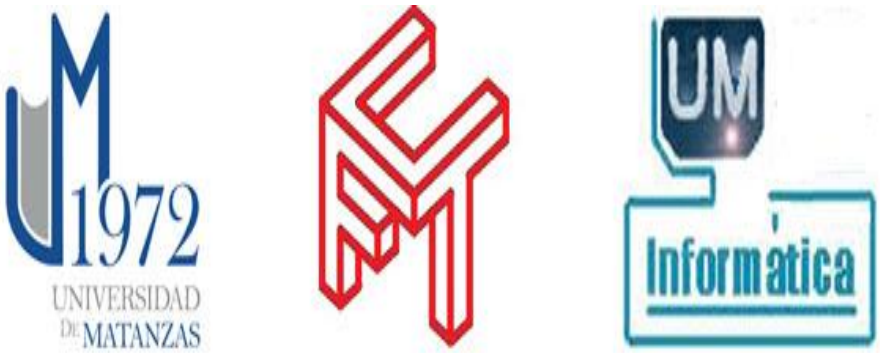
UNIVERSIDAD DE MATANZAS

Facultad de Ciencias Técnicas

Departamento de Informática



Tesis en opción al título de ingeniero en Informática

Título:

"Sistema Web para facilitar el desarrollo de funcionalidades de la Federación Estudiantil Universitaria (FEU) y la Unión de Jóvenes Comunistas (UJC) en la sede de la Universidad de Matanzas, Camilo Cienfuegos ".

Autor: Keila Mesa Pedroso

Tutor: Walfredo González Hernández

Matanzas, Cuba

julio, 2019

Pensamientos

«(…) La informática se convertirá en una poderosísima fuerza científica, económica e incluso política del país (…)».
Fidel Castro(González, 2018)

«(…) Conectarnos al conocimiento y participar en una verdadera globalización de la información que signifique compartir y no excluir, que acabe con la extendida práctica del robo de cerebros, es un imperativo estratégico para la supervivencia de nuestras identidades culturales de cara al próximo siglo». Fidel Castro(digital@juventudrebelde.cu, 2016)

Agradecimientos

Primero que todo agradecer al Señor Jesucristo porque sin sus fuerzas y ayuda no habría llegado hasta aquí, a mi familia: mi mamá, hermanas, sobrinas, cuñados, que sin su apoyo y preocupación no me fuera posible avanzar, a mi tutor Walfredo el cual no solo ha estado como trabajador y profesor de la Institución, sino que se ha portado como mi padre y aun cuando dije que no él siempre dijo sí, a mis compañeros y amigos, a Alejandro Yanes por no negarse a compartir sus conocimientos en todos estos años, a mis hermanos en Cristo, y a todo aquel que de una forma u otra me entregó su apoyo, a todos, gracias por ayudarme a cumplir esta meta en la vida.

Declaración de Autoridad

Yo, Keila Mesa Pedroso, declaro que soy la única autora de este trabajo de diploma y lo pongo a disposición de la Universidad de Matanzas, Sede “Camilo Cienfuegos”, para hacer uso del mismo con el objetivo y finalidad que se estime conveniente.

_____

Firma

Nota de aceptación

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

_____
Presidente del Tribunal

_____ _____
Miembro del Tribunal Miembro del Tribunal

Dado en la ciudad de Matanzas a los___ días del mes de _____ del año 2019.

Resumen

Este Trabajo de Diploma propone una Aplicación web para mejorar la gestión de las actividades y funciones de la FEU y la UJC en la Universidad de 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 modular, ligero y fácil de aprender y como gestor de base de datos MongoDB. Se pone de manifiesto 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 mejorar la gestión de las actividades y funciones de la FEU y la UJC en la Universidad de Matanzas.

Summary

This Diploma Work proposer a application to improve the management of the activities and functions of the FEU and the UJC at the University of Matanzas. For this, the Scrum software development methodology was used, it was used as web Programming language for the TypeScript client side and the Angular framwaork, which is modular, lightweight and easy to learn and as a MongoDB database manager. It reveals the research methodology used, the definitions associated with the object of study and of the software life cycle were properly documented and the acceptance tests that validate the software functionality are completed, obtaining a functional web application that allows improving the management of the activities and functions of the FEU and the UJC in the University of Matanzas.

Índice

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 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 automaticen procesos relacionados con distintas actividades, facilitando la organización, ejecución y control en la gestión de estas.

En Cuba el 18 de abril de 1970 fue presentada la primera computadora cubana, la CID 201. Desde hacía cinco meses atrás, jóvenes ingenieros, físicos y matemáticos de la Universidad de La Habana se habían encaminado en el empeño de diseñar y construir el equipo, un reto a gran escala, pues cualquier ayuda del exterior para adquirir las partes y piezas necesarias era truncada por el bloqueo del Gobierno norteamericano(González, 2018). No obstante, se hizo. Y detrás del hito estaba la figura de Fidel Castro, nuestro Comandante en Jefe, quien con su mirada visionaria comprendió desde bien temprano la necesidad de impulsar el desarrollo de la informática y la industria electrónica en Cuba. Tal empeño permitió, además, utilizar el modelo inicial de la primera computadora en áreas clave para la economía del país, en aquellos años, como la zafra azucarera y el control de ferrocarriles(González, 2018)

Así recordó este jueves Melchor Gil Morell, miembro del Consejo Nacional de la Unión de Informáticos de Cuba, en una de las sesiones de la Feria y Convención Internacional Informática 2018, que hasta hoy sesiona en el Palacio de Convenciones de La Habana y en el recinto ferial Pabexpo(González, 2018). A partir de los resultados que fueron alcanzándose, el líder de la Revolución Cubana estimuló el potencial de los especialistas y técnicos cubanos, y para la década de 1970 cerca del 90 % de la capacidad de cómputo del país estaba soportada en la producción nacional, comentó.(González, 2018)

Pero no solo había que producir –dijo Gil Morell–, Fidel sabía de la importancia de la universalización del conocimiento y del acceso masivo a la computación, de ahí que impulsara el surgimiento de los Joven Club de Computación y Electrónica y fue el artífice de otro gran proyecto: la Universidad de Ciencias Informáticas (UCI), que se creó en el 2002(González, 2018).

La UCI, según sus propias palabras, se trataba de un centro de nuevo tipo, de alcance nacional, de tareas concretas en el proyecto de la informatización de la sociedad cubana y con énfasis en la producción de software. Debía convertirse en el motor impulsor para el desarrollo tecnológico en Cuba, rememoró el especialista durante el encuentro(González, 2018).

Las organizaciones juveniles

La **Federación Estudiantil Universitaria**, más conocida por sus siglas FEU, es una organización cubana y que acoge en su seno a todos los estudiantes universitarios cubanos. Sigue las orientaciones del Partido Comunista de Cuba y la Unión de Jóvenes Comunistas.

Inicialmente sólo pertenecían a ella los alumnos matriculados en la Universidad de La Habana. Actualmente se han unido a los que pertenecen a las tradicionales universidades aquellos que estudian en la municipalización de la universidad cubana. Fue fundada en 1922 por Julio Antonio Mella, quien fue un líder revolucionario de la década de los años 20 del siglo XX en Cuba. Esta organización surge al calor de las reformas universitarias desarrolladas en América.("Federación Estudiantil Universitaria,").

La Unión de Jóvenes Comunistas desempeña su misión con los niños, adolescentes y jóvenes, a través del trabajo directo de los militantes y de la Organización de Pioneros José Martí (OPJM), la Federación de Estudiantes de la Enseñanza Media (FEEM), la Federación Estudiantil Universitaria (FEU) y los movimientos juveniles, la Asociación Hermanos Saíz (AHS), la Brigadas Técnicas Juveniles (BTJ) y el Movimiento Juvenil Martiano (MJM) que aceptan libre y conscientemente su conducción política, a las cuales orienta y controla sobre la base del respeto a su funcionamiento autónomo y a su independencia orgánica. ("Unión de Jóvenes Comunistas," 2019)

Misión

Contribuir a la educación comunista de las nuevas generaciones, sustentada en el patriotismo, la fidelidad al Partido Comunista de Cuba, la defensa de los más altos valores humanos, el aporte al desarrollo económico y social del país y en el espíritu profundamente antiimperialista e internacionalista que ha distinguido a la Revolución Cubana; propicia la participación creadora, consciente y entusiasta de todos los niños adolescentes y jóvenes en la construcción del Socialismo, expresada en el estudio, el trabajo, y la defensa de la Patria, simbolizados en su emblema a través del ejemplo de Mella, Camilo y el Che("Unión de Jóvenes Comunistas," 2019).

Para poder cumplir con cada misión estas organizaciones en la Universidad de Matanzas, sede Camilo Cienfuegos, realizan una serie de actividades y procesos de forma manual utilizando un tiempo grande que necesita cada miembro del secretariado de ellas para llegar a abarcar más campos de acción en medio de la Institución y poder realizar un trabajo mejor y con mayor efectividad, poder impactar a la juventud de la población universitaria y satisfacer sus necesidades. Al mismo tiempo, estos documentos que se generan ocupan gran espacio, son de difícil análisis para la toma de decisiones por lo que es necesario otra forma de representar la información.

Problema Científico

Todo lo que nos lleva al siguiente **problema científico** a resolver, ¿cómo mejorar el proceso de gestión de la información de los procesos de la UJC y la FEU en la Universidad de Matanzas? Su **objeto de estudio** sería la gestión de la información los procesos de la UJC y la FEU y su **campo de acción** la informatización de la mencionada gestión.

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

Si se desarrolla una sistema web para la gestión de la información los procesos de la UJC y la FEU en la Universidad de Matanzas entonces se incrementa la eficiencia en el que se desarrollan dichos procesos. Esta hipótesis nos deja 2 variables una **independiente** (sistema web para gestión de la información de los procesos de la UJC y la FEU en la Universidad de Matanzas.) y la otra **independiente** (gestión de la información)

El objetivo general de la investigación es desarrollar una aplicación web para facilitar la gestión de la información de los procesos de la UJC y la FEU en la Universidad de Matanzas.

Los objetivos específicos son:

1. Determinar los procesos para la gestión de la información de los procesos de la UJC y la FEU.
2. Seleccionar la metodología más adecuada para informatizar los procesos para la gestión de la información de los procesos de la UJC y la FEU.
3. Seleccionar las tecnologías más adecuadas para informatizar los procesos para la gestión de la información de los procesos de la UJC y la FEU.

4. Implementar el sistema web para informatizar los procesos para la gestión de la información de los procesos de la UJC y la FEU.
5. Validar el sistema web para informatizar los procesos para la gestión de la información de los procesos de la UJC y la FEU.

Se utilizaron diversos métodos y técnicas para el desarrollo del software, teóricos y empíricos:

Dentro de los **métodos teóricos**:

- Método Inductivo Deductivo: Este método fue utilizado para plantearse las hipótesis de trabajo respecto a los procesos de las organizaciones juveniles y la pertinencia del sistema web que se propone.
- Método Histórico – Lógico: Este método aportó el análisis de las metodologías, tecnologías y los procesos a informatizar a partir de su desarrollo histórico y las fases en que estuvo compuesto.
- Método Análisis – Síntesis: Este método permitió la descomposición de los procesos en acciones que posteriormente fueran posible analizar su computabilidad.

Los empíricos fueron:

- Observación: Se observaron los procesos referentes a la gestión de la información de las organizaciones que permitieran detallar las acciones que ejecutabas los roles fundamentales de la organización.
- Entrevistas: Permitió la apreciación de los directivos de las organizaciones acerca de sus funciones y documentos que generan, así como los roles fundamentales.
- Análisis de documentos: La aplicación de este método permitió a la autora la determinación de los procesos fundamentales que componen el accionar de estas organizaciones desde un marco jurídico.

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

Capítulo I. Marco Teórico - Referencial: Se exponen las bases teóricas que sustentan esta investigación. Se define el objeto de estudio y conceptos relacionados con el mismo. Se realiza un estudio sobre el estado de la planificación de la organización, 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 II. 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 problemática, presentando una planificación por iteraciones y un estudio de factibilidad que incluye el análisis de costos y beneficios.

Capítulo III. Aproximación a la validación práctica de la propuesta de solución del problema científico. Se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio del cliente y los propios de la metodología de software. Además, también se efectúa un estudio de los beneficios tangibles e intangibles de la realización del software.

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

También quedan recogidos la Bibliografía y los Anexos empleados que fueron necesarios para el desarrollo del trabajo y un mejor entendimiento del mismo.

Capítulo I: Marco Teórico Referencial

En este capítulo se plantea la base teórica y referencial tomada en cuenta para el desarrollo de la aplicación que se propone como resultado de esta 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

1.1 Objeto de estudio

El objeto de estudio es la gestión de la información de las actividades y procedimientos de a UJC y a FEU, por tanto, surge la necesidad de hacer una descripción más detallada de estos elementos que se pretenden transformar y perfeccionar.

1.2 Caracterización de la organización

El objetivo de la UJC Y la FEU es asegurar la unidad de los jóvenes cubanos, movilizarlos en torno a la Revolución socialista y contribuir a su educación. Estos principios rectores de la organización no sólo son para su membresía, sino también los extiende a toda la juventud, la cual representa y vela por sus intereses. Esta organización política, posee prerrogativas constitucionales en materia de política de juventud, representando al Estado cubano en los organismos internacionales relacionados con este sector de la población("Unión de Jóvenes Comunistas," 2019).

1.3 Flujo actual de trabajo.

La gestión de los procesos en la institución se realiza de forma manual por cada uno de los miembros de los consejos que ella conforma en la Universidad, ellos en lo que le corresponde por separado tienen q llenar algunos modelos para realizar uno por uno los procedimientos que correspondan en cada etapa de las actividades q realizan. Para trasladar los documentos de un sitio a otro también tienen q hacerlo ellos mismos apoyándose en algunos casos de los profesores lo que no solo retrasa sus actividades o consume más tiempo, sino que también afecta a otras áreas de la Institución.

Una de las tareas más importantes es la realización de las actas donde primero el presidente o secretario de la UJC hace sus orientaciones generales para todos las presidentes de facultad y estos también así pueden poner orientaciones para cada secretario de comité de base y de esta manera en la reunión en las aulas se toman acuerdos y se llena cada modelo de acta donde también el presidente del grupo puede emitir algunas observaciones que hizo, todo lo anteriormente expuesto se lleva al archivo y se guarda de una manera no muy segura.

Analizando el estado actual del negocio se pudo comprobar que hasta el momento se realizan los procesos de forma manual dado que los miembros del secretariado tienen que dirigirse hasta la oficina del presidente varias veces para buscar o llevar el modelo de las actas y cada orientación de él, se incurre en gastos de papel 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 realización de las actividades.

1.4 Antecedentes del trabajo.

Previo a la elaboración de este trabajo se realizó una investigación con el fin de mejorar el flujo de la información de los procesos de la militancia utilizando una herramienta informática. Además, se encontraron algunas aplicaciones o sistemas desarrollados sobre el tema:

- Sistema Informático para la Gestión de los Militantes en el Órgano Político del MININT Matanzas: Este sistema no se ajusta a las condiciones que presenta la UJC de la Universidad de Matanzas.
- SICOM, UJC: Este sistema, a pesar de estar desarrollado para la Universidad de Matanzas no cumple todas las reglas del negocio pues no permite una de las funciones fundamentales de la UJC que es dirigir el funcionamiento de la FEU. De la misma manera, este sistema no cumple con los requisitos no funcionales establecidos por la universidad para la implementación del sistema.
- Herramienta web para la gestión de los procesos de los militantes en la UJC Municipal de Unión de Reyes: Este sistema no se ajusta a las condiciones que presenta la UJC de la Universidad de Matanzas.
- SIGESUU: Este sistema no se ajusta a las condiciones que presenta la UJC de la Universidad de Matanzas.

A pesar de la existencia de aplicaciones web relacionados con este proceso las organizaciones han decidido que se implemente un sistema web que capture la información necesaria para ser almacenada en una base de datos y que sea puesta en el dominio de la escuela. Con esto lograr el acceso rápido a la misma, y registrar todas las operaciones requeridas sin dejar de hablar de la seguridad informática que brinda en este caso ya que sería administrada por informáticos en el departamento de redes. Es necesario que el sistema web que se genere se integre al dominio de la universidad por lo que se unifica los procesos de autenticación dejando en mano de los administradores de redes la creación de los usuarios. De esta manera se incremente la seguridad del sistema web.

1.5 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.

Arquitectura cliente-servidor:

La **arquitectura cliente-servidor** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras (J. C. Moreno et al., 2018).

En esta arquitectura existe una división en cuanto a capacidad del proceso entre los clientes y los servidores que deben estar conectados entre sí mediante una red, esto trae ventajas a nivel organizativo porque se centraliza la gestión de la información y se separan las responsabilidades, lo que facilita y clarifica el diseño del sistema.

N-capas:

El objetivo de la arquitectura n-capas es separar física y lógicamente los componentes mínimos de una aplicación (Tabares Morales, 2013). 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.

Fig.1 Arquitectura en N-Capas. Elaboración de la autora.

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-legislativo, se comprende como software libre a los programas licenciados en términos, que garantizan a sus usuarios el

derecho de ejecutarlos, copiarlos, distribuirlos, estudiarlos, cambiarlos y mejorarlos (Benavides López & Viscaino Naranjo, 2016; Ramírez Pérez & Bernardo Peña, 2016).

Patrones de diseño:

Los patrones de diseño son un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Estos reconocen y detallan abstracciones que van más allá del simple ámbito de clases e instancias, o componentes (Magallanes, Sánchez, Cervantes, & Wan, 2018; Patricia Villareal-Freire, Felipe Aguirre Aguirre, & Alberto Collazos Ordoñez, 2019). Para que una solución sea considerada un patrón debe tener 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.

Patrón arquitectónico:

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 propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario (Alkhraisat, 2016). 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

El MVC es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido, 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.

1.7 Metodologías de desarrollo

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 (González-Hernández & Coloma-Carrasco, 2018).

Durante el transcurso de los años han surgido diferentes metodologías. Algunas hacen más énfasis en el control del proceso, imponiendo 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 que 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 creciente del software con iteraciones muy cortas y la documentación necesaria.

Las metodologías ágiles de desarrollo de software han captado la atención de muchos 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. Estos cambios generalmente implican altos costos, demanda de tiempo y la reestructuración total del proyecto que se esté llevando; por otro lado, los métodos ágiles permiten un desarrollo iterativo y adaptable que permite la integración de nuevas funcionalidades a lo largo del progreso del proyecto; para que tanto el cliente como el desarrollador queden satisfechos porque el producto final tiene una calidad apropiada.

El objetivo de las metodologías de desarrollo ágil de software es la organización de un trabajo creativo, que suele ser bastante problemático. Se intenta dar primacía a la construcción 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 (Gren, Knaussa, & Johann Stettina, 2018; Wagenaar, Overbeek, Lucassen, Brinkkemper, & Schneider, 2018).

Ágil quiere decir, adaptable. Desde este indicio, los cambios son bienvenidos, resultan de un mejor entendimiento del problema y son una oportunidad para mejorar el software.

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.

Tras haber realizado una exposición de ambas Metodologías de realización del software, sus ventajas y desvantajas, 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 exhibir 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 evaluados.

Ahora si está leyendo este documento quizás pueda llegar a ser esta su pregunta:

¿Por qué usar SCRUM?

La metodología Scrum es una metodología ágil y flexible que gestiona el desarrollo de software y pretende maximizar el retorno de la inversión de la empresa. Sus principios son la inspección continua, adaptación, auto-gestión e innovación. Algunos de sus beneficios son("Los 10 beneficios de la metodología SCRUM,"):

1. Fomenta la motivación y el compromiso del equipo porque los profesionales se encuentran en un ámbito propicio para desarrollar sus capacidades.

2. Esto provoca una mayor productividad al eliminar la burocracia.
3. La organización horizontal promueve la autonomía y la auto-organización.
4. El desglose del trabajo favorece a una mayor flexibilidad a los cambios. Las necesidades del cliente y las evoluciones del mercado se analizan e integran a las tareas en menos tiempo.
5. Este trabajo intensificado conlleva una alta predicción de tiempos puesto que se conoce la velocidad y rendimiento del equipo.
6. Dominar estos rasgos del equipo de trabajo reduce los riesgos al conocer las funcionalidades de cada rol y la velocidad a la que avanza el proyecto.
7. La capacidad de flexibilidad y la reducción de riesgos permiten cumplir las expectativas del cliente, quien indica el valor que le aporta cada requisito del proyecto.
8. También, reduce el Time to Market: el cliente puede empezar las funcionalidades principales del proyecto antes de que este esté acabado.
9. El método de trabajo y la revisión continua produce una mayor calidad del software.

Al principio se comenzó creyendo usar la metodología Extreme Programming (XP), pero existen varias razones por lo cual no sucedió de esa manera, con la tabla presentada a continuación se da respuesta a la incógnita que podan tener:

XP	SCRUM
Se centra en la programación y creación del producto	Metodología enfocada a la administración del proyecto.
Se sigue estrictamente el orden de prioridad de las actividades definidas por el cliente	Puede modificar el orden de prioridades establecido por el Product Owner en el Sprint Backlog
Los miembros del equipo trabajan en parejas durante el proyecto	Cada miembro del Equipo Scrum trabaja de manera individual
Su estructura es más cambiante y menos organizada	Tiene una estructura más jerárquica y organizada
Las iteraciones de entrega son de 1-3 semanas	Los Sprint (iteraciones de entrega) se realizan cada 2-4 semanas
Las tareas entregadas al cliente son susceptibles a modificaciones durante el proyecto, incluso si funcionan correctamente	Al término de un Sprint, las tareas realizadas durante el Sprint Backlog y aprobadas por el cliente (Product Owner) no se vuelven a modificar

Fig. 2. Comparación entre XP y Scrum. Elaborada por Holguín Barrera (2015)

1.8Lenguajes de programación web:

JavaScript

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como Node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

JavaScript es desarrollado por ECMA, una empresa que desarrolla estándares para las tecnologías, por lo que JavaScript también se conoce como ECMAScript y se utiliza la numeración para nombrar las versiones.

ECMAScript 6 (ES6) fue publicado por ECMA en junio de 2015. También existen referencias a este como “Harmony”, “ES6”, “ES2015” o “ECMAScript 2015”. ES6 representa un avance significativo en JavaScript, y algunos de los mayores problemas de ES5 fueron atendidos en esta nueva versión(Brown, 2016). En conjunto con ES6 también surgieron los transpiladores, programas que interpretan ES6 y producen el código en ES5, como un medio para escribir ES6 para los entornos donde todavía no es soportado.

TypeScript

Solución de Microsoft para el desarrollo de aplicaciones con Javascript a gran escala, para ellos y para sus clientes. Steve Lucco y un equipo de más de 50 personas que incluía a Anders Hejlsberg, Lead Architect de C# y creador de Delphi y Turbo Pascal desarrollaron Typescript en Microsoft, un proyecto que originalmente se conoció como Strada, implementando características en el lenguaje que permitan desarrollar herramientas más avanzadas para el desarrollo de aplicaciones (Hernández, 2015).

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 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, 2010).

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).

CCS y SCSS

Los CSS nacieron algo más tarde que el HTML, al ver que todos los estilos aplicados en las etiquetas hacían poco eficiente el desarrollo. Separar los colores y los tamaños de las fuentes fue un gran paso que permite hoy en día generar una consistencia a lo largo de las páginas, centralizando y separando el contenido de cómo se visualiza. 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. Actualmente, con las novedades de HTML5, con Javascript, que da gran parte de la vida a las millones de *websites* que se visitan cada día enriqueciendo la experiencia a los usuarios y facilitando el trabajo a los desarrolladores, los CSS necesitan de alguna forma reinventarse, y es verdad que lo está haciendo actualmente, vía CSS3, cuya propagación depende de los navegadores que interpretan y dan soporte a nuevas propiedades y posibilidades.

En todo este estado de efervescencia de las tecnologías web es en la que se enmarca el preprocesador SASS (con un nombre que lo dice todo, *Syntactically Awesome Style Sheets*), para dar una nueva esperanza (Villacampa, 2015).

NodeJS

Node.js es un intérprete de JavaScript asíncrono orientado a eventos, está diseñado para construir aplicaciones de red escalables. Esto en contraste con los modelos concurrentes utilizados comúnmente hoy en día, donde se utilizan hilos del sistema. Los sistemas de red basados en hilos son relativamente ineficientes y bastante difíciles de usar. Por el contrario, los usuarios de Node no necesitan preocuparse por los errores de bloqueos mortales de procesos ya que no hay bloqueos. Casi ninguna función en Node realiza procesos de E/S, por lo que el proceso no se bloquea (Node.js).

El código JavaScript ejecutado en los navegadores y en Node es compilado utilizando el motor de JavaScript V8, que es el encargado de compilar el código a código de máquina que es mucho más rápido. El código de maquina es un código de bajo nivel que la computadora es capaz de interpretar directamente (Mead, 2018).

Frameworks:

Un framework constituye una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Express

Express es un *framework* de servidor web minimalista y flexible escrito JavaScript utilizando Node.js que proporciona un conjunto sólido de características para el soporte de las aplicaciones *web* y móviles modernas, implementando una arquitectura de microservicios y el estándar REST para la comunicación.

Está basado en el módulo http de NodeJS y componentes *Connect*. Esos componentes son llamados *middlewares* y son la piedra angular de la filosofía del *framework*, la cual es configuración sobre convención (configuration over convention), lo que significa que los desarrolladores son libres de escoger cualquier otra librería que necesiten para un proyecto. Este enfoque provee flexibilidad y una alta capacidad de personalización (Mardan, 2014).

Angular

Es un *framework* de JavaScript de código abierto, mantenido por Google, que ayuda con la gestión de lo que se conoce como aplicaciones de una sola página. Está basado en componentes, que a su vez se agrupan en módulos, permitiendo descomponer la aplicación en pequeñas partes, lo que supone mayor reusabilidad y facilidad de mantenimiento. Angular trae consigo algunas ventajas significativas a la vez que provee una estructura común para que los desarrolladores en un equipo puedan colaborar más fácilmente. Algunos de los conceptos principales son (Seshadri, 2018):

- Componentes personalizados: Permiten escribir componentes que agrupan funcionalidades y lógica visual en partes pequeñas y reusables.
- Inyección de dependencias: Permite escribir servicios de forma modular e inyectarlos en los componentes donde se necesiten

- Abarcador: Angular es un framework completamente desarrollado y provee soluciones para la comunicación con el servidor, manipulación de rutas dentro de la aplicación.

1.9 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD), es un *software* que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y además sirve de intermediario entre las bases de datos y las aplicaciones. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado (Espinosa Oliva, 2017). Los programas de aplicación operan sobre los datos almacenados en la base utilizando las facilidades que brindan los SGBD, los que, en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios.

MongoDB

MongoDB es un sistema de gestión de bases de datos diseñada para el rápido desarrollo de aplicaciones web. El modelo de datos y la estrategia de persistencia fueron creados para altas tasas de lectura – escritura y la habilidad de escalar fácilmente(Banker, Bakkum, Verch, Garret, & Hawkins, 2016). Es una de las numerosas tecnologías de bases de datos no relacionales surgidas a mediados del 2000, utilizadas en aplicaciones de Big Data y otros trabajos de procesamiento que involucran datos que no encajan en un modelo rígido relacional. La arquitectura de MongoDB está compuesta por colecciones de objetos (Rouse).

MongoDB es una base de datos sin esquemas, lo que significa que se puede almacenar objetos BJSON (Binary JSON), que son una variante de los conocidos objetos JSON pero que aceptan mayor cantidad de tipos de datos; la estructura de estos puede cambiar, ya que no es requerida, como en las bases de datos SQL. Esto es una de las ventajas de usar NoSQL, ya que agiliza el desarrollo de aplicaciones y reduce la complejidad a la hora del despliegue permitiendo que se agreguen nuevos campos sin afectar los objetos ya existentes.

Una entrada en MongoDB es un documento, que es una estructura compuesta por pares <campo, valor>, los valores de los campos pueden incluir otros documentos, arreglos y/o

arreglos de documentos, además de los tipos de datos primarios comunes a todos los lenguajes de programación.

Mongoose

Mongoose es una herramienta de modelado para MongoDB y Node.js. Lo que significa, en términos prácticos, que permite definir los modelos de datos en un solo lugar, el código. Esto evita la definición de esquemas en la base de datos. La estructura de los datos se puede definir en formato *JSON* dentro del proyecto.(Holmes, 2013)

Mongoose es utilizado principalmente cuando se quiere interactuar con datos estructurados dentro de MongoDB, también ayuda con muchas de las tareas comunes de MongoDB a la vez que remueve algunos niveles de complejidad que puede causar la comunicación nativa entre MongoDB y una aplicación de NodeJS.

1.10 Herramientas utilizadas

VS Code es un editor de código fuente sofisticado que admite muchas funcionalidades practicas al momento de trabajar con el código. Estas son algunas de ellas(E. Moreno, 2016):

Lenguajes de programación:

La edición de código no está limitada para C# y VB (lenguajes propietarios de Microsoft) si no que de nueva cuenta el Open Source está en el paquete: Java, Go, C, C++, Ruby, Python, PHP, Perl, JavaScript, Groovy, Swift, PowerShell, Rust, DockerFile, CSS, HTML, XML, JSON, Lua, F#, Batch, SQL, Objective-C…

Multiplataforma:

Fue creado y diseñado para que funcione en los tres sistemas operativos mayormente utilizados: Windows, Linux y Mac OS.

Plugins:

VS Code es una herramienta que se actualiza constantemente, tiene la posibilidad de adaptar *plugins* para trabajar con el cómputo en la nube de Microsoft Azure y desplegar proyectos directamente.

Intellisense:

Se le denomina «*Intellisense*» a la capacidad que tiene un editor de texto para predecir la instrucción que estamos por escribir, y con esto no tenemos la necesidad de escribir toda la

instrucción, ya que esta se puede autocompletar con el editor, esto nos hace más productivos y acorta la posibilidad de errores de sintaxis.

Open Source:

VS *Code* se encuentra en la red social de desarrolladores más popular del momento GitHub, por lo que es posible bajarlo a la computadora, analizar el código, hacer cambios y enviarlos mediante Git al equipo de Microsoft para que los valore y si cree conveniente, incluirlo como *core* del producto (E. Moreno, 2016).

Herramienta para el modelado.

Como herramienta para el modelado del negocio se ha utilizado Bizagi Process Modeler es un Freeware que sirve para diagramar,documentar y simular procesos usando la notación estándar BPMN (Business Process Modeling Notation) (Ruiz, 2017) Esta solución de procesos del negocio permite ejecutar o automatizar procesos o flujos de trabajo.

BPMN

Business Process Model and Notation (BPMN), en español Modelo y Notación de Procesos de Negocio, es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow)("¿QUÉ ES BPMN Y PARA QUÉ SIRVE?," 2016). Es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Business Process Diagram, BPD).

Robo3T

Robo3T mantenida y proporcionada por los desarrolladores del cliente MongoDB.Estudio 3T es una herramienta de gestión de GUI MongoDB multiplataforma disponible para Windows,MacOS y Linux. Características de Robo3T:()

Es una interfaz fácil de manejar.

Proveedor de GUI MongoDB ligero.

Menor consumo de recursos del sistema.

Herramienta visual que te ayuda a administrar la base de datos.

Autocompletado real.

Soporte para importar desde cadenas de conexión MongoDB SRV.
GUI: Interfaz Gráfica de Usuario,es un programa que actúa de interfaz de usuario,utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz

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: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1Introducción:

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 BPMN.

2.2 Modelo de Proceso

En las figuras 2.2.1 y 2.2.2 se refleja el flujo del negocio actual:

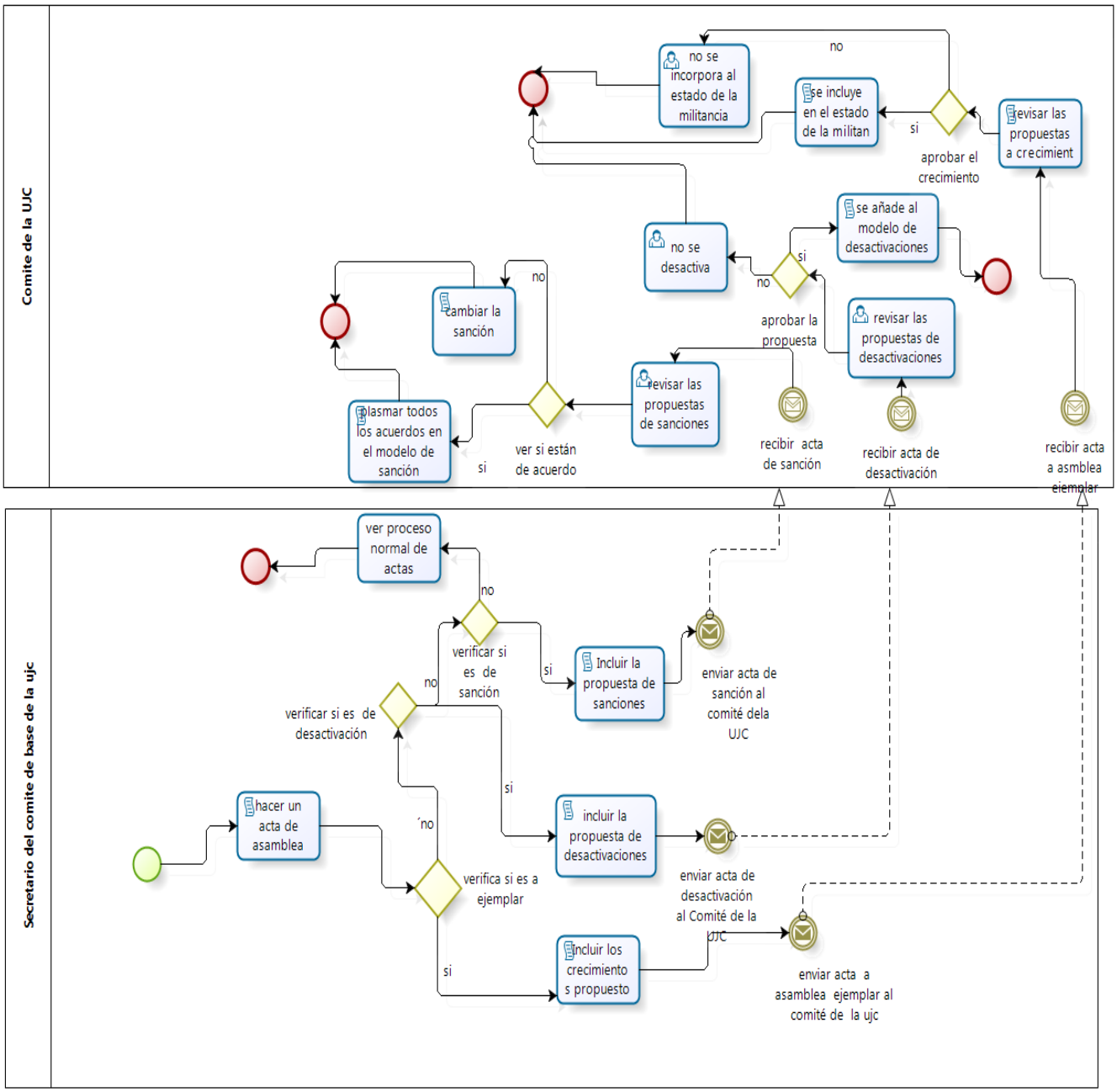

Comite de la UJC
no se incorpora al estado de la militancia
no
se incluye en el estado de la militan
si
aprobar el crecimiento
revisar las propuestas a crecimient
se añade al modelo de desactivaciones
no se desactiva
si
no
aprobar la propuesta
revisar las propuestas de desactivaciones
cambiar la sanción
no
revisar las propuestas de sanciones
plasmar todos los acuerdos en el modelo de sanción
si
ver si están de acuerdo
recibir acta de sanción
recibir acta de desactivación
recibir acta a asmblea ejemplar
Secretario del comite de base de la ujc
ver proceso normal de actas
no
verificar si es de sanción
si
Incluir la propuesta de sanciones
enviar acta de sanción al comité dela UJC
no
verificar si es de desactivación
si
hacer un acta de asamblea
´no
incluir la propuesta de desactivaciones
enviar acta de desactivación al Comité de la UJC
verifica si es a ejemplar
si
incluir los crecimiento s propuesto
enviar acta a asamblea ejemplar al comité de la ujc

Figura 2.2.1

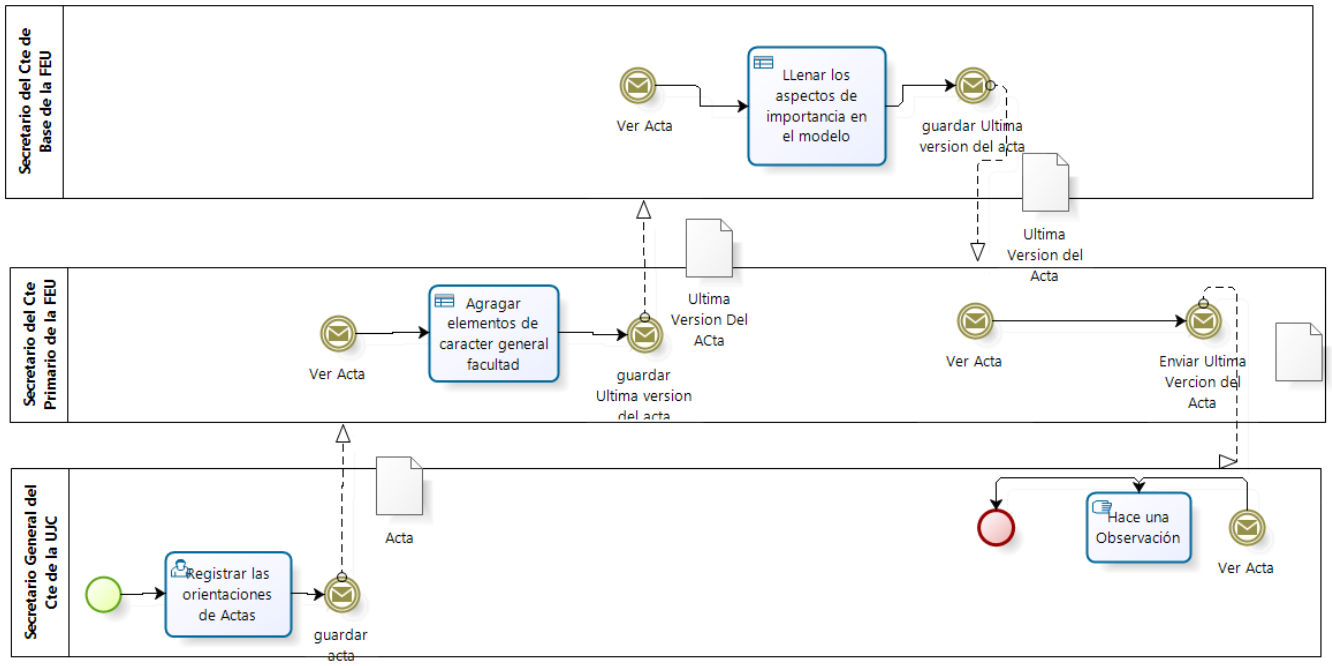


Figura 2.2.2. Modelado del Negocio. Elaboración Propia.

2.3 Descripción de la solución

Se propone el desarrollo de una aplicación web que les permita a los miembros del Secretariado de la UJC y la FEU mejorar la gestión de la información de los procesos que realizan. Además, se almacena y se gestiona las actas según la necesidad q tengan. 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 permiso al presidente, crearlo, editarlo y eliminarlo. A través del sistema de dominios en redes de la Universidad da permiso también a todos los miembros de la institución a acceder al sitio.

Joven: Trabaja directamente con el sistema, solo puede entrar y solicitar entrar o salir de la organización, puede ser cualquier miembro de la escuela que tenga cuenta en redes.

Presidente: Trabaja con el sistema, es quien asigna los demás roles (presidente de facultad, presidente del comité de base, joven).

Presidente de Facultad: trabaja con el sistema, pero solo con las actas y hace algunas observaciones en ella que ve el presidente.

Presidente del Comité de Base: trabaja con el sistema, pero solo con las actas y hace algunas observaciones en ella que ve el presidente de la facultad y el presidente.

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

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.

2.4 Etapa de planificación

2.4.1 Pila del Producto (Product Backlog)

Es el inventario de funcionalidades, mejoras, tecnologías y corrección de errores que deben incorporarse al producto a través de los sucesivos sprints. Representa todo aquello que espera el cliente, los usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo debe estar reflejado en esta pila. Lista ágil de los requisitos del cliente o requisitos del sistema:("Pila del Producto,")

Simples: expresados de forma breve, con una sentencia para cada uno, habitualmente con formato de historia de usuario similar.

Estimados: Está estimado el esfuerzo de construcción de cada requisito

Priorizados: Ordenados según la importancia para el cliente o responsable de la lista.

En la planeación de las entregas se establecen los plazos dentro de los cuales se debe desarrollar las funcionalidades especificadas. Cada entrega se puede dividir en una serie de historias, que a su vez pueden estar compuestas por tareas. Esta fragmentación permite asignar

y desarrollar múltiples labores al mismo tiempo. Las fechas de las entregas pueden variar a medida que aparecen cambios en los componentes de la pila del producto o algún otro tipo de complicaciones en el desarrollo.

Tabla 2.4.1 Planeación de las entregas

Sprint	Fecha de Inicio	Fecha de Fin
Diseño de la Base de Datos	08 enero 2019	11 enero 2019
Arquitectura inicial	13 enero 2019	20 enero 2019
Funcionalidad Actas	28 marzo 2019	8 marzo 2019
Funcionalidad Sancionados	10 marzo 2019	17 marzo 2019
Funcionalidad Desactivados	20 marzo 2019	29 marzo 2019
Funcionalidad Solicitud	1 abril 2019	8 abril 2019
Funcionalidad Controlador	9 abril 2019	14 abril 2019
Funcionalidad Usuarios	17 abril 2019	26 abril 2019
Funcionalidad Cotización	28 abril 2019	07 mayo 2019
Autenticación	22 abril 2019	03 mayo 2019
Crear aplicación	05 mayo 2019	13 mayo 2019

2.5 Descripción de las Entregas

A continuación, se detallan la composición de las entregas en historias y tareas.

2.5.1 Diseño de la Base de Datos

Historia	Tarea	Nombre
1		**Diseño de la Base de Datos**
	1	Diseño de la Base de Datos

2.5.2 Arquitectura inicial

Historia	Tarea	Nombre
1		**Crear la Arquitectura inicial**
	1	Crear la aplicación de servidor
	2	Crear la aplicación de gestión

	3	Crear la aplicación para los usuarios finales

2.5.3 Funcionalidad Actas

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear los las vistas de gestión**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3		**Crear los las vistas de actas**
	1	Crear el modelo y servicio para conectar con el servidor
	2	Agregar las actas al menú

2.5.4 Funcionalidad Sancionados

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3		**Integrar los sancionados**
	1	Crear el modelo y servicio para conectar con el servidor

	2	Crear las vistas y componentes para representar los sancionados
	3	Crear la vista de detalles
	4	Agregar sancionados al menú

2.5.5 Funcionalidad Desactivados

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3		**Integrar los desactivados**
	1	Crear el modelo y servicio para conectar con el servidor
	2	Crear las vistas y componentes para representar los desactivados
	3	Crear la vista de detalles
	4	Agregar desactivados al menú

2.5.6 Funcionalidad Solicitud

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar

	3	Crear el formulario para crear / editar
3		**Integrar las solicitudes**
	1	Crear el modelo y servicio para conectar con el servidor
	2	Crear las vistas y componentes para representar las solicitudes
	3	Crear la vista de detalles
	4	Agregar las solicitudes al menú

Para ver los demás dirigirse a los anexos en las tablas de la 2.5.7 a las 2.5.11.

2.6 Definición del Equipo

Un equipo de Scrum está compuesto por el dueño del producto, el equipo de desarrollo y un Scrum master. Los equipos son organizados por sí mismos y multifuncionales. Equipos auto organizados escogen como es mejor llevar a cabo su tarea, sin ser dirigidos por otros fuera del equipo. Equipos multifuncionales tienen todas las capacidades necesarias para completar su trabajo sin depender de otros que no pertenecen al equipo.

Los equipos de Scrum entregan partes del producto de forma iterativa e incremental, maximizando las oportunidades de retroalimentación. Las entregas crecientes del producto "hecho" aseguran que siempre exista una versión potencialmente usable del producto funcional.

Tabla 2.6.1 Definición de los roles del equipo de Scrum

Rol	Miembro
Dueño del producto	Walter ….
Equipo de desarrollo	Keila Mesa Pedroso
Scrum master	Walfredo ….

2.7 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

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

EI- EQ

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

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

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

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

EIF

Componente	Bajo	Medio	Alto	Total
EI	Eb * 3 = _	4 * 3 = _12	5 * 6 = _30	PFTe=42
EO	Ob * 4 = 16	Om * 5 = _	Oa * 7 = _	PFTo=16
EQ	10 * 3 = 30	2 * 4 = _8	Qa * 6 = _	PFTq=38
ILF	IFb * 7 = _	IFm * 10 = _	IFa * 15 = _	PFTif=15
EIF	EFb * 5 = _	EFm * 7 = _	EFa * 10 = _	PFTef=0
				PFSA =111

PFSA = PFTe + PFTo + PFTq + PFTif + PFTef

PFSA =42+16+38+15+0

PFSA =111

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	3
12. Facilidad de operación	4
13. Instalación Múltiple	2
14. Facilidad de cambio	2
FCT	50

Puntos de Función Ajustados (PFA)

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

PFA=111*[0.65+ (0.01*50)]

PFA=111*[0.65+0.50]

PFA=111*1.15

PFA=127.65

Líneas de código (LC)

LC= PFA *(Líneas * PF)

LC=164.4*100

LC= 16440

Esfuerzo hora/persona

E=PFA/ (1/8 persona/hora)

E=PFA/ (1/5) 1 persona trabaja 5h

E= 184.4/0.2

E=922horas/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

922 horas/persona/1 persona =922 horas

DM=922 horas /192 horas/mes

DM= 4,80 aproximadamente 5 meses + 1mes y 1 semana por el margen de error de la estimación por puntos de función y 1mes para la fase de prueba

DM =aproximadamente 6 meses

Costo total del proyecto

CT= sueldo de 1 persona/ cant de personas*DM

CT= (500/1) *6

CT=3000

CONCLUSIONES PARCIALES

- La solución propuesta responde a los requerimientos de la gestión de los procesos de las organizaciones juveniles que se encuentran en la Universidad de Matanzas.
- En la pila de producto se listan 11 funcionalidades agrupadas en 11 sprint.
- Al aplicar el modelo de estimación por puntos de función el costo de la aplicación web se estima en $ 3000.

CAPÍTULO 3: RESULTADOS DEL TRABAJO DESARROLLADO. ELEMENTOS DE LA VALIDACIÓN PRÁCTICA DE LA PROPUESTA DE SOLUCIÓN DEL PROBLEMA CIENTÍFICO.

En este capítulo se presentan los resultados que se obtienen al aplicar las metodologías explicadas en el capítulo 2, así como las pruebas al software realizadas, que permiten conocer el grado de calidad del producto, y de esta forma, comprobar si el sistema es capaz de realizar todas las funcionalidades detallas anteriormente.

3.1Demostración de requisitos completados

En esta fase se hace una reunión informal en la que el equipo muestra al cliente en un tiempo máximo de cuatro horas, los requisitos completados en la iteración, en forma de incremento de producto listo para ser entregado.

3.1.1Beneficios

El cliente puede visualizar objetivamente la forma en que se han desarrollado los requisitos que proporcionó antes de comenzar el ciclo y si se han cumplido sus expectativas. Además puede tomar mejores decisiones con respecto al proyecto.

El equipo puede verificar si realmente comprendió los objetivos que fueron solicitados por el cliente y ver dónde debe mejorarse la comunicación entre ambos.

El equipo se siente satisfecho al mostrar los resultados que va consiguiendo.

3.1.2Restricciones

Sólo se muestran los requisitos completados, para que el cliente no se haga falsas expectativas y pueda tomar decisiones correctas y objetivas en función de la velocidad de desarrollo y el resultado realmente completado. Un requisito no completado quedará como un requisito más a replanificar.

3.2 Pruebas al software

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. 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. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba.

La fase de pruebas es uno de los elementos que sostiene la metodología. En las pruebas de software los procesos soportados por la aplicación se cumplen completamente, es decir, los procesos fluyen desde su inicio hasta el final.

Es importante conocer que existen, según (Pressman, 2010), principios básicos que guían el buen funcionamiento de las pruebas de software y que es necesario conocerlos antes de aplicar algún método de prueba. Entre ellos se mencionan:

- Que a todas las pruebas se les debería poder hacer un seguimiento hasta los requisitos del cliente.
- Deben planificarse mucho antes de que empiecen.
- Deben comenzar por lo pequeño y progresar hacia lo grande.

La prueba de software se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecutan en condiciones previamente especificadas, registrándose los resultados obtenidos. Seguidamente se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el software. (Pressman, 2010)

3.2.1 Objetivo de las pruebas:

Las pruebas persiguen como objetivo llevar a cabo el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar al menos un error no descubierto hasta el momento.

En AEGIS-MD las pruebas son imprescindibles durante el proceso de desarrollo, antes de dar por terminadas las tareas que conforman una Historia de Usuario, deben realizarse pruebas que corroboren el buen funcionamiento del software, de manera tal que al terminar un ciclo y al hacer una entrega del producto, este debe cumplir con los requerimientos del cliente sin presentar errores. En AEGIS-MD las pruebas deben ser realizadas por el cliente y el equipo en conjunto durante la demostración de requisitos completados o en el momento en que el cliente lo decida. Entre las técnicas de evaluación empleadas para encontrar errores o fallas en el sistema se encuentran las de Verificación y Validación. Las mismas se clasifican en:

*Estáticas: se aplican sobre el sistema en reposo, tanto a requisitos como a modelos de análisis, diseño y código.

*Dinámicas: generan entradas al sistema con el objetivo de detectar fallos, al ejecutar dicho sistema sobre esas entradas. Se aplican sobre el código y se conocen como pruebas del software o testing.

Las pruebas que se realizan al software son acciones con propósitos y contextos definidos, en las que se almacenan los resultados para la realización de un posterior estudio. Las técnicas de reproducción de casos de prueba se agrupan en: (Juristo, 2005)

*Técnicas de caja blanca o estructurales, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.

*Técnicas de caja negra o funcionales, que realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. Se pueden combinar ambas técnicas para lograr un método que valide la interfaz del software y asegure que el funcionamiento interno del software sea correcto.

3.2.2 Prueba de Caja Blanca

En las pruebas de caja blanca se analiza el código del software asegurando que se ejercite 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).

```
15
16  router.post("/api/auth/signin", auth.optional, (req, res, next) => {
17    const { user: { usuario, password } } = req.body;
18
19    ldapService.checkUserCredentials(usuario, password).then(ldapResponse => {
20
21
22      if (ldapResponse.code === 200) {
23       Users.findOne({ login: usuario })
24        .then(user => {
25          if (user) {
26            return res.status(200).json({ ...(user.toAuthJSON()) });
27          } else {
28            const ldapUser = ldapService.createUserFromLDAP(ldapResponse, usuario);
29            const user = new Users(ldapUser);
30            user.save().then(savedUser => {
31              return res.status(200).json({ ...(savedUser.toAuthJSON()) });
32            });
33          }
34        });
35      } else {
36       Users.findOne({ login: usuario }).then(user => {
37        if (!user || !user.validatePassword(password)) {
38          return res.status(401).json(ERROR_USERNAME_PASSWORD_INVALID);
39        }
40        else {
41          return res.status(200).json({ ...(user.toAuthJSON()) });
42        }
43       });
44      }
45    });
```

Figura 3.2.2.1 Método para recibir la petición de autenticación al sistema

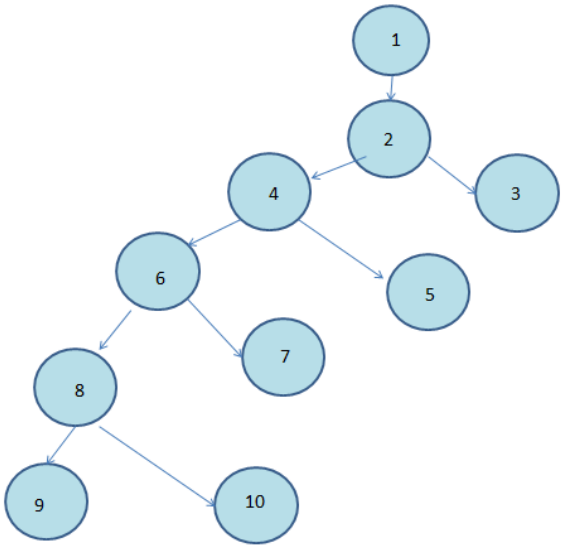


Figura 3.2.2.2 Grafo de caminos posibles

Resultados al aplicar la prueba

a = 9, n = 10, r = 1 y c = 4

V(G) = a – n + 2 = 9 – 10 + 2 = 1

V(G) = r = 1

V(G) = c + 1 = 4 + 1= 5

Caminos independientes:

- 1, 2, 4, 6, 8, 9
- 1, 2, 4, 6, 8, 10
- 1, 2, 4, 6, 7
- 1, 2, 4, 5
- 1, 2, 3
- Prueba de caja blanca al método para confirmar las credenciales de un usuario
- El método “confirmar las credenciales de un usuario” se encarga de utilizar el microservicio de comunicación con el dominio LDAP de la universidad para verificar que el usuario que está en el proceso de autenticación sea un usuario real de la universidad, si el servicio responde con un código 200, significa que en usuario si existe por lo que se pasa a verificar si existe en la base de datos del sistema, de ser así, se comprueba si se está entrando a la aplicación de gestión para ver si es necesario comprobar los roles del usuario, si no es necesario, se envía al usuario el token de autenticación para utilizar en las peticiones futuras. Si el usuario lo existe en la base de datos del sistema, se considera que es su primera vez utilizándolo por lo que se pasa a crear un registro con sus datos para su posterior uso. Esta parte del proceso evita la necesidad de un flujo de registro en el que el usuario tenga que introducir todos sus datos y permite tener una sola fuente de información con respecto a los usuarios. Si por el contrario, el servicio del dominio no responde con un código 200, se procede a una última verificación, ya que el usuario que está intentando autenticarse puede ser el usuario primario o root, que es un usuario que se crea inicialmente con el propósito de poder gestionar los usuarios, para esto, se busca el nombre de usuario en la base de datos y si es correcto se verifica también la contraseña, si estas verificaciones son correctas se autentica el usuario, sino, no quedan más alternativas válidas por lo que se devuelve al usuario un error de que las credenciales enviadas son invalidas.

```
ldapService.checkUserCredentials(usuario, password).then(ldapResponse => {


  if (ldapResponse.code === 200) {
   Users.findOne({ login: usuario })
     .then(user => {
      if (user) {
        return res.status(200).json({ ...(user.toAuthJSON()) });
      } else {
        const ldapUser = ldapService.createUserFromLDAP(ldapResponse, usuario);
        const user = new Users(ldapUser);
        user.save().then(savedUser => {
          return res.status(200).json({ ...(savedUser.toAuthJSON()) });
        });
      }
     });
  } else {
    Users.findOne({ login: usuario }).then(user => {
     if (!user || !user.validatePassword(password)) {
       return res.status(401).json(ERROR_USERNAME_PASSWORD_INVALID);
     }
     else {
       return res.status(200).json({ ...(user.toAuthJSON()) });
     }
    });
```

- Figura 3.2.2.1 Método para confirmar las credenciales de un usuario

Tabla 3.2.2.3 Casos de prueba para el método recibir la petición de autenticación al sistema

Camino	Descripción	Resultado esperado
1, 2, 4, 6, 8, 9	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña, se comprueba que el usuario no esté bloqueado, se pasa a autenticar el usuario con el dominio	Se pasa a comprobar las credenciales del usuario con el dominio especificando que no hay que comprobar los roles del usuario
1, 2, 4, 6, 8, 10	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña, se comprueba que el usuario no esté bloqueado, se pasa a	Se pasa a comprobar las credenciales del usuario con el dominio especificando que si hay que comprobar los roles del usuario

	autenticar el usuario con el dominio	
1, 2, 4, 6, 7	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña, se comprueba que el usuario no esté bloqueado	Se manda un mensaje de error al usuario de que esta boqueado en el sistema
1, 2, 4, 5	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario y una contraseña	Se manda un mensaje de error al usuario de que debe proporcionar una contraseña
1, 2, 3	Se recibe la petición de autenticación, se comprueba que se haya recibido un nombre de usuario	Se manda un mensaje de error al usuario de que debe proporcionar un nombre de usuario

3.3 Algoritmos principales

A continuación, se muestran algunos de los métodos más importantes del sistema como son la autenticación en el dominio y la creación de la información de usuario necesaria a partir de la respuesta de este.

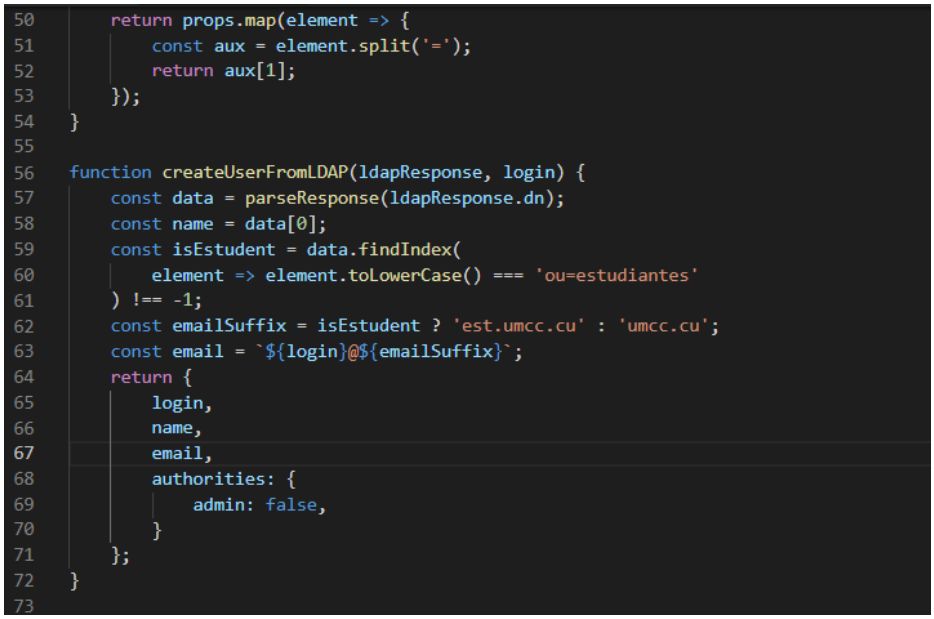
En la Figura 3.3.1 se muestra el código utilizado para hacer la petición al microservicio de comunicación con el dominio, se puede observar cómo se construye la petición, utilizando los atributos necesarios, como la dirección *IP:PORT,* que establece en qué máquina real se encuentra alojado el servicio y el puerto por el cual comunicarse con él. Se puede observar que en la parte de las cabeceras o *headers*, se utiliza una en específico que esta oculta a propósito, ya que consiste en la llave de la API (*API Key*), que es una cadena de caracteres que identifican a la aplicación que esta tratando de utilizar el servicio, en este caso el sistema SGUF. Esta clave se obtiene del departamento DGIU cuando este registra la información sobre el sistema, así como los servicios a los que tiene acceso y quien es el usuario responsable del sistema, de esta

forma se controla el acceso a los distintos servicios que el departamento provee y el uso que de estos se hace por parte de las aplicaciones de 3ros, o sea, los sistemas externos.

```
function checkUserCredentials(login, password) {
    const body = { "user_name": login, password };
    return fetch('http://10.34.8.190:3301/ldap/login', {
        method: "POST",
        body: JSON.stringify(body),
        headers: {
            'x-umcc-apikey': '[redacted]',
            'Content-Type': 'application/json',
        },
    }).then(response => {
        return response.json();
    }).catch((error) => {
        console.log(error);
        return Promise.resolve({ code: 400 });
    });
}
```

Figura 3.3.1 Algoritmo para la verificación de las credenciales de un usuario en el dominio

En la Figura 3.3.2 se puede apreciar el método implementado para construir un usuario a partir de la información provista por el dominio. El dominio está organizado de manera jerárquica, utilizando las Unidades Organizacionales (Organizational Units o OU) para agrupar y determinar niveles dentro del mismo. En el caso del dominio de la universidad, este contiene distintas OU para separar los distintos departamentos que la componen y existe una, llamada “Estudiantes”, donde se encuentran todos los estudiantes separados por facultades y grupos. En la respuesta que devuelve el servicio del dominio esta la lista de OU a las que pertenece el usuario que se autenticó, por lo que en el método se busca por la de los estudiantes. Esto es necesario ya que el dominio del correo de los usuarios es distinto al del resto de los usuarios y es necesario hacer esta verificación para poder construir la dirección de correo correctamente. La otra parte del correo se construye a partir del nombre de usuario ingresado por el propio usuario cuando intenta autenticarse.



```
    return props.map(element => {
        const aux = element.split('=');
        return aux[1];
    });
}

function createUserFromLDAP(ldapResponse, login) {
    const data = parseResponse(ldapResponse.dn);
    const name = data[0];
    const isEstudent = data.findIndex(
        element => element.toLowerCase() === 'ou=estudiantes'
    ) !== -1;
    const emailSuffix = isEstudent ? 'est.umcc.cu' : 'umcc.cu';
    const email = `${login}@${emailSuffix}`;
    return {
        login,
        name,
        email,
        authorities: {
            admin: false,
        }
    };
}

```

Figura 3.3.2 Algoritmo para crear un usuario a partir de la información que devuelve el dominio

CONCLUSIONES PARCIALES

En este capítulo se concluye con la solución propuesta, a cuyo proceso fue aplicada la metodología ágil SCRUM, obteniendo resultados satisfactorios. Cada entrega se realizó en el tiempo acordado con el cliente, propiciando la satisfacción del mismo en cada etapa del desarrollo.

Conclusiones

Durante el proceso de investigación se analizaron los fundamentos teóricos necesarios para el desarrollo de una aplicación que permita la validación de la aplicación SGUF.

Se seleccionaron las herramientas NoSql, Angular para una mejor implementación de la solución acorde al negocio propuesto.

Se seleccionó como metodología a Scrum por la simplicidad que esta propone además de cumplir con los requisitos de una metodología ágil necesaria para este proyecto.

Se implementó un sitio web que permite la gestión de información necesaria para la contribución al mejoramiento de los procesos desarrollados por la FEU y la UJC en la Universidad de Matanzas.

Se validó el sistema web para informatizar los procesos para la gestión de la información de los procesos de la UJC y la FEU.

Recomendaciones

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:

- Implementar funcionalidad entrevistas que permita una mejor comprensión de los crecimientos con el sistema.
- Aumentar la capacidad multimedia del sistema para que permita videos y una descripción mayor de cotización.

(s.f.). Recuperado el 17 de 4 de 2019, de https://www.todotech20.com/10-herramientas-principales-de-gui-de-mongodb-para-administrar-bases-de-datos-graficamente/

¿QUÉ ES BPMN Y PARA QUÉ SIRVE? (16 de Septiembre de 2016). Recuperado el 20 de Abril de 2019, de NEXTECH: http://nextech.pe/que-es-bpmn-y-para-que-sirve

Banker, K., Bakkum, P., Verch, S., Garret, D., & Hawkins, T. (2016). MongoDB in Action Second Edition.

Brown, E. (2016). Learning JavaScript.

Cantón, A. C. (2010). *Manual de HTML5 en español.*

Cliente-servidor. (s.f.). Recuperado el 7 de 4 de 2019, de https://es.wikipedia.org/wiki/Cliente-servidor

Consortium. (2014). *Especificaciones de HTML5.*

digital@juventudrebelde.cu. (9 de 8 de 2016). *Fidel y las tecnologías para todos*. Recuperado el 4 de 2 de 2019, de http://www.juventudrebelde.cu/cuba/2016-08-09/fidel-y-las-tecnologias-para-todos

Espinosa Oliva, R. (2017). *Conferencia No. 1: SGBD: CONCEPTOS BASICOS.* Matanzas: Universidad de Matanzas Camilo Cienfuegos.

Federación Estudiantil Universitaria. (s.f.). Obtenido de https://es.wikipedia.org/wiki/Federaci%C3%B3n_Estudiantil_Universitaria

González, Y. d. (22 de 3 de 2018). *Fidel,visor e impulsor de la informática en Cuba*. Recuperado el 12 de 5 de 2019, de http://www.granma.cu/cuba/2018-03-22/fidel-visor-e-impulsor-de-la-informatica-en-cuba

Hernández, U. (2015). *¿Qué es Type Script?* Recuperado el 2019

Holmes, S. (2013). Mongoose for Application Development.

Juristo, N. H. (2005). *Técnicas de Evaluación de Software.*

Los 10 beneficios de la metodología SCRUM. (s.f.). Recuperado el 3 de 5 de 2019, de https://alfatecsistemas.es/los-10-beneficios-la-metodologia-scrum/

Mardan, A. (2014). Pro Express.js.

Mead, A. (2018). Learning Node.js Development.

Moreno, E. (22 de 08 de 2016). *Visual Studio Code ¿Qué es? y ¿Qué no es?* Recuperado el 2019

Node.js. (s.f.). *About Node.js*. Recuperado el 19 de Abril de 2019, de Node.js.

Pila del Producto. (s.f.). Recuperado el 1 de 6 de 2019, de https://www.srcummanager.net/bok/index.php?

Pressman, R. (2010). *Software Engineering.*

Rouse, M. (s.f.). *MongoDB*. Recuperado el 19 de Abril de 2019, de Tech Target.

Seshadri, S. (2018). Angular Up & Running.

Unión de Jóvenes Comunistas. (s.f.). Recuperado el 2 de 4 de 2019, de https://www.ecured.cu/Uni%C3%B3n_de_J%C3%B3venes_Comunistas

Villacampa, Ó. (15 de 04 de 2015). *Qué es SASS y por qué los CSS pueden volver a divertirnos.* Recuperado el 2019, de Desarrollo Web.

Wiki. (s.f.). *Bizagi*. Recuperado el 8 de 4 de 2019, de httpa://es.m.wikipedia.org/wiki/Bizagi

Anexo1

FEU

HACEMOS CUBA

UJC

2.5.7 Funcionalidad Controlador

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE

2		**Crear**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3		**Integrar los procesos de controlador**
	1	Crear el modelo y servicio para conectar con el servidor
	2	Crear las vistas y componentes para representar los controles
	3	Crear la vista de detalles
	4	Agregar controlador al menú

2.5.8 Funcionalidad Usuarios

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear los las vistas de gestión**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar, modificar y eliminar
	3	Crear el formulario para agregar / editar
3		**Integrar los usuarios (AUF)**
	1	Agregar la funcionalidad de cambiar la foto de perfil

2.5.9 Funcionalidad Cotización

Historia	Tarea	Nombre
1		**Crear los endpoints en el servidor**
	1	Crear el modelo y los métodos GET
	2	Crear los endpoint POST / PUT
	3	Crear el endpoint DELETE
2		**Crear**
	1	Crear el modelo y el servicio para conectar con el servidor
	2	Crear la vista para listar y eliminar
	3	Crear el formulario para crear / editar
3		**Integrar los procesos de Cotización**

	1	Crear el modelo y servicio para conectar con el servidor
	2	Crear las vistas y componentes para representar las cotizaciones
	3	Crear la vista de detalles
	4	Agregar cotización al menú

2.5.10 Autenticación

Historia	Tarea	Nombre
1		**Crear la configuración en el servidor**
	1	Crear el endpoint de autenticación
	2	Crear los servicios para chequear las peticiones y los roles
	3	Crear el servicio para usuarios bloqueados
2		**Implementar la integración con LDAP**
	1	Implementar la función de autenticación con LDAP
	2	Implementar la función de verificar un usuario en LDAP
	3	Modificar el endpoint de autenticación para utilizar LDAP
3		**Implementar la autenticación (AG)**
	1	Crear el servicio de autenticación y la vista
	2	Implementar los servicios de autenticación de peticiones y manejo de errores
	3	Proteger las vistas según los roles
4		**Implementar la autenticación (AUF)**
	1	Crear el servicio de autenticación y la vista
	2	Implementar los servicios de autenticación de peticiones y manejo de errores
	3	Proteger las vistas según los roles

2.5.11 Crear la aplicación

Historia	Tarea	Nombre
1		**Crear la vista para listar las actas**
	1	Importar el modelo y servicio para conectar con el servidor
	2	Crear la vista que carga los actas según la opción escogida en el menú de navegación

2		**Crear la vista para listar los sancionados**
	1	Importar el modelo y servicio para conectar con el servidor
	2	Crear la vista que carga los sancionados según la opción escogida en el menú de navegación
	3	Crear la vista de detalles de los sancionados
3		**Crear la vista de autenticación**
	1	Importar los servicios de autenticación y manejo de errores
	2	Crear la vista de autenticación y conectarla con las funciones que lo requieren
4		**Crear la vista de Perfil**
	1	Crear la vista de perfil