



Facultad 1

Aplicación Web hacia la Exportación de los estándares BPMN

CMMN y DMN para el modelado de procesos de negocio.

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

Autora:

Dismaray Pérez Calvi

Tutor(es):

Ing. Lisdania de la Caridad Delgado Olivera

Ing. Mayleidis López Fernández

Dr. Sayda Coello González

La Habana, 2021

“Año 63 de la Revolución”

Declaración de autoría:

Dismaray Pérez Calvi la autora del trabajo de diploma con título “Aplicación Web hacia la Exportación de los estándares BPMN 2.0, CMMN y DMN para modelado de procesos de negocios flexibles”, concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los días del mes de _____ del año _____.

Dismaray Pérez Calvi

Firma del Autor

Firma del Tutor

Firma del Tutor

Tutor: Prof.Instr. Lisdania de la Caridad Delgado Olivera (@uci.cu), Ingeniera en Ciencias Informática graduada de la Universidad de las Ciencias Informática en el año 2015. Profesora del Departamento de Inteligencia Computacional donde imparte las asignaturas de Probabilidades y Estadística e Investigación de Operaciones.

Tutor: Prof.Instr. Mayleidis López Fernández, Ingeniera Industrial especializada en Organizaciones de Empresas graduada del Instituto Superior Politécnico “José Antonio Echavarría” (CUJAE) en 2013. Metodóloga de la facultad 1, Profesora Auxiliar imparte Practica Profesional.

Tutor: Dr. Sayda Coello González, Profesor auxiliar.

Agradecimientos

Agradecer primeramente a dios sin el cual nada hubiese sido posible.

A mis padres que son el más preciado tesoro y mi mayor apoyo en este camino tan difícil. Gracias por su apoyo y por cada palabra de aliento. A mi hermano por su amor y cuidado.

A mi familia a mis tíos en especial a lidia que día a día me ha mostrado el valor de la familia.

A todos mis amigos de la universidad que me han regalado tantos momentos inolvidables.

A mi amigo yasmani porque nunca falto su apoyo en los momentos difíciles.

A mi compañera de estos últimos años gracias por la paciencia y cada momento de felicidad.

A todo aquel que de alguna manera me apporto algo. En fin a todos muchas gracias por haber compartido una parte de sus vidas.

Resumen

Las organizaciones en nuestros tiempos requieren ser competitivas, flexibles y adaptables a todos los cambios. Los analistas de negocio y el personal (TI) Tecnología información enfrentan dificultades al momento de gestionar el cambio en la automatización y operación de los procesos empresariales. La integración de la fase análisis y diseño con la implementación aun representan un gran reto. Para ellos es necesario la existencia de una plataforma tecnológica que permita la creación y exportación de los procesos de negocio. Tomando la situación anterior como punto de partida, el objetivo de la siguiente investigación fue: El desarrollo de una aplicación web hacia la exportación de los estándares (BPMN) Modelo y notación de procesos de negocio, (CMMN) Modelo y notación de manejo de casos y (DMN) Modelo y notación de decisiones para el modelado de negocio flexible. Este servicio ayudará a reducir costos y controlar su inversión, además de permitir a la organización conocerse identificando su capacidad de adaptación, la realidad de la organización, su estructura, recursos y estrategias, ayudará a enfrentar un entorno sumamente variable como el actual. El sistema fue desarrollado utilizando como metodología de desarrollo de software AUP-UCI, como lenguaje de modelado UML y Visual Paradigm como herramienta CASE, lo que permite el entendimiento entre el cliente y el equipo de desarrollo. Para la construcción de la solución se utilizó el sistema de Administración e Contenidos Drupal. La solución desarrollada contribuye a agilizar el proceso de la información.

Palabras clave: Exportación, aplicación, estándares, modelado, procesos

Índice

INTRODUCCION.....	10
Capítulo 1: Fundamentos de la información, metodologías y tecnologías empleadas para darle solución a la problemática.....	15
1.1 Principales conceptos asociados al dominio del problema	15
1.1.1 Modelación de negocio:.....	15
1.1.2 Procesos de negocio:	15
1.1.3 Estándar:.....	16
1.1.4 Diagramas de procesos de negocio:	16
1.1.5 Objetivos de flujo:	17
1.1.6 Eventos:	17
1.1.7 Eventos de inicio:.....	17
1.1.8 Eventos intermedios:.....	18
1.1.9 Eventos fin:	18
1.1.10 Actividad:.....	19
1.1.11 Compuertas(Gateway):.....	19
1.1.12 Objetivos de conexión:	19
1.1.13 Canales:	20
1.1.14 Artefactos:.....	20
1.2 CMMN.....	21
1.2.1 Eventos	21
1.3 Metodología de desarrollo del Software.....	22
1.3.1 AUP.....	22
1.3.4 Herramientas y tecnologías y lenguaje	24
Capítulo 2: Descripción de la solución propuesta	29
2.1. Modelo conceptual	29
2.2. Especificación de los requisitos del software.....	29
2.2.1. Requisitos no funcionales.	32
2.3 Diagrama de clases del diseño.	33
2.4 arquitectura.....	35
2.5 Arquitectura basada en capas.....	35

2.6 Historia de Usuario	37
2.7 Diagrama de Secuencia.....	38
Capítulo 3: Evaluación de la solución propuesta.	41
3.1 Estándares de codificación	41
3.2 Estándares de codificación de Drupal.....	41
3.3 Diagrama de despliegue.....	43
3.4 Patrones de diseño	43
3.4.1 Patrones GRASP.....	44
3.4.2 Patrones GOF	44
3.5 Estrategia de Pruebas	45
3.5.1 Pruebas de usabilidad	45
3.5.2 Pruebas de aceptación	46
3.5.3 Pruebas de rendimiento	46
3.5.4 Pruebas de caja blanca	47
3.5.4 Métodos de caja negra. Técnicas de partición equivalente.....	47
Bibliografía.....	51

INDICE DE TABLAS

Figura 1. Process Model and Notation, BPMN 2.0, Eventos.....17

Figura 2. Process Model and Notation, BPMN 2.0, Evento de inicio.....18

Figura 3. Process Model and Notation, BPMN 2.0, Eventos intermedios..... 18

Figura 4. Process Model and Notation, BPMN 2.0, Evento fin19

Figura 5. Process Model and Notation, BPMN 2.0, Objetivos de conexión.....20

Figura 6. Process Model and Notation, BPMN 2.0, Artefactos.....21

Figura 7. *Diagrama del modelo conceptual para la integración de los estándares. Fuente: Elaboración propia.....29*

Figura 8. Diagrama de clases de diseño del requisito Modificar modelo BPMN. Fuente: Elaboración propia.....34

Figura 9. Diagrama de clases de diseño del requisito Mostrar modelo BPMN. Fuente: Elaboración propia.....35

Figura 10. Estilo arquitectónica n capas de CMS Drupal. Fuente: Elaboración propia.....36

Figura 11. Diagrama de Secuencia del requisito Modificar modelo BPMN. Fuente: Elaboración propia.....39

Figura 12. Diagrama de Secuencia del requisito Mostrar modelo BPMN. Fuente: Elaboración propia.....39

Figura 13. Diagrama de despliegue. Fuente: Elaboración propia.....43

INDICE DE FIGURAS

Tabla 1. Requisitos funcionales. Fuente: Elaboración propia.....	32
Tabla 2. Historias de usuario del requisito Modificar Modelo BPMN	37
Tabla 3. Historias de usuario del requisito Mostrar Modelo BPMN.....	38
Tabla 4. Historias de usuario del requisito Crear Modelo BPMN	38

INTRODUCCION

La globalización o el incremento de la competencia están demandando mayores exigencias tanto a las empresas como a las organizaciones en su capacidad de reacción frente a los cambios exigidos por el mercado, y forzado a las organizaciones a mejorar y a centrarse en sus procesos de negocio. Esto ha constituido un medio para subsistir y progresar y las organizaciones han tenido que adaptarse a las nuevas condiciones comerciales, así como a las presiones competitivas en las que deben asegurar la calidad de sus productos y la eficiencia de sus servicios y la evolución de sus sistemas de información.

La modelación de procesos de negocio constituye una constante de estudio ya que se ha llegado a la conclusión que el éxito de un sistema tiene como primer paso la identificación y comprensión de los procesos de negocios de una empresa. Los procesos de negocio describen como trabaja un negocio, como este cumple sus misiones, actividades o tareas.

A principios de los 90 nace en los países industrializados la idea de integrar las diferentes disciplinas de gestión cooperativas directamente con la operación de los procesos (BPM) Business Process Management se define como una "disciplina de gestión por procesos de negocio y de mejora continúa apoyada fuertemente por la TI". Por tanto, gestionar estos procesos implica gestionar todo el ciclo de vida, en el cual uno de ello es el modelado de procesos. Allí es donde entra en juego BPMN.

BPMN es la nomenclatura estándar para el modelado de procesos de negocio. Fue diseñado como una notación de tipo de diagrama de flujo robusto, fácil de usar y completamente independiente de la implementación propuesto inicialmente por la (BPMI) Business Process Management Institute, organización que actualmente trabaja de manera conjunta con la (OMG) Object Management Group en la promoción y desarrollo de estándares abiertos, completos y totalmente libres basados en XML, con el objetivo de proporcionar el soporte necesario al ciclo de vida de la gestión de procesos de negocio, desde su diseño, a través del desarrollo, ejecución, mantenimiento y optimización, creando un puente estandarizado para el vacío existente entre las fases de diseño del proceso de negocio y su implementación.

La notación BPMN 2.0 es una evolución en el uso de iconos y símbolos estándar para determinar claramente y sin confusión, los flujos y procesos de negocio diseñados en un diagrama de procesos. Por tanto, al diseñar un diagrama de procesos, se puede utilizar

símbolos universales que profesionales que tienen acceso que estas notaciones estándar internacionales entenderán fácilmente. Estas son algunas de las razones que hicieron que la notación BPMN 2.0 sea tan difundida y aceptada por los profesionales como una solución excelente para la representación de procesos.

El objetivo de BPMN 2.0 es utilizar la modelación para mejorar la eficiencia y obtener potencialmente una ventaja competitiva. EL diagrama utilizado comparte algunas similitudes con el diagrama de BPMN que fue creado inicialmente por la iniciativa de BPML. Ofrece una notación estándar que no requiere una capacitación elaborada para su comprensión. BPMN 2.0 también puede utilizarse para garantizar que los documentos XML puedan visualizarse utilizando una notación común.

OMG ha producido tres estándares BPMN, CMMN y DMN sucesivos, potentes y complementarios que juntos pueden modelar la gama de métodos de trabajo. Se pueden usar de forma independiente, pero fueron cuidadosamente diseñados para ser complementarios. Las organizaciones se benefician de estos tres estándares es por ello que constituyen la triple corona del proceso.

En resumen, se puede decir que la notación BPMN 2.0 es un lenguaje especialmente modelado para integrar todos los niveles jerárquicos y profesionales dedicados al BPM en la organización, facilitando mucho la gestión de procesos.

El CMMN fue creado por la (OMG) en 2014. Su objetivo era abordar los procesos impredecibles y complementar el estándar BPMN. CMMN es útil para trabajos no estructurados e impredecibles que requiere la intervención humana. Es utilizada para capturar métodos de trabajo que se basan en el manejo de "casos" que requieren diversas actividades que pueden realizarse en una orden impredecible en respuesta a situaciones cambiantes. CMMN expande los límites de lo que no se puede modelar con BPMN. Usando una combinación de BPMN y CMMN permite a los usuarios cubrir un campo mucho más amplio de métodos de trabajo.

DMN es una notación para la especificación precisa de decisiones y reglas Comerciales. Es fácilmente legible para todo tipo de persona involucrada en la gestión de decisiones. La notación ha sido diseñada para funcionar junto con. BPMN o CMMN proporcionando un mecanismo para modelar la toma de decisiones tanto dentro de los métodos de procesos como de los modelos de casos. El estándar DMN permite modelar y describir decisiones de forma declarativa en dos niveles, el nivel de requisitos y el nivel de lógica de decisiones (DRD).

La notación DMN es el estándar o norma que se utiliza para describir, modelar y visualizar decisiones de negocio. Es una de las tres normas complementarias introducidas por la OMG. De esta manera las decisiones pueden ser comprendidas y compartidas por diferentes plataformas compatibles y organizaciones. La Intención es que DMN proporcione una tercera perspectiva, el modelado de Requisitos de Decisión. Formando un Puente entre los Modelos de Procesos de Negocio y los Modelos de Lógica.

- Los Requisitos de Decisión estos definirán las decisiones que se tomarán en esas tareas, sus interrelaciones y sus requisitos para la Lógica de Decisión
- Los Modelos de Procesos de Negocio que son los que definirán tareas donde se requiere la toma de decisiones.
- La lógica de Decisión definirá las decisiones requeridas con suficiente detalle para permitir la validación y/o automatización.

Los BPMN y los DMN son complementarios, durante los trabajos de mejorar de procesos donde se realiza el análisis y diseño de los procesos de negocio utilizando diagramas de flujo siempre se llega a niveles de detalle donde se Vuelve más práctico utilizar DMN y no seguir bajando de nivel el diagrama de flujo.

El DMN es uno de los tres lenguajes importantes para el modelado de procesos de negocio. Trabajando junto con BPMN y CMMN las organizaciones tienen acceso a una completa biblioteca de idiomas para ayudar a los ciudadanos a desarrollar y a los profesionales de la tecnología de la información a colaborar en los flujos de trabajo y a las automatizaciones.

En situación descrita anteriormente desde la aparición de BPMN, y mucho más desde la absorción de BPMI por parte de la OMG la notación BPMN ha tenido un éxito, pero debido algunas carencias se crean los estándares CMMN y DMN en los cuales se puede ver la compatibilidad de los estándares, el beneficio que trae consigo el uso combinado de estos tres estándares a las organizaciones es necesario crear una aplicación que ayude a exportar los tres estándares y con ello facilitares el trabajo a los usuarios y con esta herramienta puedan tener una mejor utilización de los estándares (manual Bpmn,2018).

A partir de la situación problemática anteriormente, se identifica **el Problema a resolver**:
¿Cómo contribuir a la creación y exportación de los estándares BPMN 2.0, CMMN y DMN para el modelado de procesos de negocio flexibles?

Se identifica como **Objeto de estudio**: El modelado de proceso de negocio Campo de acción: La exportación de los estándares BPMN 2.0, CMMN y DMN en el modelado de procesos de negocios flexible.

Se propone como **Objetivo general**: Desarrollar una aplicación que permita la exportación de los estándares: BPMN 2.0, CMMN y DMN para el modelado de procesos de negocios flexible.

Para darle cumplimiento al objetivo planteado se definieron los siguientes **Objetivos específicos**:

1. Elaborar el marco teórico de la investigación relacionado con las principales tendencias sobre el modelado de procesos de negocio y la exportación de los estándares: BPMN 2.0, CMMN y DMN.
2. Diseñar la aplicación que permita la exportación de los estándares: BPMN 2.0, CMMN y DMN para el modelado de procesos de negocios flexible.
3. Desarrollar la aplicación que permita la exportación de los estándares: BPMN 2.0, CMMN y DMN para el modelado de procesos de negocios flexible.
4. Validar la aplicación propuesta.

Entre los métodos de la **investigación científica** utilizados se encuentran:

Histórico-lógico: Se utiliza para indagar sobre el estudio de los estándares y dar seguimiento a la evolución y tendencias actuales de los modelos de negocio, con el propósito de diseñar una solución acorde a las exigencias actuales.

Analítico- sintético: utilizado para, a partir de la situación problemática determinar una variante de solución mediante la búsqueda y análisis de los elementos más importantes asociados a los modelos de negocio empleando los estándares BPMN, CMMN y DMN, según la bibliografía relacionada con el tema, realizando una síntesis de los elementos más relevantes de la misma y arribar a las conclusiones de la investigación.

Modelado: Permite reflejar la estructura, relaciones y características de la solución a través de diagramas facilitando el diseño y la comprensión de los elementos necesarios para la implementación de la aplicación.

Inductivo-deductivo: Se utilizó para el razonamiento de la información consultada y llegar así a la obtención de un grupo de conocimientos particulares y generales.

El presente documento se compone de tres capítulos, los cuales están formados de la

Capítulo 1: Fundamentación Teórica de la Investigación. Este capítulo trata sobre los principales conceptos asociados a la investigación. Se realizó un estudio acerca de las principales herramientas o sistemas existentes para la gestión de cuadros. Se seleccionaron las metodologías empleadas para el desarrollo del sistema, los lenguajes, herramientas de desarrollo y patrones a utilizar. Se describieron las pruebas empleada para llevar a cabo la validación verificación del sistema.

Capítulo 2: Características de la solución propuesta. Se describió la propuesta de solución a la problemática planteada. Se recolectaron los requisitos funcionales y no funcionales del sistema. Se realizó el modelo de los artefactos correspondientes a las fases de planificación, diseño y desarrollo de la solución. Se describieron los patrones de diseño y los patrones arquitectónicos, así como los estándares que se utilizaron.

Capítulo 3: Resultados y Validación. Se realizó la estrategia de pruebas a seguir para evaluar el cumplimiento de los objetivos planteados. Se realizaron pruebas al sistema, prueba de aceptación y se valida el sistema desarrollado a partir de la validación de la solución.

Capítulo 1: Fundamentos de la información, metodologías y tecnologías empleadas para darle solución a la problemática.

En el presente capítulo son abordados los principales conceptos que intervienen en el proceso del modelo de negocios flexibles.

1.1 Principales conceptos asociados al dominio del problema

A continuación, se muestran algunos conceptos importantes relacionados con el problema planteado, con el objetivo de lograr una mayor comprensión de los términos asociados a la investigación: metodologías, lenguaje, herramientas utilizadas para dar solución al problema de investigación.

1.1.1 Modelación de negocio:

Desde el punto de vista empresarial los procesos de negocio son definidos como el conjunto de dos o más procedimientos o actividades enlazadas que realizan de forma colectiva un objetivo del negocio o meta política, normalmente dentro del contexto de una estructura organizacional en donde se definen roles funcionales y relaciones (Stephen A. White, 2009).

Es la captura de una secuencia de actividades de negocio, y la información de soporte. Los procesos de negocio describen la manera como una empresa alcanza sus objetivos. Entre los principales objetivos del modelado de procesos de negocio se destaca: la mejora del entendimiento de una situación, así como su comunicación y que puedan ser vistos como una herramienta para alcanzar las metas de un proyecto de desarrollo.

Existen diferentes niveles del proceso de modelado:

- Mapas de procesos. Son diagramas de flujo simple de las actividades.
- Descripción de proceso. Conforman una extensión del anterior, y manejan información adicional pero no suficiente para definir completamente el funcionamiento actual.
- Modelos de proceso. Son diagramas de flujo extendido con suficiente información para que el proceso pueda ser analizado, simulado, y/o ejecutado.

1.1.2 Procesos de negocio:

Un proceso de negocios es una secuencia estructurada o semi-estructurada de tareas que ejecutadas en serie o en paralelo por dos o más individuos permiten lograr un objetivo común. Los procesos de negocio son representados de forma gráfica mediante un diagrama o modelo, el cual refleja las acciones indispensables para producir los resultados esenciales

de la organización, siendo activado en función del cliente, independientemente de cómo, cuándo, por quién o el medio por el cual se produzcan dichos resultados, mostrando lo que un sistema hace o debe hacer de manera independiente a la implementación.

- Un proceso de negocios consiste en una “secuencia” de tareas. Una sola tarea ejecutada por una persona no es un proceso de negocios. Una simple revisión de la secuencia puede permitir detectar tareas susceptibles de eliminar o automatizar generando mejoras importantes.
- Un proceso de negocios es “estructurado o semi-estructurado”. Esto significa que existe alguna lógica o reglas que regulan la ejecución de la secuencia de tareas. Las tareas son realizadas de una manera específica –ad-hoc. Cuando existe una lógica claramente definida, la automatización puede usarse para eliminar errores y tomar decisiones oportunas.

1.1.3 Estándar:

Según la International Organization for Standardization (ISO), compuesta por las diferentes organizaciones nacionales de estandarización, la estandarización es la actividad que tiene por objeto establecer, ante problemas reales o potenciales, disposiciones destinadas a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto dado, que puede ser tecnológico, político o económico. Por lo tanto, podríamos definir un estándar como una normativa según la cual se establecen unas pautas particulares destinadas a realizar una función o acción particular.

1.1.4 Diagramas de procesos de negocio:

Dentro de un BPD existe un conjunto de elementos gráficos que permiten representar un proceso, tanto para usuarios del negocio como para desarrolladores.

Los elementos de BPMN que componen un BPD se pueden clasificar en cuatro categorías básicas

- Objetivos de flujo
- Objetivos de conexión
- Canales
- Artefactos

1.1.5 Objetivos de flujo:

Los objetivos de flujo que pertenecen a BPMN son rectángulos con las puntas redondas y consisten solamente en tres elementos especiales: eventos, actividades y compuertas

1.1.6 Eventos:

Un evento es representado por un círculo y es algo que sucede durante el proceso de negocio y que afecta el flujo de procesos. Suelen tener una causa o un resultado, de acuerdo en el momento en el que afecta el flujo, se dividen en tres tipos: inicio, intermedio y fin.



figura 1. Fuente: Business, Process Model and Notation, BPMN 2.0

1.1.7 Eventos de inicio:

Los eventos de inicio una nueva instancia de procesos cada vez que el proceso ocurre. Un evento de inicio también puede iniciar un subproceso

- None: No tiene establecida una condición o requisito para dar inicio al proceso.
- Message: Un proceso o aplicativo envía un mensaje para dar inicio al proceso.
- Timer: Se puede fijar una hora-fecha específica en la que activara el inicio del proceso.

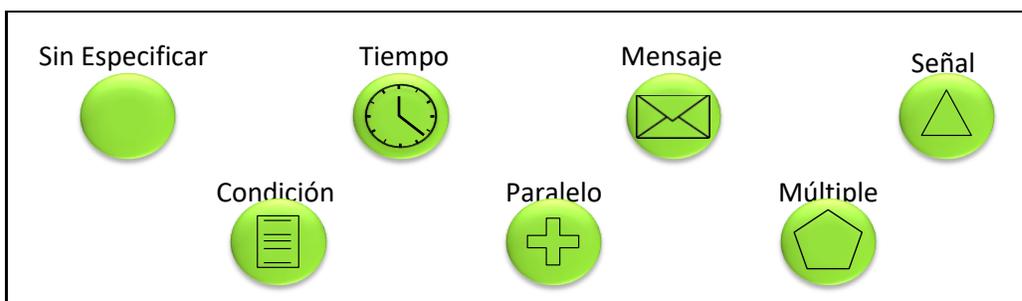


Figura 2. Fuente: Business, Process Model and Notation, BPMN 2.0.

1.1.8 Eventos intermedios:

Forman parte directa del flujo del proceso en la secuencia normal del mismo. Pueden o no anteceder a una actividad o subprocesso.

- Message: Es usado tanto para enviar o recibir un mensaje de otros procesos o aplicativos, y debe tener el mismo nombre en el mensaje
- Timer: Es un mecanismo de retraso dentro del proceso. Este tiempo puede ser definido en una expresión fecha o unidad de tiempo.
- Link: Permite conectar dos secciones de un proceso para crear flujo largas o cruzadas y están limitados a un nivel de proceso

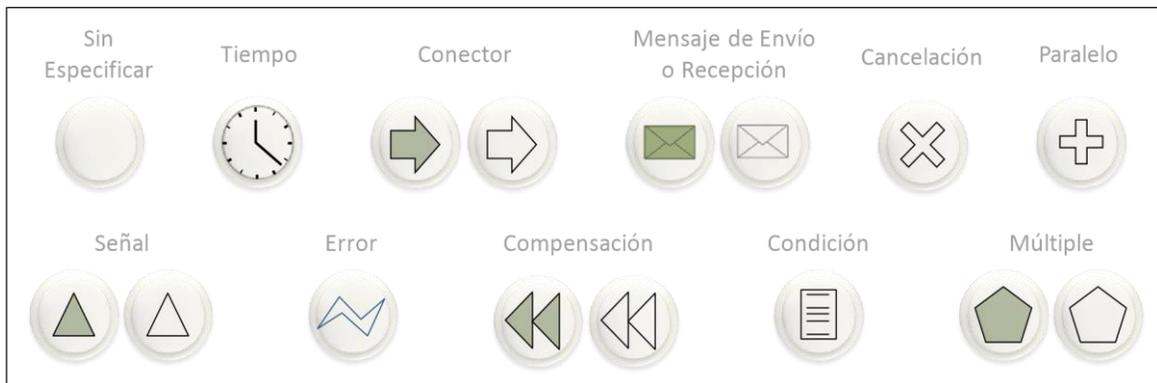


Figura 3. Fuente: Business, Process Model and Notation, BPMN 2.0

1.1.9 Eventos fin:

Todo proceso o subprocesso debe tener un evento de fin, el cual se clasifica en los siguientes tipos:

- None: No tiene establecida ninguna condición o requisito para finalizar el proceso o subprocesso
- Message: Un proceso o aplicativo envía un mensaje específico para dar fin a un proceso

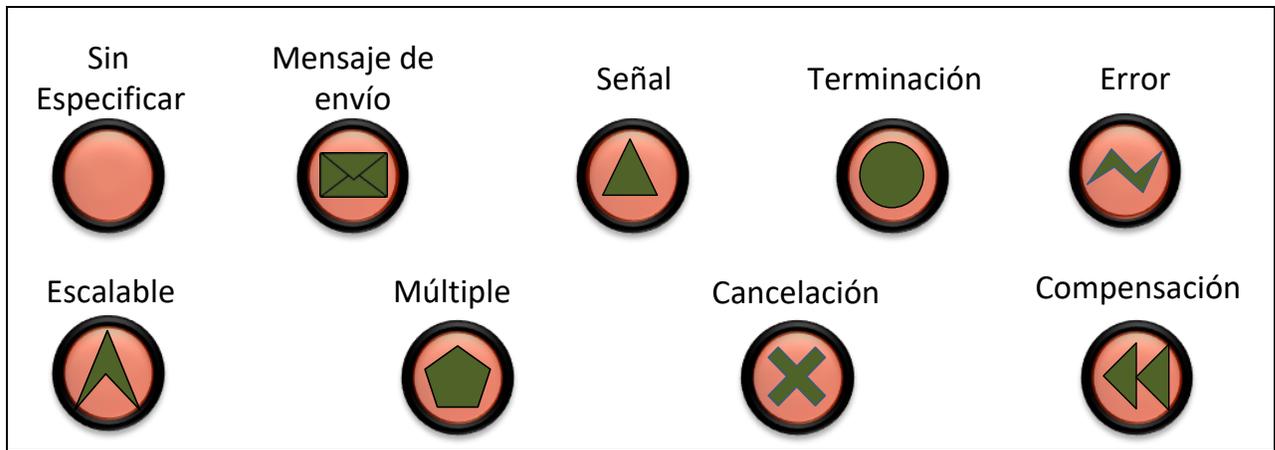


Figura 4. Fuente: Business, Process Model and Notation, BPMN 2.0

1.1.10 Actividad:

Describen el trabajo desarrollado dentro de un proceso de negocio, pueden ser atómicas o compuestas, se utilizan para modelar tareas y subprocessos y pueden ser iterativas, se representa como un rectángulo con bordes redondeados

1.1.11 Compuertas(Gateway):

Controles de secuencia de flujos dentro de un proceso, se utiliza para controlar la convergencia o divergencia del flujo de secuencia, se representa con un rombo y determina diferentes decisiones

1.1.12 Objetivos de conexión:

Los objetos de flujo se conectan entre sí utilizando los objetos de conexión. Existen tres tipos diferentes de objetos de conexión y representan el orden en que se desarrollan las actividades en el proceso.

- Flujo de secuencia: Define el orden de ejecución de las actividades.
- Flujo de secuencia condicional: Tiene una condición asignada que define si se activa la ruta.
- Flujo de mensaje: Muestra el flujo de mensajes entre dos procesos participantes.

Asociación: Se utiliza para conectar artefactos o elementos de datos a un objeto de flujo.

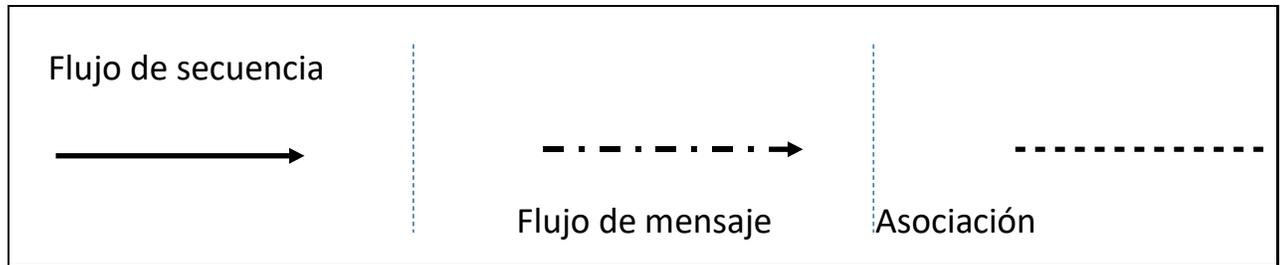


Figura 5. Fuente: Business, Process Model and Notation, BPMN 2.0

1.1.13 Canales:

Un canal o swimlane es un mecanismo visual para organizar diferentes actividades en categorías con la misma funcionalidad. Existen dos tipos diferentes de canales

- Pool: Representa los actores externos con los cuales interactúa un proceso, estos actores pueden ser un proceso o aplicativo. Contiene Objetos de Flujo, Objetos de Conexión y Artefactos.
- Lane: Representa un participante dentro de un proceso, el cual contiene un conjunto de actividades. Son usadas para organizar y categorizar los Objetos de Flujo, Objetos de Conexión y Artefactos con mayor precisión

1.1.14 Artefactos:

Los artefactos permiten a los desarrolladores incluir más información dentro del diagrama o modelo. De esta manera el diagrama o modelo es más legible. Son objetivos gráficos que provee información adicional de los elementos dentro de un proceso, sin afectar el flujo de procesos.

- Objetos de datos: Muestra los datos requeridos o producidos en una actividad como documentos del negocio, e-mails, cartas, etc. Los tipos de datos son: dato de colección de objetos de datos, almacén y mensaje.
- Grupo: Agrupa actividades que están en la misma categoría. No afecta al flujo en el diagrama.
- Anotación: Ofrecen al lector del modelo/diagrama BPMN información adicional
- Comentario. Se utiliza para incluir una nota o comentario para descubrir o documentar algún aspecto específico del diagrama.
- Símbolos propios, que los define el usuario.

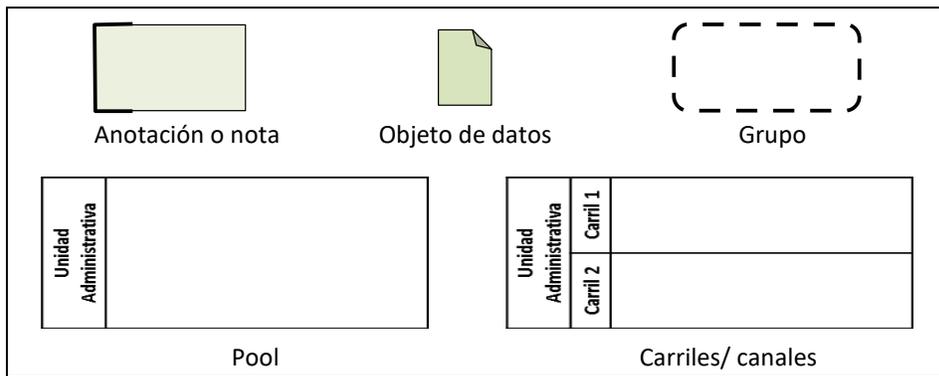


Figura 6. Fuente: Business, Process Model and Notation, BPMN 2.0

1.2 CMMN

El caso es el concepto principal, que enumera todos los roles de casos posibles y hace referencia al modelo de archivos de casos y al modelado de plan de casos. Los casos generalmente se resuelven en ad hoc, lo que requiere flexibilidad durante el tiempo de ejecución. Por lo tanto, los casos se refieren más al trabajo no estructurado y no rutinario realizado por trabajadores. Esto es en contraste con un proceso estructurado y rutinario como el modelado de BPMN.

CMMN se trata de administrar un contexto partícula en términos de casos en los que la ejecución de la actividad depende de las circunstancias del tiempo de ejecución proporcionando un cierto grado de flexibilidad en el momento de la ejecución.

El archivo de casos contiene todos los datos utilizados en el caso. Los miembros del archivo de casos son elementos del archivo de casos que pueden ser cualquier tipo de información que va desde un archivo simple normalmente codificado en XML.

El modelado de plan contiene los aspectos de comportamiento: eventos, tareas e hitos, denominados Elementos del plan.

1.2.1 Eventos

Los eventos pueden ser activados manualmente por el usuario, por el sistema después de un tiempo transcurrido o por cambios detectados en el modelo de archivo de casos.

1.3 Metodología de desarrollo del Software

Para alcanzar la calidad del producto actualmente es necesario llevar a cabo un conjunto de procedimientos, técnicas y herramientas que permiten ajustar el proyecto a las exigencias del usuario. Es por ello que la selección de la metodología a aplicar posee tanta importancia.

1.3.1 AUP

Proceso Unificado Ágil (AUP, por sus siglas en ingles) de Scott Amblar es una versión simplificada del proceso Unificado de desarrollo (RUP, por sus siglas en ingles). Se decidió trabajar con esta metodología debido a que esta describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocios usando técnica ágiles y conceptos que se mantienen validados en RUP.

1.3.2 Descripción de las fases de AUP:

De las cuatro fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, se unifican las restantes tres fases de AUP en una sola, a la que se llamará Ejecución y se agrega la fase de Cierre.

- Inicio: se lleva a cabo las actividades relacionadas con la planeación del proyecto. Se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance de proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.
- Ejecución: se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, se obtienen los requisitos, se elabora la arquitectura, el diseño, se implementa y se libera el proyecto.
- Cierre: se analizan tanto los resultados del proyecto como su ejecución y se realizaron las actividades formales de cierre del proyecto.

1.3.3 Las disciplinas definidas en la variación AUP para la UCI son:

- Modelado de negocio: destinada a comprender los procesos de negocio de una organización. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC).

Capítulo 1: Fundamentos de la Información, metodologías y tecnologías empleadas para darle solución a la propuesta.

- Requisitos: comprende la administración y gestión de los requisitos funcionales y no funcionales del producto, agrupados en cuatro escenarios condicionados por el Modelado de negocio. Existen tres formas de encapsular los requisitos [Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP)].
- Análisis y diseño: se modela el sistema y su forma (incluida su arquitectura) para que soporte los requisitos, incluyendo los requisitos no funcionales.
- Implementación: a partir de los resultados del Análisis y Diseño se construye el sistema.
- Pruebas internas: se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.
- Pruebas de liberación: pruebas diseñadas y ejecutadas por una entidad certificadora externa, de la calidad a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- Pruebas de Aceptación: es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Todas las disciplinas antes definidas se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen iteraciones y se obtengan resultados incrementales.

A partir de que el modelado de negocio propone tres variantes a utilizar (CUN, DPN o MC) y existen tres formas de encapsular los requisitos (CUS, HU, DPR), surgen cuatro escenarios para modelar el sistema en los proyectos UCI.

- **Escenario No.1** Proyectos que modelan el negocio con CUS solo pueden modelar el sistema con CUS. $CUS+MC=CUS$.
- **Escenario No.2** Proyectos que modelan el negocio con MC solo pueden modelar el sistema con CUS. $MC=CUS$.
- **Escenario No.3** Proyectos que modelan el negocio con DPN solo pueden modelar el sistema con DRP. $DPN+MC=DRP$.
- **Escenario No.4** Proyectos que no modelan negocios solo pueden modelar el sistema con HU.

Para las características del sistema desarrollado se define como escenario a utilizar el No.4. Debido a que muestra un fuerte vínculo entre cliente-desarrollador.

1.3.4 Herramientas y tecnologías y lenguaje

Se opta como CMS para la implementación de la propuesta Drupal 8 porque facilita la construcción de cualquier sitio web en un tiempo récord y permite sacarle el máximo rendimiento, ya que resulta realmente intuitiva y eficaz. Se trabaja con CSS3 en la propuesta de solución ya que es un lenguaje de diseño gráfico que permite definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Se decidió trabajar con HTML en la propuesta de solución debido a que es gratuito no se necesita ningún tipo de programa especial para trabajarlo, garantiza un código ordenado debido a la adición de las nuevas etiquetas que ayudan a nombrar partes de la estructura básica de toda la página web. Se utiliza JavaScript ya que es un lenguaje dinámico, orientado a objeto y eventos, es un lenguaje multiplataforma, se puede ejecutar en Windows, Linux o Mac, en cualquier navegador. Se trabaja con Apache ya que es el Servidor Web más utilizado y además líder con el mayor número de instalaciones a nivel mundial muy por delante de otras soluciones como el IIS (Internet Information Server) de Microsoft. Permite trabajar con varios lenguajes del lado del servidor.

Drupal 8

Drupal es un software de código abierto basado en el lenguaje de programación PHP y no depende de ninguna plataforma puede ser ejecutado en cualquier página local. Es un CMS o software desarrollado para que cualquier usuario pueda administrar y gestionar contenidos de una web con facilidades y sin conocimientos de programación.

En su última entrega ha renovado por completo su interfaz de administración y ha mejorado su sistema de caché, lo que se traduce en un importante aumento del rendimiento respecto a las anteriores versiones y en una rapidez mucho mayor

Lenguaje del lado del servidor

PHP 7

El lenguaje PHP o HyperText Pre-processor, es un lenguaje de programación de código abierto que se ejecuta desde el servidor que puede ser incorporado en HTML 5. PHP 7. PHP es un lenguaje multiplataforma que separa el diseño de la lógica a la hora de desarrollar. Es uno de los lenguajes de programación más utilizados del mundo, permite aplicar técnicas de programación orientada a objetos. No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución. (1). Es un lenguaje de programación interpretado, es decir que el

código se interpreta cuando se ejecuta el script lo que lo hace un poco más lento que otros lenguajes compilados.

Tecnologías y lenguaje del lado del cliente

HTML 5

(HyperText Markup Language) es la quinta revisión del lenguaje de marcado estándar que se emplea para la web. Es uno de los lenguajes de marcado más usados en todo el mundo y la razón es bastante obvia: gracias a HTML podemos crear la estructura de una página web. Texto, imágenes y material multimedia pueden mostrarse correctamente gracias a HTML 5. Posee compatibilidad con los navegadores más modernos y populares como Chrome, Firefox, Safari y Opera.

Bootstrap

Bootstrap es un framework de css de código abierto creado por desarrolladores de twitter que permite dar formas a sitios web a través de librerías css. Incluye cuadros, botones, menús, tipografías y otros elementos que pueden ser utilizados en cualquier sitio web. Permite la creación de sitios web responsive, es decir, el sitio web se adapta automáticamente al tamaño del dispositivo en el que se visualiza sin perder la estructura. Brinda agilidad a la hora de cargar contenido (Fontela, 2015). Bootstrap utiliza las librerías de JQuery para la validación de datos y para la visualización de ciertos elementos como son las tablas. Bootstrap puede ser personalizable de manera sencilla, no presenta grandes dificultades a la hora de adaptarlo a las necesidades de nuestros proyectos. También son personalizables los plug-ins de JQuery que Bootstrap utiliza.

CSS 3

Es un lenguaje de diseño gráfico, es muy usado para establecer El diseño virtual de los documentos web. Es una tecnología que ha tenido una evolución en el tiempo, que actualmente, como su propio nombre indica. Sus siglas corresponden a "Cascading Style Sheets", que tiene el siguiente significado:

- Cascading, que significa que los estilos que aplicamos a los elementos de una página web se propagan a los elementos que contiene, se propagan en cascada.
- Style, porque mediante CSS lo que hacemos es aplicar estilos visuales a los distintos elementos de nuestra página web.
- Sheets, que significa hojas, porque los estilos de una página web se añaden en ficheros aparte, en ficheros con la extensión .css de manera general.

Capítulo 1: Fundamentos de la Información, metodologías y tecnologías empleadas para darle solución a la propuesta.

CSS ha ido evolucionando desde hace 25 años a la versión actual, que es CSS 3, pero su función es la misma, sirve para dar estilos visuales a las páginas web. Es muy usado para establecer el diseño visual de los documentos web e interfaces de usuario escritas en HTML. En la propia definición de CSS vemos que está muy ligado desde su nacimiento a HTML. Desde que nació, el objetivo de CSS fue poner un poco de orden a la hora de aplicar los estilos a las páginas web.

JavaScript

JavaScript es un lenguaje ligero e interpretado orientado a objeto multiparadigma basado en prototipos que puede ser incrustado en documentos HTML. También se utiliza para desarrollar sitios web dinámicos, pero brinda una gran cantidad de funcionalidades adicionales como son el desarrollo de Interfaces de programación de Aplicaciones del Navegador (APIs), los marcos de trabajo y librerías de terceros.

JQuery

Es una biblioteca de JavaScript rápida y concisa que simplifica el documento HTML. Su característica principal es que permite cambiar el contenido de una página sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones Ajax. Para ello utiliza funciones `$()` o `JQuery`. Al igual que otras bibliotecas ofrece una serie de funcionalidades basadas en JavaScript con las cuales se logra más resultados en menos tiempo y espacio.

Servidores de aplicaciones

Apache 2.4.33

El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento.

Gestor de base de datos

Un Sistema Gestor de Bases de Datos es un conjunto de programas que administran y gestionan los datos que están contenidos en una base de datos. Los usuarios pueden acceder a la información usando herramientas específicas o bien mediante aplicaciones.

MYSQL

Capítulo 1: Fundamentos de la Información, metodologías y tecnologías empleadas para darle solución a la propuesta.

Es un sistema de bases de datos principalmente utilizado en las aplicaciones web en conjunto con PHP. En su versión 5. Y está considerada como base de datos de código abierto más popular del mundo.

En el desarrollo de la solución se selecciona MySQL por su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

PhpMyAdmin

Es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web. Por las facilidades que brinda esta herramienta para la gestión de Bases de Datos se decide utilizarla. Permite crear y eliminar Base de Datos, crear, eliminar y alterar tablas, borrar editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en el campo, administrar privilegios, y exportar datos en varios formatos y está disponible en 7 idiomas.

Herramientas de modelado

Para la creación de los diagramas se selecciona el Visual Paradigm for UML, herramienta de Ingeniería de Software asistida por computadora que soporta el modelado mediante UML. Proporciona existencia para sus usuarios durante todo el ciclo de vida del desarrollo del software.

Librería

Una librería o biblioteca es un Kit de herramientas software pequeño y autónomo que ofrece una funcionalidad muy específica al usuario. Normalmente se usa junto con otras librerías y herramientas para hacer una aplicación completa, ya que por lo general las bibliotecas no son ejecutables, pero sí pueden ser usadas por ejecutables que las necesiten para poder funcionar.

Bpmn js

Bpmn-js es un kit de herramientas de representación BPMN 2.0 y un modelador web está escrito en JavaScript e integra el diagrama de flujo BPMN 2.0 en los navegadores modernos sin la necesidad de soporte de servidor de fondo. Esta biblioteca puede ser un visor web o un modulador web.

Conclusiones Parciales

Como resultado de la investigación y análisis bibliográfico se lograron identificar y explicar:

Capítulo 1: Fundamentos de la Información, metodologías y tecnologías empleadas para darle solución a la propuesta.

- Los principales conceptos asociados a la propuesta de solución como son los estándares, los procesos de negocios, los modelos de negocios y los diagramas de procesos de negocios demostrándose no solo la estrecha relación entre ellos sino a su vez la que guardan con el tema.
- Se escogió la metodología de software ágil AUP-UCI enmarcando a la vez dicha propuesta de solución en el escenario cuatro de esta metodología. Se realizó un estudio minucioso de las diferentes herramientas y sus características.
- para el desarrollo del sistema se utilizó MYSQL, PhpMyAdmin, Apache 2.4.33 y la utilización de Visual Paradigm para la modelación de la propuesta de solución.

Capítulo 2: Descripción de la solución propuesta

En este capítulo se hace referencia a las principales características de la propuesta de solución y su funcionamiento como los requisitos funcionales y no funcionales se definen las historias de usuarios y los artefactos necesarios que servirán de base para la fase de implementación. Usando la información recopilada en el capítulo anterior se propone como solución el desarrollo de un portal web.

2.1. Modelo conceptual

El Modelo conceptual tiene como propósito organizar y representar el conocimiento de un área o campo específico asociado a un sistema de gestión o de información. Está orientado a identificar y describir la información del dominio de un problema. Es un artefacto útil que permite obtener conocimiento acerca de cómo se desenvuelve el problema en el contexto real. (Fowler y Scott, 1999)

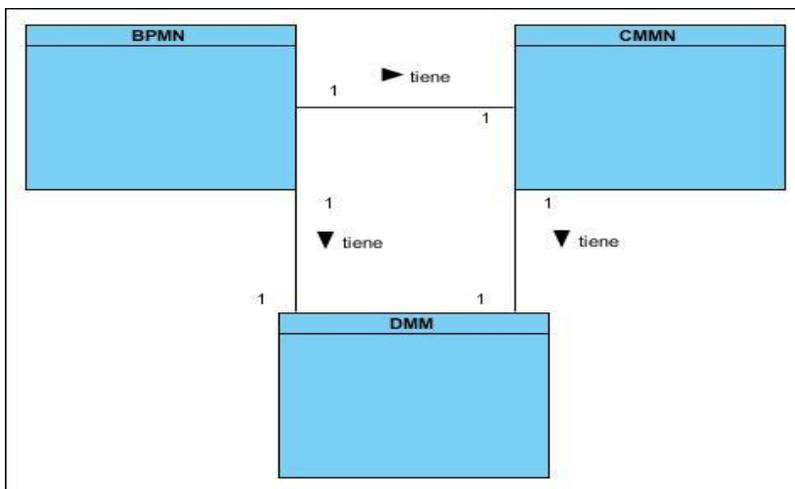


Figura 7. Diagrama del modelo conceptual para la integración de los estándares. Fuente: Elaboración Propia.

2.2. Especificación de los requisitos del software.

Los Requisitos funcionales son aquellos que describen cualquier actividad, los cuales establecen el comportamiento del software.

Se presenta la especificación de requisitos que son necesarios para la elaboración de la propuesta de solución. Fueron identificados 26 requisitos funcionales a partir de la encuesta aplicada a 4 trabajadores de gestión de software.

De cada requisito se indica la prioridad a partir de la importancia dada por el cliente

No	Requisitos	Prioridad	Descripción
RF1	Autenticar usuario	Medio	El usuario introduce los datos, el sistema comprueba si son correctos, en caso de serlo el usuario queda autenticado. Si los datos no son válidos el sistema muestra un mensaje de error.
RF2	Agregar rol al usuario	Medio	El sistema permite que los usuarios con rol administrador modifiquen los usuarios
RF3	Crear modelo BPMN	Medio	El sistema permite que los usuarios puedan crear un diagrama de tipo Bpmn
RF4	Mostrar modelo BPMN	Alta	El sistema permite a los usuarios ver todos los diagramas de tipo Bpmn.
RF5	Eliminar modelo BPMN	Medio	El sistema permite a los usuarios Eliminar el diagrama de tipo Bpmn.
RF6	Modificar modelo BPMN	Medio	El sistema permite a los usuarios modificar el diagrama de tipo Bpmn.
RF7	Importar archivo BPMN	Medio	El sistema permite a los usuarios importar un archivo de tipo Bpmn.
RF8	Exportar archivo BPMN	Medio	El sistema permite a los usuarios exportar un archivo de tipo Bpmn en imagen. jpg.
RF9	Sincronizar modelo BPMN	Medio	El sistema permite a los usuarios sincronizar el diagrama de tipo Bpmn con cualquier otro diagrama
RF10	Guardar modelo BPMN como imagen	Medio	El sistema permite a los usuarios guardar el diagrama de tipo Bpmn como imagen.jpg.
RF11	Crear modelo CMMN	Medio	El sistema permite que los usuarios puedan crear un diagrama de tipo cmmn

RF12	Mostrar CMMN	modelo	Alta	El sistema permite a los usuarios ver todos los diagramas de tipo cmmn
RF13	Eliminar CMMN	modelo	Medio	El sistema permite a los usuarios Eliminar los diagramas de tipo cmmn
RF14	Modificar CMMN	modelo	Medio	El sistema permite a los usuarios modificar el diagrama de tipo cmmn
RF15	Importar CMMN	archivo	Medio	El sistema permite a los usuarios importar un archivo de tipo cmmn.
RF16	Exportar CMMN	archivo	Medio	El sistema permite a los usuarios exportar un archivo de tipo cmmn en imagen. jpg.
RF17	Sincronizar CMMN	modelo	Medio	El sistema permite a los usuarios sincronizar el diagrama de tipo cmmn con cualquier otro diagrama
RF18	Guardar CMMN como imagen	modelo	Medio	El sistema permite a los usuarios guardar el diagrama de tipo cmmn como imagen.jpg.
RF19	Crear modelo DMN		Medio	El sistema permite que los usuarios puedan crear un diagrama de tipo dmn
RF20	Mostrar modelo DMN		Alta	El sistema permite a los usuarios ver todos los diagramas de tipo dmn
RF21	Eliminar modelo DMN		Medio	El sistema permite a los usuarios Eliminar los diagramas de tipo dmn
RF22	Modificar DMN	modelo	Medio	El sistema permite a los usuarios modificar el diagrama de tipo cmmn
RF23	Importar archivo DMN		Medio	El sistema permite a los usuarios importar un archivo de tipo dmn
RF24	Exportar DMN	archivo	Medio	El sistema permite a los usuarios exportar un archivo de tipo dmn en imagen. jpg

RF25	Sincronizar modelo DMN	Medio	El sistema permite a los usuarios sincronizar el diagrama de tipo dmn con cualquier otro diagrama
RF26	Guardar modelo DMN como imagen	Medio	El sistema permite a los usuarios guardar el diagrama de tipo dmn como imagen.jpg.

Tabla 1. Requisitos funcionales. **Fuente:** Elaboración propia

2.2.1. Requisitos no funcionales.

según Ian Sommerville (2) define el concepto de requerimientos no funcionales como que: “son las limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares”. Establece también que los requerimientos no funcionales se aplican al sistema como un todo y no a características individuales del sistema en sí. Los requerimientos establecen características necesarias del sistema y determinan las funcionalidades que debe poseer el mismo. Los requerimientos no funcionales suelen ser más críticos que los requerimientos funcionales, dado que la ausencia de uno de los mismos puede inutilizar al sistema.

usabilidad

- ✓ Rnf1. La aplicación debe presentar una interfaz amigable que permita la fácil interacción con la misma.
- ✓ Rnf2. Para utilizar la web es necesario que el dispositivo cuente con navegación a internet.

Seguridad

- ✓ Rnf3. En caso de que el sistema presente fallos los errores deben mostrar la menor cantidad de detalles posibles.
- ✓ Rnf4. El sistema mostrara mensajes de error.

Hardware

- ✓ Rnf5. El servidor de base de datos debe poseer una capacidad minima de 20 GB.
- ✓ Rnf6. El servidor de aplicaciones web debe poseer una capacidad de 80 GB.

- ✓ Rnf7. Los servidores web y de base de datos poseen un mínimo de 1 GB de memoria RAM.

Eficiencia

- ✓ Rnf8. El sistema debe demorar en cada transmisión menos de (8) segundos.

Software

- ✓ Rnf9. Servidor de base de datos MYSQL en su versión 5.7.24 o superior.
- ✓ Rnf10. Servidor web apache en su versión 2.4.
- ✓ Rnf11. Lenguaje de programación PHP 7 y como sistema de gestor de contenido Drupal 8.x.

2.3 Diagrama de clases del diseño.

El diagrama de clases del diseño, permite la representación gráfica y de forma estática la estructura general de un sistema, son clases del diseño con estereotipos web.

Requisito funcional Modificar modelo BPMN

- Descripción textual:
Para modificar una campaña la página servidora SP_ModificarModeloBPMN crea la página cliente CP_ModificarModeloBPMN que contiene el formulario Frm_ModificarModeloBPMN donde se modifican los datos y que será posteriormente ejecutado por la página servidora. Para acceder a los datos se hace a través de la clase controladora CC_ModificarModeloBPMN.
- Representación gráfica

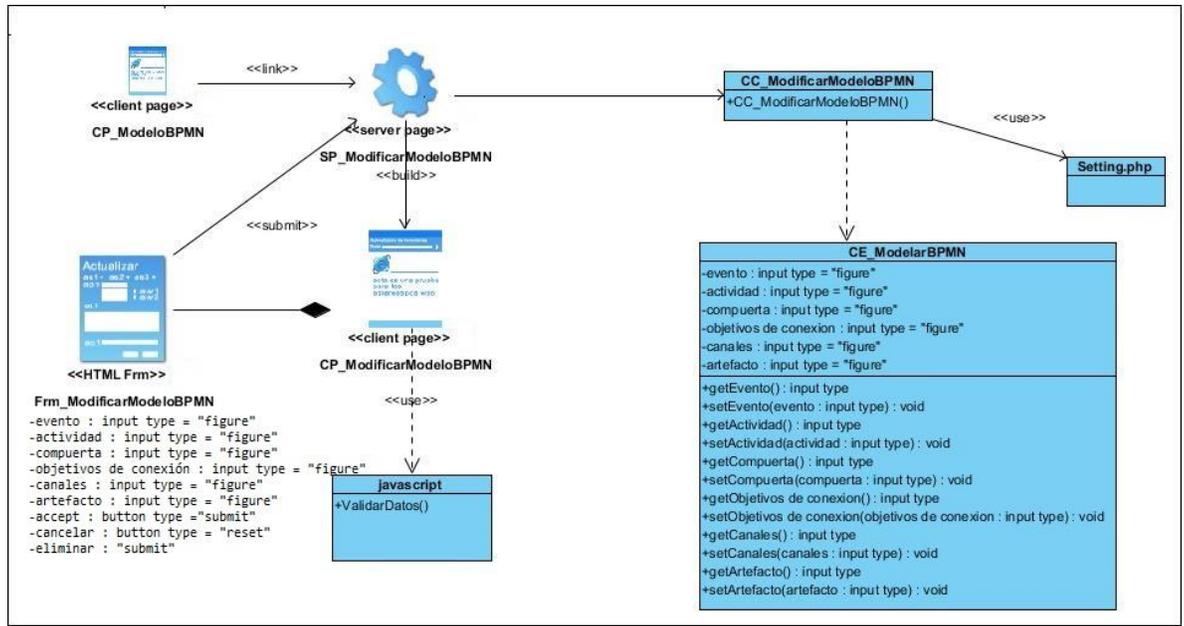


figura 8. Diagrama de clases de diseño del requisito Modificar modelo BPMN

- Descripción textual:

Para mostrar servicios se accede a la página cliente CP_MostraModeloBPMN construida por la página servidora SP_MostrarModeloBPMN que permite la visualización del contenido. Para acceder a los datos se hace a través de la clase controladora CC_MostrarModeloBPMN.

- Representación gráfica

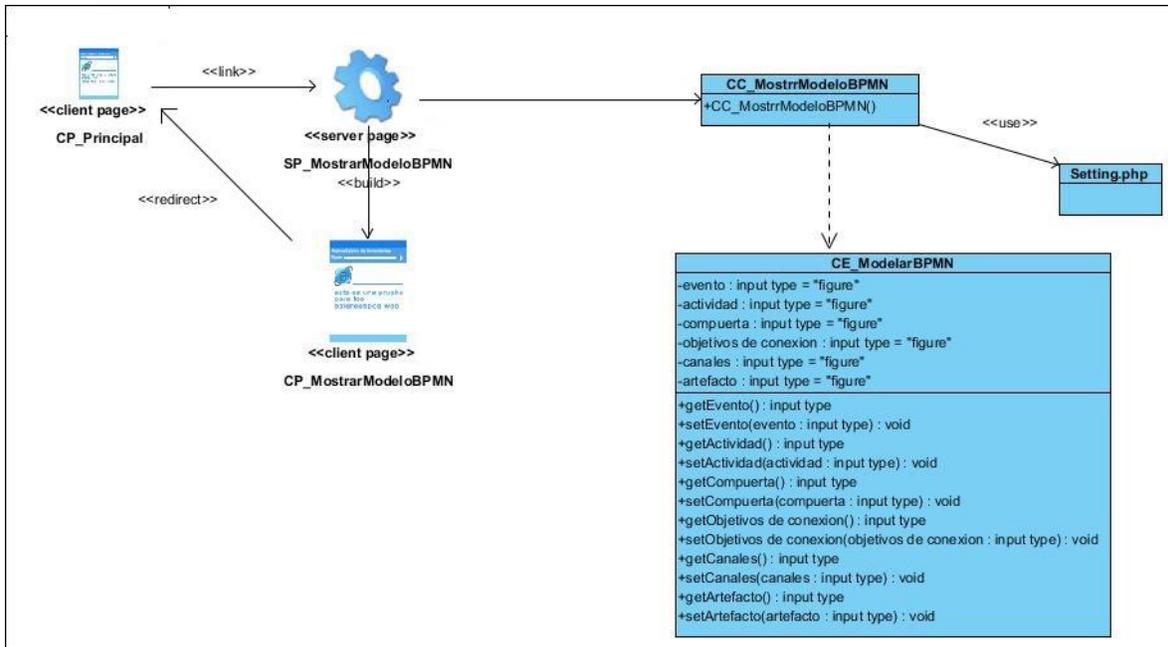


figura 9. Diagrama de clases de diseño del requisito Mostrar modelo BPMN

2.4 arquitectura

Los patrones arquitectónicos representan el nivel más alto dentro del sistema de patrones y expresan el esquema de la estructura fundamental de la organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos, especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. Cada patrón ayuda a lograr una propiedad específica del sistema global como es la adaptabilidad de la interfaz de usuario. (Almeira & Cavenago Pérez, 2007)

La arquitectura seleccionada para realizar el diseño del sistema a desarrollar, es la arquitectura basada en capas, ya que permite una mejor estructuración e independencia al sistema

2.5 Arquitectura basada en capas

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. (Carlos Pelaez, 2009). Posibilita lograr una mayor abstracción de la vista con respecto al modelo, sin dejar de detallar las relaciones entre las capas. Permite el encapsulamiento en lo referente a los tipos de datos, métodos, propiedades o implementación. Separa las funcionalidades

correspondientes a cada capa. Permite que el diseño del sistema al que se le aplica este patrón sea reutilizable, favorece además a la alta cohesión.

Los principales beneficios que se obtienen al utilizar este estilo arquitectónico son:

- **Abstracción:** Las capas permiten cambios que se realicen en un nivel abstracto. Usted puede incrementar o disminuir el nivel de abstracción usado en cada capa de la “pila” jerárquica.
- **Aislamiento:** El estilo de arquitectura de capas permite aislar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total.
- **Rendimiento:** Distribuir las capas entre múltiples sistemas (físicos) puede incrementar la escalabilidad, la tolerancia a fallos y el rendimiento.
- **Mejoras en Pruebas:** La capacidad de realizar pruebas se beneficia de tener unas interfaces bien definidas para cada capa, así como de la habilidad para cambiar a diferentes implementaciones de las interfaces de cada capa.
- **Independencia:** El estilo de arquitectura basado en capas el requerimiento de considerar el hardware y los problemas de instalación, así como las dependencias de interfaces externas.

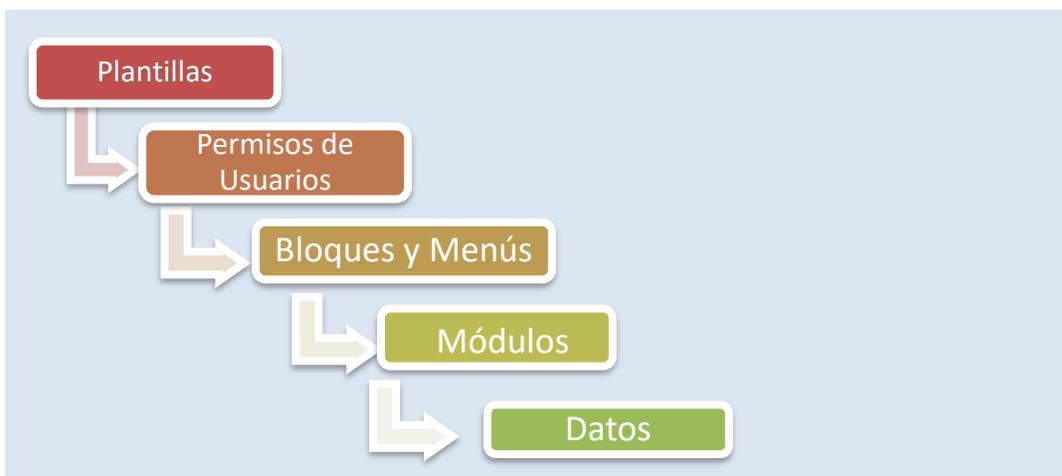


Figura 10. Estilo arquitectónica n capas de CMS Drupal. Fuente: CMS Drupal.

Al utilizarse el CMS Drupal para el desarrollo de la propuesta de solución, la arquitectura de *software* a utilizar es la definida por el mismo, la cual es una arquitectura 5-capas.

Plantillas: En la propuesta de solución la capa de plantilla es la que contiene su codificación en los lenguajes *JQuery* y PHP.

Permisos de usuarios: Donde se configuran los parámetros para determinar quiénes tienen acceso para ver y editar contenido o funciones del sitio.

Bloques y Menús: fragmentos de información que se muestran en las diferentes regiones de la plantilla de diseño

Módulos: se ubican los complementos desarrollados por la comunidad de Drupal que brindan o extienden las características del sistema

Datos: Son los elementos básicos donde el CMS Drupal almacena la información y los contenidos.

2.6 Historia de Usuario

Son descripciones cortas y simples de una característica contada desde la perspectiva de un usuario o cliente.

Tabla 2. Historias de usuario del requisito Modificar Modelo BPMN

Historia de Usuario	
Numero: 3	Nombre del requisito: Modificar modelo BPMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: Modificar modelo BPMN permite modificar un nuevo diagrama en el sistema. Luego de seleccionar la opción modificar modelo Bpmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Tabla 3. Historias de usuario del requisito Mostrar Modelo BPMN

Historia de Usuario	
Numero: 11	Nombre del requisito: mostrar modelo BPMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: Mostrar modelo BPMN permite mostrar un nuevo diagrama en el sistema. Luego de seleccionar la opción mostrar modelo bpmn el sistema muestra una interfaz donde se debe mostrar el diagrama.	
Observaciones: no aplica.	

Tabla 4. Historias de usuario del requisito Crear Modelo DMN

Historia de Usuario	
Numero: 19	Nombre del requisito: : Crear modelo DMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: Crear modelo DMN permite crear un nuevo diagrama en el sistema. Luego de seleccionar la opción agregar modelo dmn el sistema muestra una interfaz donde se debe modelar el diagrama .	
Observaciones: no aplica.	

2.7 Diagrama de Secuencia

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos(Tedeschi,2020).

A través de los diagramas de secuencia es posible:

- Representar los detalles de un caso de uso.
- Modelar la lógica de una operación, una función o un procedimiento.

- Ver cómo los objetos y los componentes interactúan entre sí para completar un proceso.
- Planificar y comprender la funcionalidad detallada de un escenario actual o futuro.

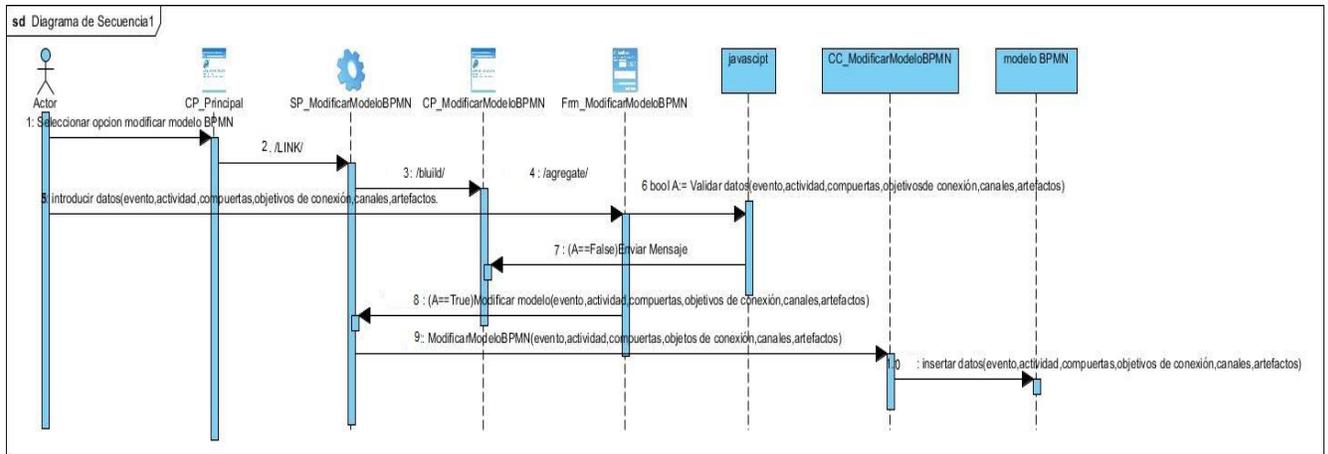


Figura 11. Diagrama de Secuencia del requisito Modificar modelo BPMN. Fuente: Elaboración propia

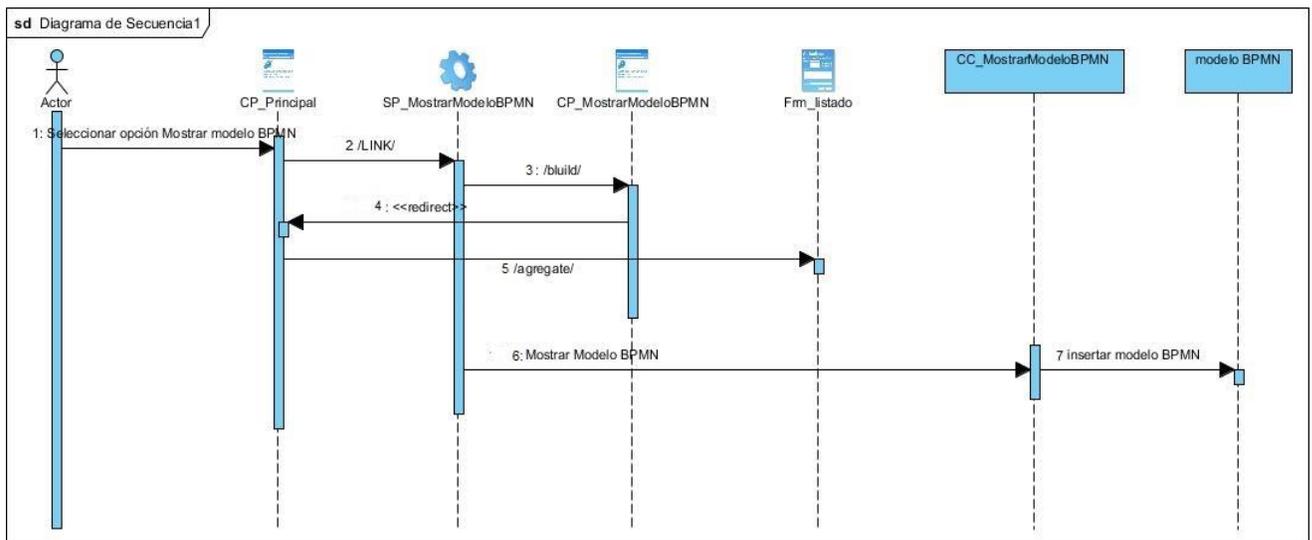


Figura 11. Diagrama de Secuencia del requisito Mostrar modelo BPMN. Fuente: Elaboración propia

Conclusiones

En el presente capítulo:

- Se realizó el diseño de la propuesta de solución, evidenciando los elementos que conforman el dominio del sistema para una mayor comprensión.
- Se analizaron los requerimientos funcionales y no funcionales que otorga las bases para la creación del sistema.
- Se analizaron las características y funcionalidades fundamentales las que fueron representadas mediante Historias de Usuarios.
- Se describió la arquitectura a utilizar en la propuesta de solución y los patrones de diseño a emplear en el desarrollo de la misma.

Todo lo anterior permitió el diseño de un sistema de manera correcta para pasar la posterior implementación del mismo.

Capítulo 3: Evaluación de la solución propuesta.

En este capítulo se definen los componentes que se abordan los componentes utilizados en la fase de implementación y prueba de la solución propuesta.

3.1 Estándares de codificación

Se define como estándares de codificación a un conjunto de pautas a seguir previamente establecidas para la escritura de código (Merkury, 2017). Los estándares de codificación forman parte de las buenas prácticas de programación, las cuales bien utilizadas incrementan la calidad del código.

3.2 Estándares de codificación de Drupal

Los estándares de programación de Drupal se aplican a código dentro de Drupal y a sus módulos contribuidos. Los estándares de programación de Drupal son independientes de versión y siempre actualizados. Todo nuevo código debería seguir los actuales estándares sin importar la versión

Sangrías y espacios en blanco

usar una sangría de 2 espacios, sin etiquetas. Las líneas no deben tener espacios en blanco al final. Los archivos deben tener el formato `\n` como final de línea (finales de línea de Unix), no `\r\n` (finales de línea de Windows).

Todos los archivos de texto deben terminar en un solo salto de línea (`\n`). Esto evita el parche de alerta “\ No existen saltos de línea al final del archivo” y hace a los parches más legibles ya que se hace claro que está siendo cambiado cuando se agregan líneas al final del archivo. (Negrita: No se le pudo dar una traducción coherente.

Operadores

Todos los operadores binarios (operadores que están entre dos valores), como `+`, `-`, `+=`, `!=",` `==`, `>`. Deben tener un espacio antes y después del operador, para facilitar la lectura. Por ejemplo, una asignación debe tener el formato `$foo = $bar`; en vez de `$foo = $bar`; Los operadores unarios (operadores que operan sobre un solo valor), como `++`, no deben tener espacios entre el operador y la variable o número que se utiliza.

Estructuras de control

Las estructuras de control deben tener un espacio entre las palabras claves (Keyword) y el paréntesis de apertura, para distinguirlas de la llamada de la función. Se utiliza siempre paréntesis de llave, incluso en situaciones en las que son técnicamente opcionales. Tenerlos incrementa la legibilidad y disminuye la probabilidad de introducir errores lógicos cuando nuevas líneas son agregadas.

Punto y coma

El lenguaje de PHP requiere puntos y comas al final de la mayoría de las líneas, pero permite ser omitidos al final del bloque de código. Los estándares de programación de Drupal los requiere, incluso al final de bloques de código. En particular para una línea de bloque PHP.

Comillas

Drupal no tiene un estándar para el uso de las comillas simples vs. Comillas dobles. Mientras sea posible, se debe mantener la consistencia en cada módulo, y respetar el estilo personal de otros desarrolladores.

Etiquetas de apertura y cierre de PHP: Cuando se utiliza el lenguaje *PHP*, siempre se deben utilizar las etiquetas `< ? PHP y ? >` *para* delimitar el código *PHP*, y en ningún caso la versión corta `<? y ?>`. En general se omite la etiqueta de cierre de *PHP* (`?>`) al final de los archivos. *module* y *.inc*. Esta convención evita que se puedan quedar olvidados espacios no deseados al final del archivo (después de la etiqueta de cierre `?>`), que serían identificados como salida *HTML* y podrían provocar un error muy típico, "*Cannot modify header information - headers already sent by...*" ²³.

```
<?php if (!empty($item)): ?>
    <p><?php print $item; ?></p>
<?php endif; ?>
```

```
<?php foreach ($items as $item): ?>
    <p><?php print $item; ?></p>
<?php endforeach; ?>
```

Figura 12. Ejemplo de sintaxis (Etiquetas de apertura y cierre de *PHP*).

3.3 Diagrama de despliegue

El diagrama de despliegue describe el despliegue físico de información generada por el software en los componentes de hardware. Muestra donde y como se desplegará el sistema.

El diagrama de despliegue representa de forma visual las relaciones físicas que existen entre los componentes de *software* y hardware en el portal. la solución desarrollada se compone por los elementos dispositivo_cliente representa el conjunto de dispositivos utilizados por los usuarios para visualizar la información que se encuentra en el servidor web a través de un navegador. La comunicación entre el dispositivo_cliente y el servidor web se establece utilizando el protocolo de comunicación segura *HTTPS*. El servidor de base de datos, se encarga de mantener las informaciones generadas como parte de la base de datos del *CMS Drupal* y permite el acceso a ella mediante el servidor de aplicación web. Estos dos servidores se comunican mediante el protocolo *TCP*. Además, establece una conexión con el servidor de correo a través del protocolo *SMTP*.

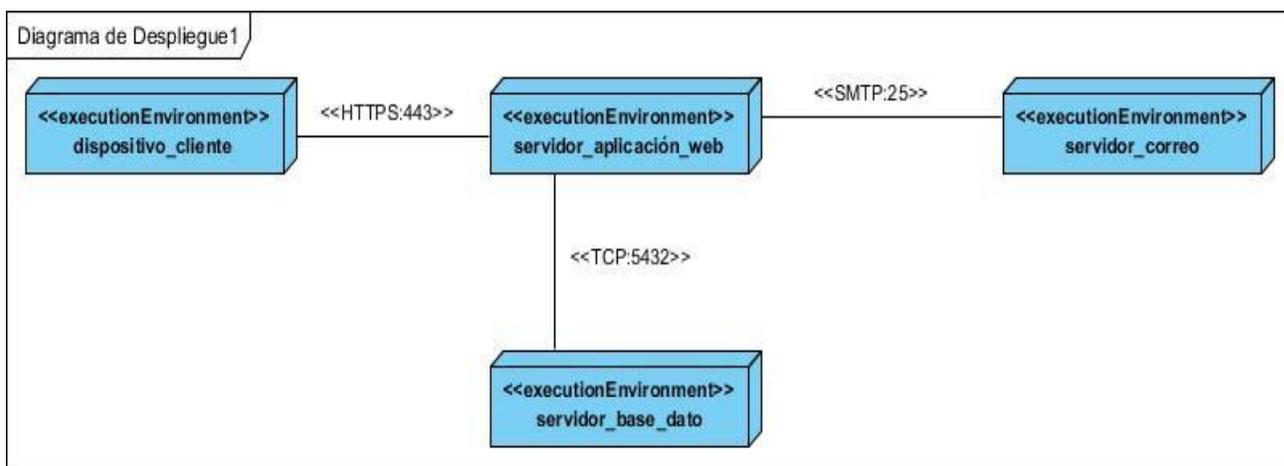


Figura 5. Diagrama de despliegue de la propuesta de solución (elaboración propia).

3.4 Patrones de diseño

Los patrones de diseño son formas estandarizadas de resolver problemas comunes de diseño en el desarrollo de software. Estos conforman un amplio catálogo de problemas y

soluciones, estandarizan la resolución de determinados problemas, condensan y simplifican el aprendizaje de las buenas prácticas y proporciona un vocabulario común entre desarrolladores. (Sánchez, 2017)

3.4.1 Patrones GRASP

Acrónimo de General Responsibility Assignment Software Patterns. Describen los principios fundamentales para asignar responsabilidades a los objetos. A continuación, se describen los cuatro principales patrones GRASP:

Experto: El experto en la información es la clase que tiene la información necesaria para cumplir la responsabilidad.

Creador: Es el responsable de crear una nueva instancia de una clase.

Bajo acoplamiento: el acoplamiento mide el grado en que una clase está conectada a otra, tiene conocimiento de otra o, de alguna manera, depende de otra. El bajo acoplamiento permite crear clases más independientes, más reutilizables, lo que implica mayor productividad.

Alta cohesión: la cohesión mide el grado en que están relacionadas las responsabilidades de una clase. Este asigna responsabilidades de manera que la cohesión se mantenga alta.

Controlador: El patrón "controlador" establece una clara separación entre la interfaz de usuario y el corazón o núcleo de procesamiento de la aplicación, donde se halla la lógica de negocios (Alem, 2014).

3.4.2 Patrones GOF

Los patrones GOF pertenecen al campo del Diseño Orientado a Objetos. Están conformados por 23 patrones que se clasifican según su propósito en creacionales, estructurales y comportamiento. (Christiansson, 2008)

Creacionales: definen la forma en la que un objeto puede ser creado teniendo en cuenta la reutilización y mutabilidad. Estos describen la mejor manera de manejar instancias.

Estructurales: describen cómo los objetos y las clases se pueden combinar para formar estructuras.

Comportamiento: describen la forma de cómo organizar, administrar y combinar conductas y responsabilidades de objetos, centrándose en la comunicación entre ellos.

3.5 Estrategia de Pruebas

El proceso de prueba se realiza de forma continua a lo largo del proceso de desarrollo con el fin de asegurar el éxito en el producto final. Las pruebas se llevan a cabo con la finalidad de demostrar que un programa hace lo que se infiere que haga, así como descubrir defectos en el programa antes de usarlo. Este proceso tiene dos metas distintas, demostrar al desarrollador y al cliente que el software cumple con los requerimientos y encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo a su especificación. Estas metas se cumplen a través de la realización de pruebas de validación y verificación. (Sommerville, 2011)

3.5.1 Pruebas de usabilidad

Las pruebas de usabilidad son procedimientos que se realizan mediante una lista de cheque aplicables a una web. El resultado después de realizar esta prueba puede llevar a cambios en el diseño de la web

Para poner en práctica las pruebas de usabilidad en la propuesta de solución se empleará la Evaluación Heurística: Consiste en valorar si los elementos de la interfaz del usuario de un sitio web o aplicación se corresponden con los principios de usabilidad establecidos a partir de una lista de principios o heurísticas. Esta técnica conseguirá detectar un gran número de problemas de usabilidad en sitios web con un menor coste que las técnicas que involucran usuarios. Para su evaluación general en sitios web existen criterios muy establecidos y probados que proporcionan la seguridad de que se está evaluando todo lo que es necesario evaluar (Araujo Falcón, 2014):

- ✓ Tipografía
- ✓ Navegabilidad
- ✓ Diseño, Estructura y Elementos de la Página
- ✓ Imágenes
- ✓ Contenido escrito para la aplicación

Se utilizará la lista de chequeo establecida por el Departamento de Calidad de la UCI para aplicaciones web para la aplicación de esta técnica, en ella se describen cuáles son los indicadores a evaluarse por cada una de las heurísticas descritas anteriormente.

3.5.2 Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra definidas por el cliente para cada historia de usuario y tienen como objetivo asegurar que las funcionalidades del sistema cumplen con lo que se espera de ellos.

La Junta Internacional de Cualificaciones de Pruebas de Software (ISTQB por sus siglas en inglés) define la “Aceptación” como: Pruebas formales con respecto a las necesidades del usuario, requerimientos y procesos de negocio, realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema (Báez, Fernanda, 2015).

3.5.3 Pruebas de rendimiento

Esta prueba es llevada a cabo por un grupo de pruebas independientes para asegurar que el rendimiento del sistema está dentro de los parámetros definidos. Se pueden utilizar herramientas para simular clientes y cargas pesadas, y se realizan las mediciones precisas de desempeño. El objetivo de las pruebas de rendimiento es determinar el rendimiento del sistema bajo una carga de trabajo definida utilizando diferentes tipos de pruebas de rendimiento tales como pruebas de carga y de estrés. Los tipos de pruebas de rendimiento que se le realizaron a la propuesta de solución fueron las pruebas de carga y estrés y se utilizó como herramienta el Apache JMeter en su versión 2.10, diseñado para pruebas de carga de comportamientos funcionales y la medición del rendimiento.

- **Pruebas de rendimiento baja intensidad:** Durante la fase de evaluación, cuando ejecuta pruebas funcionales a pequeña escala (pruebas de integración de unidad o componente), es útil supervisar los tiempos de respuesta de transacciones individuales.
- **Pruebas de rendimiento de alta intensidad:** Las pruebas de rendimiento de alta intensidad se pueden considerar como pruebas de rendimiento instrumentadas. Esta fase de las pruebas se realiza en escenarios completos, tales como pruebas de estrés y pruebas de estabilidad realizadas en procesos de negocio globales.

- **Visualizar restricciones del rendimiento:** Si el entorno de prueba consta de interacciones complejas entre componentes basadas en mensajes, las pruebas de rendimiento comprobarán que estos mensajes fluyen por el sistema a la velocidad necesaria. Para obtener una mejor comprensión de este concepto, es importante visualizar cómo los componentes procesan estos mensajes (Sayago, 2019).

3.5.4 Pruebas de caja blanca

Las pruebas de caja blanca o pruebas estructurales son un enfoque sistemático a las pruebas donde se usa el conocimiento del código fuente del programa para diseñar pruebas de defecto. Tienen como meta diseñar pruebas que proporcionen algún nivel de cobertura de programa. Estas deben garantizar que se ejecute toda ruta lógica a través del programa, con la consecuencia de que cada enunciado del programa se efectúe al menos una vez. (Sommerville, 2011)

Las pruebas de caja blanca se realizan a partir de la revisión de las rutas lógicas del código fuente a evaluar, para esto se utiliza la técnica de camino básico o ruta básica. Esta es una técnica que permite al diseñador de casos de pruebas, derivar una medida de complejidad lógica de un diseño de procedimiento y usar esta medida como guía para definir un conjunto básico de rutas de ejecución. (Pressman, 2010)

3.5.4 Métodos de caja negra. Técnicas de partición equivalente

El método de pruebas de caja negra, también llamado pruebas de comportamiento, se enfocan en los requerimientos funcionales del *software*; es decir, las técnicas de prueba de caja negra le permiten examinar el programa para que cuente con todas las funcionalidades analizando los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. Con este método se intenta encontrar los errores: de inicialización y terminación, de interfaz y en las estructuras (Peño Sánchez, 2015).

Existen varias técnicas que se pueden aplicar al método de caja negra, pero en la presente investigación se empleará partición equivalente, el cual consiste en ejecutar el flujo básico de las funcionalidades utilizando datos validos e inválidos (Pressman, 2005).

Conclusiones

En el presente capítulo se describieron y mostraron:

- La aplicación de los patrones en el desarrollo del sistema y el uso de los estándares de codificación que facilitan la reutilización del código y mejora el entendimiento de las clases.
- Se expuso el diagrama de despliegue y la descripción del mismo.
- Se mencionan los conceptos y funcionalidades de las pruebas a realizar al igual que los patrones de diseños.

Conclusiones Generales

- ✚ En la presente investigación se expusieron los conceptos asociados a los sistemas de Aplicación Web para la Exportación de los estándares BPMMN, CMMN y DMN para el modelado de procesos de negocio sistema a desarrollar.
- ✚ Se escogió la metodología de software ágil AUP-UCI enmarcando a la vez dicha propuesta de solución en el escenario cuatro de esta metodología como metodología a seguir debido a las ventajas expuestas con anterioridad de dicha metodología y las características de la presente investigación.
- ✚ Se construyó el diseño del sistema cumpliendo los patrones de diseño y la arquitectura adecuada, propiciando la creación del mismo siendo capaz de cumplir el levantamiento de requisitos realizados.
- ✚ Se procedió al análisis de los conceptos de los casos de pruebas, las cuales permitieron valorar de manera positiva el desempeño de la aplicación demostrando que estaría lista para la exportación de los estándares BPMMN, CMMN y DMN.
- ✚ Todo lo anterior contribuyó al cumplimiento de los objetivos planteados al anhelar un sistema sobre la aplicación Web para la Exportación de los estándares BPMMN, CMMN y DMN para el modelado de procesos de negocio.

RECOMENDACIONES

En función del constante proceso de mejora y evolución que es inherente a todo sistema de software se recomienda lo siguiente:

Realizar las pruebas de aceptación para validar con el cliente el portal desarrollado.
Continuar desarrollando funcionalidades que sean requeridas en el futuro respondiendo a las nuevas necesidades y garantizando la disponibilidad y usabilidad.

Bibliografía

- Alem, F. (9 de diciembre de 2014). *Patrones de Diseño (GRASP):Taringa*. Recuperado el 28 de marzo de 2019, de Patrones de Diseño (GRASP): <https://www.taringa.net>
- Almeira, A., & Cavenago Pérez, V. (2007). *Arquitectura de Software: Estilos y Patrones. vol. 28. Arquitectura de Referencia para el diseño y despliegue de Nubes Privadas*. García Perellada, Lilia
- Carlos Pelaez, J. (30 de Mayo de 2009). *Arquitectura basada en capas: Geeks.ms*. Recuperado el 10 de Marzo de 2019, de Geeks.ms: <https://geeks.ms>
- Christiansson, B. (2008). *GoF Design Patterns-with examples using Java and UML2*.
- Calás Torres, Abraham, Saria Preval, Katia y Suarez Riquenes, Ricardo E. 2015. *Arquitectura de referencia para php*. Ciudad de La Habana, Cuba : s.n., 2015.
- Carlos Pelaez, Juan. 2009. *Arquitectura basada en capas: Geeks.ms. Geeks.ms*. [En línea] 30 de Mayo de 2009. [Citado el: 10 de Marzo de 2019.] <https://geeks.ms>.
- Chuburu, Laura. 2018. *Qué es JQuery y cómo implementarlo: Laura Chuburu. Laura Chuburu*. [En línea] 2018. [Citado el: 24 de Febrero de 2019.] <https://www.laurachuburu.com.ar>
- Lucidchart. (2016). *Qué es un modelo de base de datos:Lucidchart*. Recuperado el 15 de Abril de 2019, de Lucidchart: <https://www.lucidchart.com>
- Merkury. (12 de Enero de 2017). *Estándares de codificación:Ohmyroot*. Recuperado el 1 de Mayo de 2019, de Ohmyroot: <https://www.ohmyroot.com>
- Sánchez, M. Á. (22 de Noviembre de 2017). *Patrones de Diseño de Software*. (A Medium Corporation) Recuperado el 17 de Marzo de 2019, de Medium: <https://medium.com/all-you-need-is-clean-code/patrones-de-dise%C3%B1o-b7a99b8525e>
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE Novena edición*. México: Pearson.
- Sánchez, Miguel Ángel. 2017. *Patrones de Diseño de Software. Medium*. [En línea] A Medium Corporation, 22 de Noviembre de 2017. [Citado el: 17 de Marzo de 2019.] <https://medium.com/all-you-need-is-clean-code/patrones-de-dise%C3%B1o-b7a99b8525e>.
- ICTEA. 2019. *Qué es el lenguaje de programación PHP: ICTEA. Qué es el lenguaje de programación PHP*. [En línea] ICTEA, 2019. [Citado el: 7 de Mayo de 2019.] <http://www.ictea.com>.
- Fontela, Alvaro. 2015. *¿Que es Bootstrap?:Raiola Networks. ¿Que es Bootstrap?* [En línea] 16 de julio de 2015. [Citado el: 15 de mayo de 2019.] <https://raiolanetworks.es>.

- Kord, Maia. 2011. Patrones GRASP. Patrones GoF. Diferencias: TuxNots. *TuxNots*. [En línea] 3 de Diciembre de 2011. [Citado el: 20 de Febrero de 2019.] <https://sites.google.com/site/tuxnots>.
- Larman, Craig. 2004. *UML y patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Madrid : Pearson, 2004. 84-205-3438-2.
- Pérez Valdés, Damián. 2007. ¿Qué es Javascript?: Maestros de la web. *Maestros de la web*. [En línea] 2007. [Citado el: 12 de Abril de 2019.] <http://www.maestrosdelweb.com>.
- PostgreSQL Global Development Group. 2018. PostgreSQL 11 Released!:postgresql. *PostgreSQL 11 Released!* [En línea] 18 de octubre de 2018. [Citado el: 20 de abril de 2019.] <https://www.postgresql.org>.
- . 2019. PostgreSQL: The World's Most Advanced Open Source Relational Database:Postgresql. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. [En línea] 23 de mayo de 2019. [Citado el: 25 de junio de 2019.] <https://www.postgresql.org>.
- Úbeda González, Raúl. 2009 . *Métodos ágiles para el desarrollo de software*. Catalunya : Universitat Politècnica de Catalunya, 2009 .
- Vega, Elio. 2016. 6 Ingenieria de requerimientos:Quizlet. *6 INGENIERIA DE REQUERIMIENTOS*. [En línea] 2016. [Citado el: 28 de mayo de 2019.] <https://quizlet.com>.
- EXPERTOS EN DRUPAL. TutorialDrupal. [En línea] [Citado el: 27 de junio de 2020.] <https://www.cursosdrupal.com>.
- Drupal.org. [En línea] [Citado el: 26 de junio de 2020.] <https://www.drupal.org/> . .
- Cerón, Jose Antonio Dorado. Drupalera. [En línea] 27 de marzo de 2015. [Citado el: 27 de junio de 2020.]

ANEXOS

Historia de Usuario	
Numero: 7	Nombre del requisito: importar archivo BPMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: importar archivo BPMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción importar modelo Bpmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Historia de Usuario	
Numero: 15	Nombre del requisito: importar archivo CMMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: importar archivo CMMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción importar modelo cmmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Historia de Usuario

Numero: 23	Nombre del requisito: importar archivo DMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: importar archivo DMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción importar modelo dmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Historia de Usuario	
Numero: 8	Nombre del requisito: exportar archivo BPMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: exportar archivo BPMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción exportar modelo Bpmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Historia de Usuario	
Numero: 16	Nombre del requisito: exportar archivo CMMN

Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: exportar archivo CMMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción exportar modelo cmmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	

Numero: 16	Nombre del requisito: exportar archivo DMN
Usuario: Dismaray Pérez Calvi	Iteración asignada: 1ra
Prioridad en Negocio: Media	Tiempo estimados: 24h
	Tiempo Reales: 12h
Descripción: exportar archivo DMN permite importar un nuevo diagrama en el sistema. Luego de seleccionar la opción exportar modelo dmn el sistema muestra una interfaz donde se debe modificar el diagrama.	
Observaciones: no aplica.	