(this.plan).subscribe();
}
this.consultarPlan();
this.cancelar();

```

5

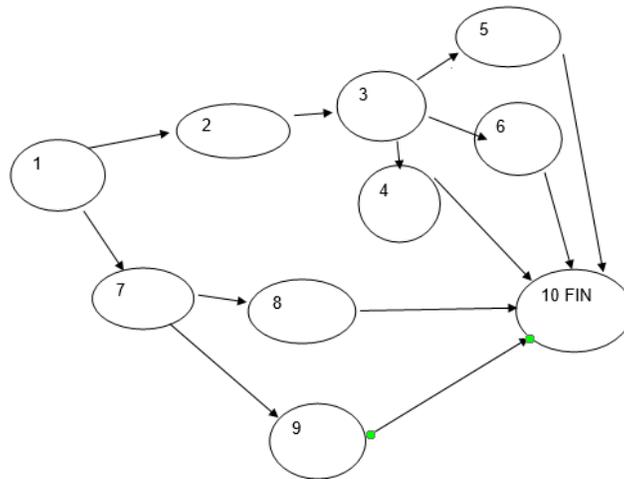
6

7

8

9

10 FIN



Grafo de flujo

Complejidad ciclomática del Grafo

V(G): Marca el límite de casos de prueba para un programa. Cuando $V(G) > 10$ la probabilidad de defectos en el módulo o programa crece mucho, entonces quizás se deba dividir el módulo. Los pasos para calcular y verificar V(G) son:

$$V(G) = A - N + 2$$

Donde:

A: número de aristas

N: número de nodos.

$$V(G) = 13 - 10 = 3$$

3.5 Interfaces de la aplicación

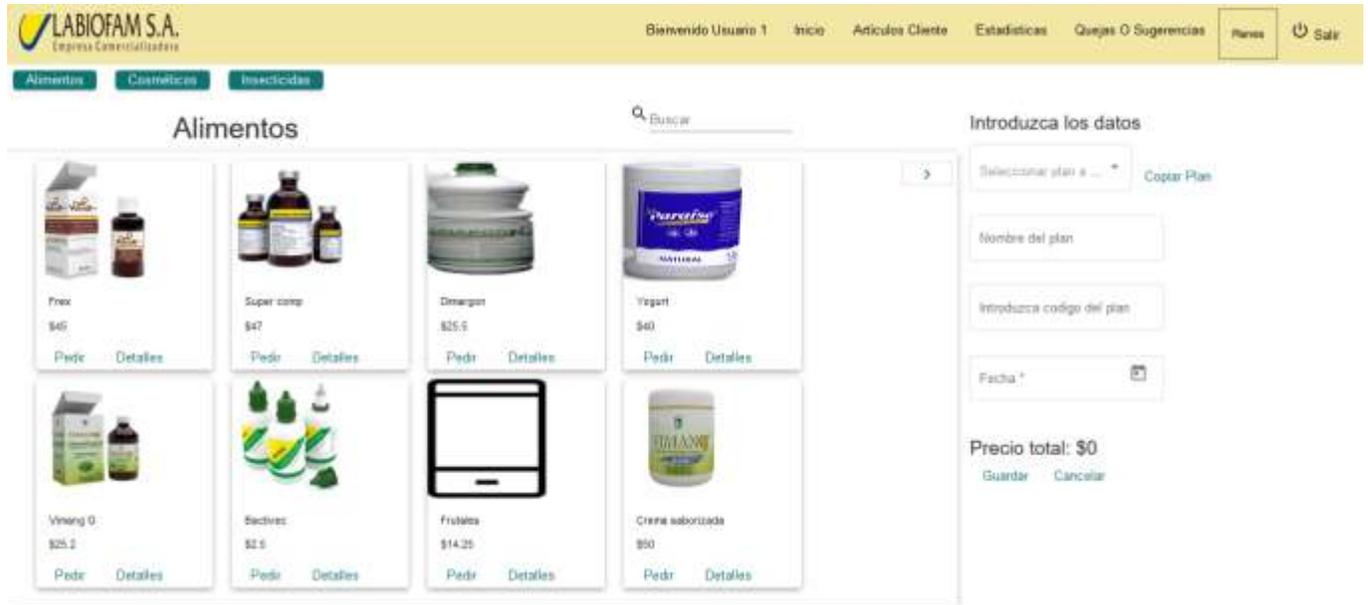


Figura 9: Interfaz de creación del plan de artículos de cada mes (Ver Demo para detalles)

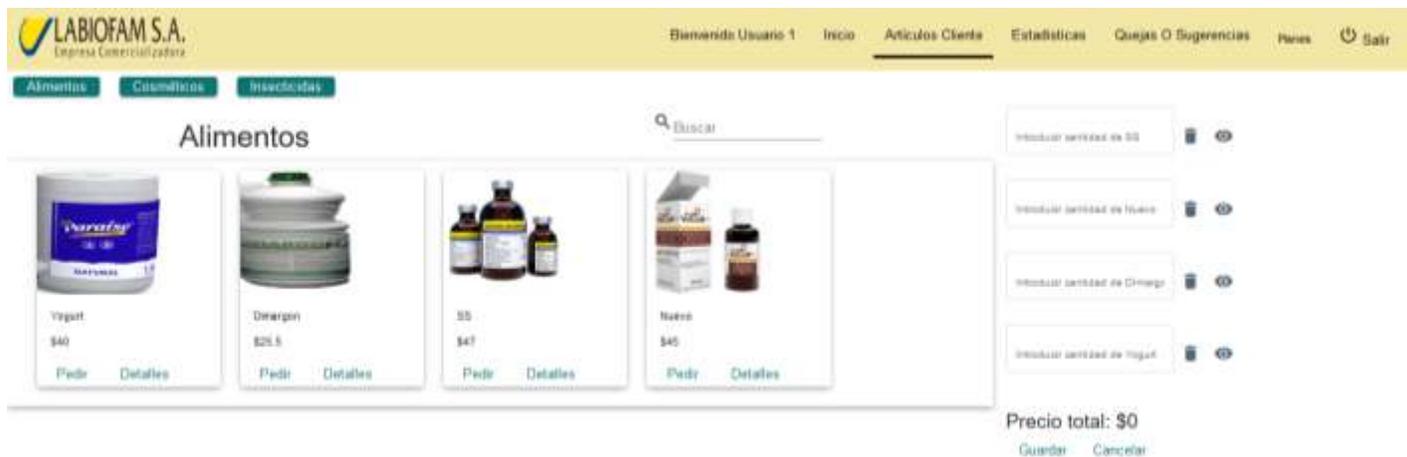


Figura 10: Interfaz para la solicitud de artículos de cada mes (Ver Demo para detalles)

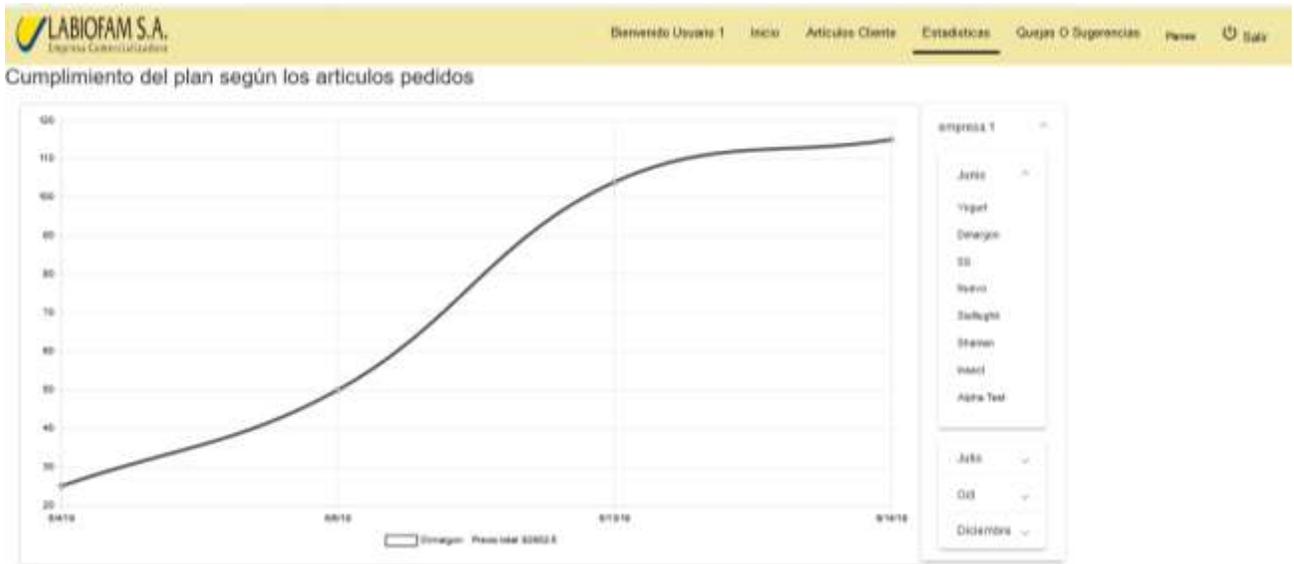


Figura 11: Interfaz que muestra grafica de solicitud de artículos (Ver Demo para detalles)

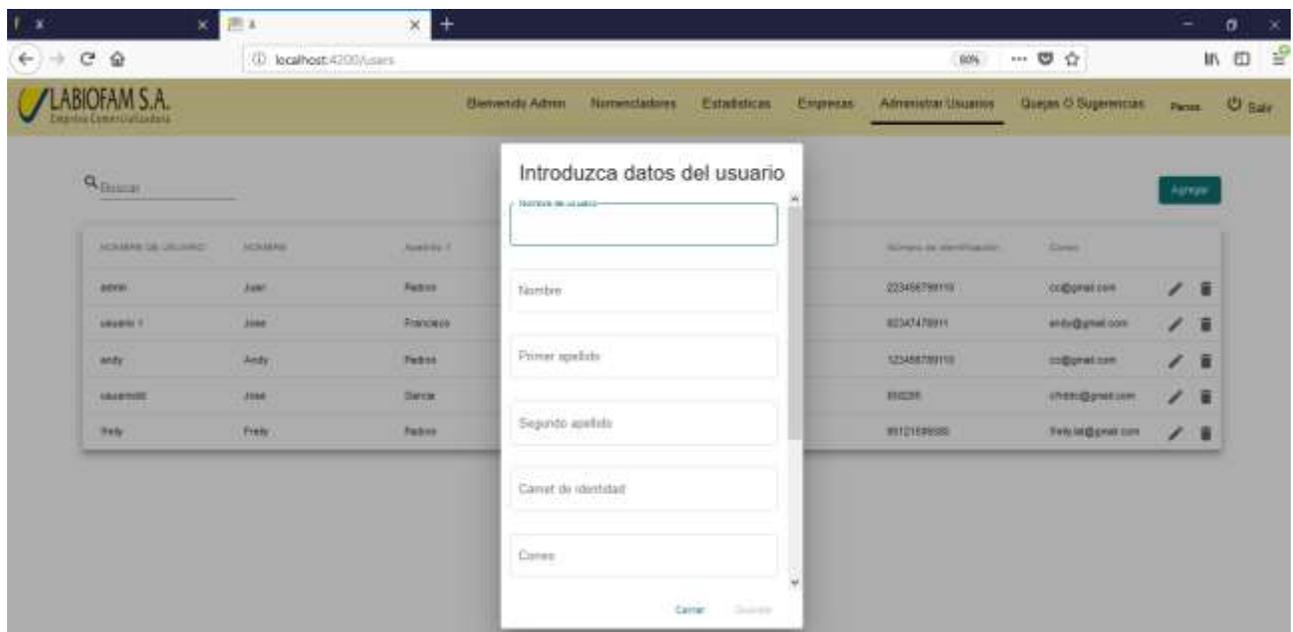


Figura 12: Interfaz para la creación de usuarios (Ver Demo para detalles)

Conclusiones Parciales

Las pruebas que se realizaron a partir de las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del *software*. Se demostró que el sistema posee un correcto funcionamiento y cumple con los requerimientos definidos en un principio para ayudar al usuario a entender la información. Se validó la aplicación en dos partes bien diferenciadas, en un primer momento realizando las pruebas tradicionales y en un segundo momento utilizando herramientas que permiten automatizar este proceso. Las pruebas también detectaron posibles mejoras en el sistema, lo que constituye al mismo tiempo, una validación del *software* desarrollado y una muestra de lo que queda por hacer.

Conclusiones Generales

- 1- Se desarrolló una aplicación web para la gestión de la solicitud de los artículos para los clientes de LABIOFAM.
- 2- Se utilizó la metodología de desarrollo de software Scrum y se demostró la factibilidad de dicha metodología para la realización del software.
- 3- Se validó la aplicación mediante la aprobación del cliente y la utilización de pruebas.
- 4- Se decidió la utilización de la metodología ágil de desarrollo de software, Scrum, así como los framework Angular entre otras herramientas para la implementación de las aplicaciones.

R e c o m e n d a c i o n e s

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

- 1- Implementar otras funcionalidades estadísticas que permitan una mejor comprensión de la interacción de los usuarios con el sistema.
- 2- Continuar el desarrollo de la aplicación móvil para que se convierta en la plataforma principal.
- 3- Continuar el desarrollo de la aplicación hasta obtener como resultado un sitio de comercio electrónico totalmente funcional.

Bibliografía

- Alcalde, A. (2013). *El baúl del programador*. Retrieved from <http://elbauldelprogramador.com/los-10-mejores-frameworks-gratis-de-aplicaciones-web/>
- André Ampuero, M., & Muñoz Castillo, V. D. (2008). ¿Metodologías tradicionales o metodologías ágiles?
- Blanch, B. (2010). *Implementación de un Sistema de Reservas para una Agencia de Viajes usando J2EE y prácticas de Desarrollo Ágil*. Ingeniería Técnica en Informática de Gestión. España: Universidad Obrera de Catalunya
- Cantón, A. C. (2017). Manual de HTML5 en español.
- Cao, L., & otros. (2016). Una explicación de la programación extrema (XP).
- Cascio, B. (2017). *Requerimientos, Técnicas de Especificación y metodologías Ágiles*.
- Castro, Llorente, C. U., & Press, K. (2010). Guía de arquitectura N-Capas orientada al dominio con .NET 4.0.
- CM Crossroads, S. (2006). Retrieved 25-3-2019, from <http://www.cmcrossroads.com/h.html>
- Colaboradores. (2018). *Introducción a Express/Node. MDN web docs*. Retrieved from https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduccion
- Consortium. (2014). Especificaciones de HTML5.
- Domínguez, M. D. (2014). *Todo Programación*. Madrid Iberprensa.
- Fernández, D. (2012). Sistema Informático en plataforma libre para el desarrollo de las técnicas de comercio electrónico en la Tecnología. OTEC.
- Foundation. (2016). *T.A.S. Apache Maven Project*. Retrieved from <https://maven.apache.org/>
- Foundation, N. j. (2016). About Node.js. Retrieved 10, abril, 2016, from <https://nodejs.org/en/about/>
- García, D. C. (2016). Aplicación informática para la gestión de reservas.
- Giraldo, Z. (2016). Herramientas de Ingeniería del Software.
- González, Y. R. (2017). *Aplicación web para mejorar la eficiencia en la planificación de la gestión de los equipos de la reserva estatal en la División de la Empresa de Atención a Equipos Matanzas*.
- Highsmith, J., & C., A. (2017). Metodologías de desarrollo de software.
- Hightower. (2016). Mean: a full JavaScript stack for web development: conference tutorial. *Journal of Computing Sciences in Colleges*, pág. 109-110.
- José H, A. (2015). *Metodologías Ágiles en el Desarrollo de Software*. España: Universidad Politécnica de Valencia
- Kniberg, H. (2017). Scrum y XP desde las trincheras Como hacemos Scrum C4Media Inc.
- Letelier, Patricio, Penadés, Carmen, M., Canós, & H, J. (2015). *Metodologías Ágiles en el Desarrollo de Software*. España: Universidad Politécnica de Valencia
- Manifiesto por el Desarrollo Ágil de Software. (s.f). <http://www.agilemanifesto.org/iso/es>.
- MVP, C. (2017). Las 5 principales ventajas de usar Angular para crear aplicaciones web. Campus MVP. Retrieved 26-3-2019, from <https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx>

- Olan, M. (2016). *Web applications with HTML5 jumpstart: tutorial presentation* (Vol. 3).
- Peña, M. A. (2007). *Arquitectura cliente-servidor*. Retrieved from <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
- Pérez, J., E. . (2016). *Introducción a JavaScript*.
- Pressman, R. (2010). *Software Engineering*.
- Risueño, P. (2015). *Comenzando con Bootstrap, framework responsive*.
- RocketTheme. (2017). *Metodologías de desarrollo de software*.
- Sánchez. (2003). *Diferencias entre metodologías ágiles y no ágiles*.
- Viviana. (2019). *MySQL vs MongoDB*. GuiaDev powered by infranetworking. Retrieved 26-3-2019, from <https://guiadev.com/mysql-vs-mongodb/>
- Zanotti, A. (2016). *El software libre su difusión en Argentina: mercado, estado, sociedad*.

A n e x o s



Figura 1 : Estructura de la arquitectura n-capas

Seleccionar plan a c... Copiar Plan

Nombre
enero

Codigo
1455

Mes
enero

Vidalox
478

Guardar Cancelar

Figura 2 : Resultado de la prueba de aceptación



Figura 3: Resultado de la prueba de aceptación

Editar

CODIGO
1455

NOMBRE
enero

MES
0

ARTICULOS

Nombre	Cantidad
Vidatox	478

[Adicionar Articulos](#)

Figura 4: Resultado de la prueba de aceptación

Seleccionar plan a c... ▼ Copiar Plan

Nombre
El nombre es requerido

Codigo
El codigo es requerido

Mes ▼
Campo requerido

Vidatox

Guardar Cancelar

Figura: 5 Resultado de la prueba de aceptación

Seleccionar plan a c... Copiar Plan

Nombre

El nombre es requerido

Código
1455

Mes
enero

Validar
200

Guardar Cancelar

Figura: 6 Resultado de la prueba de aceptación

Copiar Plan

Nombre

El código es requerido

Mes

Figura: 7 Resultado de la prueba de aceptación



Figura 8: Resultado de la prueba de software con Lighthouse