



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

**Desarrollo del entorno base y
módulos transaccionales: Gestión de
Procesos y Seguimiento, para el
Sistema de Información para
Gestión de la Calidad (SIGC).**

**Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela por el
Br. Arturo Jiménez (C.I. 20.116.026)
para optar al título de Licenciado en Computación**

Tutor: Prof. Jaime Parada

Ciudad Universitaria de Caracas, 27 de mayo de 2016

**UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN**

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "Desarrollo del entorno base y módulos transaccionales: Gestión de Procesos y Seguimiento, para el Sistema de Información para Gestión de la Calidad (SIGC)" y presentado por el Br. **Arturo Raúl Jiménez Melo C.I. V-20.116.026**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 27 de Mayo de 2016, a las 9:00am, para que el autor lo defendiera en forma pública, lo que este hizo en planta alta III de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 19 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 27 de Mayo de 2016



**Prof. Jaime Parada
(Tutor)**



**Prof. Carlos Acosta
(Jurado Principal)**



**Ing. Carmen Salazar
(Jurado Principal (E))**

AGRADECIMIENTOS

A Dios, por haberme acompañado y guiado a lo largo de mi carrera, por ser mi fortaleza en los momentos de debilidad y por brindarme una vida llena de aprendizajes y experiencias.

Le doy gracias a mis padres Aurora y Leonardo, por apoyarme incondicionalmente en todo momento, por los valores que me han inculcado, por su constancia, disciplina, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo por ser un excelente ejemplo a seguir.

A mis hermanos Evelyn y Octavio por ser parte de mi vida, por ser mi ejemplo de desarrollo profesional a seguir, por sus consejos, ayuda y apoyo.

A mis sobrinos Valeria, Flavia y Adrián, porque desde que llegaron a mi vida cada sonrisa y cada experiencia me hicieron crecer y sentirme afortunado de tenerlos.

A Keysi, por estar a mi lado y brindarme su apoyo estos últimos años en todo lo que me propongo. Por su comprensión y amor. Por hacer cada día a su lado único y por impulsarme a ser mejor.

A mis amigos, porque sin ellos la universidad no hubiese sido igual, por haber hecho esta etapa un trayecto de vivencias que nunca olvidaré.

A mis profesores y tutores, por su guía durante toda mi carrera profesional, por el tiempo dedicado a mi aprendizaje constante y porque cada uno aportó su granito de arena para esta meta.

A la Universidad Central de Venezuela, por darme la oportunidad de estudiar y ser un profesional.



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Desarrollo del entorno base y módulos transaccionales: Gestión de Procesos y Seguimiento, para el Sistema de Información para Gestión de la Calidad (SIGC).

Autor: Arturo Raúl Jiménez Melo

C.I.: 20.116.026

Correo-e: Arturo.jimenezucv@gmail.com

Tutor: Prof. Jaime Parada

Correo-e: Jaime.parada@gmail.com

Fecha: 27 de mayo de 2016.

RESUMEN

El objetivo del presente trabajo especial de grado consiste en describir la aplicación de las herramientas de software y la metodología de trabajo utilizada para solventar la situación actual de la Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A., la cual lleva de manera manual el proceso de gestión de la calidad. Dentro de este contexto, en este trabajo se presenta el desarrollo de una solución tecnológica web establecida sobre el manejador de contenidos Drupal, PostgreSQL como sistema gestor de base de datos, D3.js como biblioteca JavaScript para la generación de gráficos, SVN para el control de versiones y apache como servidor. Logrando así un sistema que automatiza y agiliza el enfoque basado en procesos y el seguimiento, del sistema de gestión de la calidad basado en la ISO 9001:2008, que sirve de base para el mejoramiento continuo de los procesos y el cumplimiento de sus objetivos, permitiendo el acceso desde cualquier parte de la organización, y el conocimiento de la gestión de los procesos y el seguimiento de los mismos a través de la navegación por su estructura organizacional y mapas de procesos asociados.

Palabras Clave: Sistema de Gestión de Calidad, Calidad, ISO 9001:2008.

INDICE.

INTRODUCCIÓN.....	1
CAPÍTULO I	2
1. PLANTEAMIENTO DEL PROBLEMA.	2
1.1. SITUACIÓN ACTUAL.....	2
1.2. ANTECEDENTES.	3
1.3. IMPORTANCIA.	4
1.4. OBJETIVO GENERAL DEL TRABAJO ESPECIAL DE GRADO.....	5
1.5. OBJETIVOS ESPECÍFICOS.	5
1.6. ALCANCE.....	5
1.7. RESULTADOS ESPERADOS.	6
CAPITULO II.....	7
2. MARCO TEÓRICO.....	7
2.1. CALIDAD.....	7
2.2. NORMATIVA ISO.....	7
2.3. SISTEMA DE GESTIÓN DE LA CALIDAD	9
2.4. SEGUIMIENTO.	11
2.5. INDICADOR.	11
2.6. MAPA DE PROCESO.	11
2.7. HERRAMIENTAS DE DESARROLLO DE SOFTWARE.	12
CAPITULO III.	18
3. MARCO METODOLOGICO.	18
3.1. METODOLOGÍA DE LA RED NACIONAL DE INTEGRACIÓN Y DESARROLLO DE SOFTWARE LIBRE (MERINDE)	19
3.1.1. ESTRUCTURA.	20
3.1.2. FASES.	23
3.1.3. ROLES.....	25
CAPITULO IV.	27
4. DESARROLLO DE LA APLICACIÓN.	27
4.1. PLATAFORMA UTILIZADA EN EL DESARROLLO.....	27
4.2. PLATAFORMA NECESARIA PARA LA EJECUCIÓN DE LA APLICACIÓN.....	27
4.3. METODOLOGÍA.	27
CONCLUSIONES.	71
REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES.	72

INTRODUCCIÓN

La Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A., utiliza un sistema de trabajo que no aprovecha las tecnologías existentes para llevar a cabo las diferentes actividades relacionadas con la gestión de la calidad. Muchas de las tareas diarias son realizadas de forma manual, lo cual afecta directamente el tiempo de respuesta en comparación a los que puede proporcionar un sistema automatizado desarrollado conforme a sus necesidades.

Identificar, entender y gestionar los procesos interrelacionados como sistema contribuye a la eficacia y eficiencia de la organización en el logro de sus objetivos.

Esto conlleva a plantear una solución para optimizar el manejo de las tareas cotidianas y automatizar el manejo de los recursos. El presente trabajo especial de grado tiene como finalidad diseñar y desarrollar una aplicación web que apoye el sistema de gestión de la calidad de la empresa, que permita el acceso desde cualquier organización, al conocimiento de la gestión de los procesos y el seguimiento de los mismos a través de la navegación por su estructura organizacional y mapas de procesos asociados.

El diseño y desarrollo se basaron en los objetivos y necesidades de la empresa así como de su estructura. El contenido se enfoca en la implementación dentro de la empresa basada en la gestión de calidad que propone la norma ISO 9001:2008.

CAPÍTULO I

1. PLANTEAMIENTO DEL PROBLEMA.

La Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A., requiere una solución tecnológica de software, que le permita gestionar, de forma automatizada, todos los procesos de la organización y sus sistemas de gestión de calidad, basados en las premisas de ISO 9001, entre otras normas de calidad. Dando prioridad a los módulos de Gestión de Procesos y Seguimiento.

1.1. Situación actual.

En la actualidad la Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A. no cuenta con un software de gestión que apoye en el seguimiento, gestión y difusión de los procesos que regularmente conforman una organización. Esto ocasiona generación de no conformidades sobre los sistemas de gestión de calidad existentes, según auditorías basadas en los requisitos de la norma ISO 9001.

Dada la importancia estratégica que tiene para el futuro de la organización la automatización del sistema de gestión de calidad, surge la necesidad de establecer un plan para la implementación de un software de apoyo para el sistema de gestión de la calidad, basado en la ISO 9001, que sirva de base para el mejoramiento continuo de los procesos. Es importante destacar que por ser PDVSA una empresa del Estado, debe seguir las premisas de la Ley de Infogobierno, cuyo objetivo principal es establecer los principios, bases y lineamientos que rigen el uso de las tecnologías de información en el Poder Público y el Poder Popular, para mejorar la gestión pública y los servicios que se prestan a las personas, instando a promover el desarrollo de las tecnologías de información sin restricciones de licencia en el Estado.

Bajo estas premisas, se cuenta con un trabajo previo cuyo resultado fue el levantamiento de requisitos de toda la solución, su arquitectura y diseño de datos en primer nivel [1]. Este estudio previo arrojó que la solución tecnológica debe considerar los siguientes módulos:

1. Gestión de Procesos.
2. Seguimiento.
3. Control de Documentos.
4. Gestión de Carpetas.
5. Revisión por Dirección.
6. Análisis y Mejoras.
7. Centro de Mensajes.
8. Programa de Reuniones.

Conforme a esto la gerencia da prioridad al desarrollo de los dos primeros módulos y así sirvan de base para los demás módulos.

1.2. Antecedentes.

La familia de normas ISO 9000 son normas de calidad establecidas por la Organización Internacional para la Estandarización (ISO), las cuales se elaboraron para asistir a las organizaciones de todo tipo y tamaño. Están compuestas de estándares y guías relacionados con sistemas de gestión y de herramientas específicas, para verificar que los sistemas de gestión cumplan con las exigencias requeridas. Hoy en día, la gran mayoría de las organizaciones aplican las normas de la serie ISO 9000 orientadas a la mejora continua basada en procesos, para garantizar la calidad de un producto o servicio.

El documento de Especificación de Requisitos del Software (ERS) tiene como objetivo general realizar el levantamiento de requisitos para una solución tecnológica de software de gestión de la calidad basado en la norma ISO 9001:2008 para la Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A. Este documento contiene todos los requisitos del sistema, describe sus funciones, requisitos no funcionales, características del diseño, y otros elementos necesarios para proporcionar una descripción completa y comprensiva de los requisitos para el software a desarrollar. El resultado de este trabajo fue un correcto levantamiento de requisitos que cumple lo solicitado por la Gerencia Sistema de Gestión de Calidad para la herramienta de software. [1]

1.3. Importancia.

El gran poder de difusión que nos ofrece la plataforma de Internet se ha convertido en la herramienta ideal para manejar grandes volúmenes de información y al alcance de un alto número de personas.

Por otro lado, hoy en día no se concibe la actividad gerencial sin el uso de información oportuna, exacta y objetiva. Los sistemas de información son ya una necesidad en cada una de las empresas para:

- Tener un control más efectivo de los procesos de la organización.
- La integración de las diferentes áreas que conforman la empresa.
- Incrementar la efectividad en la operación de procesos que interactúan.
- Proporcionar ventajas competitivas y valor agregado.
- La disponibilidad de mayor y mejor información para los usuarios en tiempo real.
- La eliminación de la barrera de la distancia trabajando con un mismo sistema en puntos distantes.
- Disminución de errores, tiempo y recursos, pues permite comparar resultados alcanzados con los objetivos programados, con fines de evaluación y control.

Buscar la información de un sistema de registro en papel tomaría mucho más tiempo; son más propensos a la pérdida de datos y son más costosos de producir que, el tiempo y los gastos que se necesitan para introducirlos.

Por lo tanto, PDVSA INTEVEP S.A. como brazo tecnológico de la corporación y específicamente la Gerencia Departamental de Sistemas de Gestión de la Calidad (STSG) requiere de un sistema de información que cubra los beneficios expuestos anteriormente.

Con la implementación de los módulos de gestión de procesos y el módulo de seguimiento, desarrollados bajo una metodología y herramientas de software de desarrollo sin restricciones de licencia, se pretende obtener productos software que den solución a la problemática planteada.

1.4. Objetivo general del Trabajo Especial de Grado.

Desarrollar el entorno base y los módulos transaccionales: Gestión de Procesos y Seguimiento, para el Sistema de Información para Gestión de la Calidad (SIGC).

1.5. Objetivos específicos.

1. Definir el diagrama de objetos del sistema.
2. Desarrollar el entorno de mapa de procesos, de acuerdo al "Enfoque basado en procesos" de la norma ISO 9001:2008, para el sistema de gestión de la calidad de PDVSA-INTEVEP S.A.
3. Facilitar a los usuarios la creación de la estructura organizacional en el sistema.
4. Desarrollar la asociación de la estructura del Plan Operativo Anual (POA) de PDVSA-INTEVEP S.A. con la estructura organizacional y todo el sistema de gestión de la calidad.
5. Desarrollar la gestión de contenido a través del módulo de administrador.
6. Desarrollar interfaces de usuario de la aplicación, enmarcado en los lineamientos de diseño de software establecidos por la organización.
7. Aplicar la metodología de desarrollo de software libre MeRinde para el proceso de desarrollo de software y su documentación.
8. Elaborar el manual de usuario de la aplicación.

1.6. Alcance.

Con la realización de este trabajo especial de grado, se quiere desarrollar un sistema web que soporte múltiples usuarios y pueda ser accedido desde distintas ubicaciones. Permitiendo la gestión de la calidad y facilitando la creación de la estructura organizacional.

Contemplando el desarrollo de dos módulos, el primero llamado *Gestión de Procesos* que permite moldear la estructura organizacional con sus procesos y procedimientos asociados, incentivando la navegación a través de ésta en el sistema, y el segundo llamado *Seguimiento*, que permitirá asociar la estructura de objetivos e indicadores a cada uno de los niveles de la estructura organizacional según corresponda, basándose en el modelo del Plan Operativo Anual (POA) de la organización.

1.7. Resultados esperados.

Ofrecer un sistema web a la Gerencia Departamental del Sistema de Gestión de Calidad de PDVSA – INTEVEP S.A. que permita gestionar, de forma automatizada, todos los procesos de la organización y sus sistemas de gestión de calidad, basados en las premisas de ISO 9001, entre otras normas de calidad, que permita la navegación del mismo a través de la estructura organizacional y los mapas de procesos asociados.

CAPITULO II

2. MARCO TEÓRICO.

En este capítulo se busca definir los conceptos asociados al Trabajo Especial de Grado, para tener una visión general de los términos que se usarán en el desarrollo del sistema, los cuales ayudarán a entender e interpretar los resultados que se obtendrán del mismo.

Los tópicos a tratar en este trabajo, cuya comprensión es fundamental están relacionados con la norma ISO 9001 y la gestión de procesos.

Es necesario destacar que en este caso se incluyen los fundamentos de la Normativa ISO 9001:2008, necesario para desarrollar el Sistema de Gestión de la Calidad.

2.1. Calidad.

Según el modelo de la norma ISO 9000, la calidad es el "grado en el que un conjunto de características inherentes cumple con los requisitos", entendiéndose por requisito "necesidad o expectativa establecida, generalmente implícita u obligatoria" [2]. Considerando esto, se puede decir que calidad es el conjunto de propiedades y características de un producto o servicio que permiten satisfacer necesidades, gustos y preferencias, y de cumplir con expectativas del consumidor.

2.2. Normativa ISO

La globalización ha influenciado directamente los procesos de estandarización, siendo necesario aplicar técnicas y lenguajes homogéneos en un mundo cada vez más global y competitivo. La mayoría de los Sistemas de Gestión de la Calidad están siendo aplicados a partir de las normativas de la Organización Internacional de Estandarización, ISO por sus siglas en inglés. Esta organización nace en 1946 en Ginebra, después de la Segunda Guerra Mundial. El objetivo fundamental de esta organización es promover el desarrollo de normas internacionales que permita homogeneizar las normas utilizadas a nivel internacional. Es un organismo no gubernamental por lo que no presenta dependencia de ninguna entidad,

además de que dichas normas no son impuestas en ninguna nación, sino que las entidades se acogen voluntariamente a ellas. La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costes y efectividad.

ISO 9000:

La serie ISO 9000 es un conjunto de normas orientadas a ordenar la gestión de la empresa que ha ganado reconocimiento y aceptación internacional debido al mayor poder que tienen los consumidores y a la alta competencia internacional acentuada por los procesos integracionistas. Algunas de estas normas especifican requisitos para sistemas de calidad (como la norma ISO 9001) y otras dan una guía para ayudar en la interpretación e implementación del sistema de calidad. El objetivo de este conjunto de normas es proporcionar elementos para que una organización pueda lograr la calidad del producto o servicio, y mantenerla en el tiempo, de manera que las necesidades del cliente sean satisfechas permanentemente. Además, la ISO 9000 contiene las directrices para seleccionar y utilizar las normas para el aseguramiento de la calidad [3].

ISO 9001:

Es una norma internacional de requisitos que se aplica a los Sistemas de Gestión de la Calidad (SGC) y que se centra en todos los elementos de administración de calidad con los que una empresa debe contar para tener un sistema efectivo que le permita administrar y mejorar la calidad de sus productos o servicios. Posee ocho cláusulas principales que especifican los requisitos mínimos para el diseño e implementación de un SGC. El propósito de esta norma no es proporcionar uniformidad en la documentación o en la estructura del sistema, puesto que cada organización posee determinados procesos, objetivos y necesidades particulares, simplemente orienta en la evaluación de la capacidad de la organización para cumplir con los requisitos del cliente y sus reglamentos propios. Las tres primeras cláusulas son de carácter introductorias y ellas se encuentran en las declaraciones de principios, normativas y definiciones. Las últimas 5 cláusulas se refieren al proceso como tal, y en ellas se establecen los requisitos para poder implantar el Sistema de Gestión de la Calidad [4].

2.3. Sistema de Gestión de la Calidad

Un Sistema de Gestión de la Calidad (SGC) reúne un conjunto de elementos mediante los cuales se administra de forma planificada la calidad de una organización, incluyendo empleados, departamentos y/o actividades que se desarrollen en ella. El objetivo principal que persigue todo Sistema de Gestión está enfocado en garantizar la mejora continua de la calidad mediante un mejor conocimiento y control de todos los procesos que se lleven a cabo dentro de la organización, logrando satisfacer los requisitos del cliente [5].

Existen diversas normativas que rigen todo Sistema de Gestión de la Calidad, las cuales están orientadas a mejorar los aspectos organizativos de una empresa u organización. De acuerdo a estas normativas todo sistema debe ser manejado bajo ciertos principios fundamentales, los cuales se muestran a continuación según la ISO 9001:2008 [6]:

- **Enfoque al cliente:** el cliente es la razón por la cual una organización existe y por tal motivo, se debe comprender las necesidades presentes y futuras del cliente para lograr su satisfacción y mejorar sus expectativas.
- **Liderazgo:** la participación de los líderes es indispensable para establecer un sistema de calidad. Ellos son los responsables de crear un entorno que propicie productos de calidad y, a su vez, deben contribuir al crecimiento y mejoramiento de la misma.
- **Participación del Personal:** el involucramiento del personal es crucial para el logro de los objetivos, y deben participar desde el director de la organización hasta el último de los empleados, sin importar su actividad específica.
- **Enfoque basado en los procesos:** la mejor manera de producir los resultados deseados es plantear las actividades como un proceso, ya que de esta manera se alcanzan resultados más eficientes. Al visualizar las actividades como procesos se hace más sencilla la identificación de oportunidades de mejora.
- **Enfoque del Sistema para la Gestión:** la organización es un conjunto de procesos que se relacionan como un sistema constituido por actividades, personal y recursos que deben administrarse como un solo proceso con el objetivo fundamental de la mejora continua de la organización y la satisfacción del cliente.
- **Mejora Continua:** la mejora continua del desempeño global de la empresa debe ser el objetivo permanente de la misma; para lograr el desempeño óptimo de las actividades realizadas.

- **Enfoque basado en hechos para la toma de decisión:** se basa en un análisis de los datos y la información disponible.
- **Relaciones recíprocas beneficiosas con el proveedor:** las organizaciones tienen proveedores y éstos mantienen una íntima relación con el éxito de la organización, deben tratarse como socios y reconocer la necesidad de ambos de la existencia y participación del otro, de tal manera que la relación sea beneficiosa para ambos.

La Figura N° 1 ilustra el sistema de gestión de la calidad basado en procesos descrito en la familia de Normas ISO 9000:2008. Se puede observar un esquema general y bastante sencillo que resume cómo se debe orientar el desarrollo de un proyecto, siguiendo los principios antes mencionados.

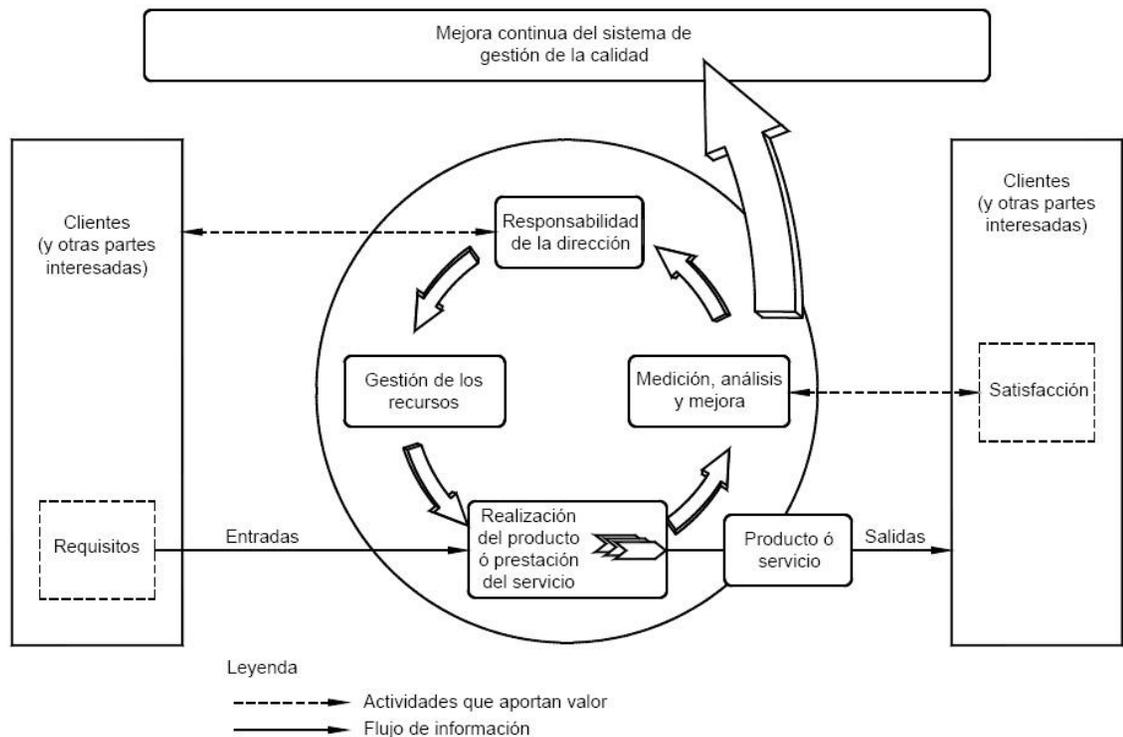


Figura N° 1: Modelo de un Sistema de Gestión de Calidad basado en procesos (ISO 9001:2008)

Fuente: [6]

2.4. Seguimiento.

La organización debe aplicar métodos apropiados para el seguimiento. Siendo estos métodos el modo de controlar los procesos del sistema de gestión de calidad, y quienes definen las características de cada uno de estos procesos (misión, entradas, salidas, recursos necesarios) [7].

2.5. Indicador.

Se puede definir como el instrumento de monitorización y observación de un sistema, construido a partir de la evaluación y relación de variables del sistema. La medición de estas variables y su posterior comparación con los valores meta establecidos permitirá determinar el logro de los objetivos del sistema de gestión de la calidad, así como su tendencia de evolución [8].

2.6. Mapa de proceso.

Según algunos actores tenemos las siguientes definiciones:

"Es una metodología que permite la representación gráfica de un proceso, puntualizando la secuencia de tareas que se ejecutan. Es una herramienta de calidad orientada al mejoramiento de los procesos existentes para optimizarlos" [9]

"Es una representación gráfica, de la forma en que los empleados de una organización hacen su trabajo, mostrando el orden y las interacciones de las tareas que componen cada uno de los procesos" [10]

"Es una actividad que se realiza para establecer una representación gráfica (mapa) de un proceso determinado de trabajo en una empresa, mostrando las entradas, salidas y secuencia de actividades" [11]

La construcción grafica de un mapa de procesos permite:

- Examinar y comprender todos los procesos de la empresa.
- Determinar las relaciones entre los procesos y las áreas de trabajo.
- Analizar a la empresa en base a las actividades y tareas que se efectúan.

2.7. Herramientas de desarrollo de software.

Para el desarrollo de la aplicación se consideró la reutilización de componentes ya desarrollados, para facilitar el desarrollo y enfocarse a la implementación de mejoras o nuevas funcionalidades más complejas en los casos que fueron necesarias. El uso de manejadores de contenidos, bibliotecas u otras herramientas, permiten tener tiempos de respuesta de desarrollo más cortos.

2.7.1. Manejador de contenido.

Es una herramienta que permite administrar el contenido de páginas web de forma automática, es decir, nos permite publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para los visitantes de la página web [12].

2.7.2. Modulo.

Es un software que agrupa un conjunto de subprogramas y estructuras de datos. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas (o varias, en algún caso). [13]

2.7.3. Herramientas de desarrollo Web.

Se tienen dos grupos muy importantes de tecnologías, las que están del lado del cliente y las que están del lado del servidor, esto nos facilita el desarrollo de la aplicación.

- **HTML** (*HyperText Markup Language*, Lenguaje de marcas de hipertexto).

Es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web.

HTML es un lenguaje de marcación de elementos para la creación de documentos hipertexto, fácil de aprender, lo que permite que cualquier persona, aunque no tenga conocimientos previos, pueda realizar la tarea de crear una página. [14]

- **CSS** (*Cascading Style Sheets*, Hojas de Estilo Cascada).

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto obliga a crear documentos HTML/XHTML bien definidos. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otras. [15]

- **JavaScript.**

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Es decir, que incorpora efectos como texto que aparece y desaparece,

animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. [16]

- **AJAX** (*Asynchronous JavaScript + XML*, JavaScript asíncrono + XML).

Es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones [17].

- **PHP** (*Hipertext Preprocessor*).

Es un lenguaje de programación del lado del servidor, rápido y con una gran biblioteca de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. [18]

2.7.4. Drupal.

Es una herramienta que permite administrar el contenido de páginas web de forma automática, es decir, permite publicar, editar, borrar, otorgar permisos de acceso o establecer los módulos visibles para los visitantes de la página web.

Drupal es la mejor opción en proyectos con una complejidad alta, y aunque tenga una curva de aprendizaje mayor a los otros manejadores de contenido, ofrece una gran flexibilidad y escalabilidad en los proyectos que desempeñemos.

2.7.5. JSON.

JSON (*JavaScript Object Notation* - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Es de simple interpretación y generación. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras. [19]

En cuanto a seguridad, la interpretación puede realizarse con las técnicas de evaluación de código; los cuales convierten un objeto a su representación JSON y viceversa respectivamente. Antes de llegar a la parte de evaluación es necesario validar la cadena de texto (*string*) por razones de seguridad, evitando que el JSON en cuestión altere el contexto en el cual se ejecuta. [20]

2.7.6. SVG.

SVG (*Scalable Vector Graphic*, gráficos basados en vectores escalables) es un formato gráfico basado en XML para crear archivos vectoriales en 2D, con un lenguaje de marcado por medio de etiquetas.

Además, a los objetos gráficos les podemos aplicar transformaciones (traslaciones, escalamiento, etc.) y animaciones.

Algunas ventajas de este formato son:

- No pierde calidad al ampliar la imagen.
- Permite aplicar escalamientos.
- Se muestra de forma progresiva, eliminando la espera de que se descargue todo el archivo.
- Se puede reducir su tamaño a través del formato de compresión GZIP (*GNU ZIP*).
- Permite aplicar estilos CSS.
- Permite modificaciones con JavaScript.
- Al ser XML, es indexable por los buscadores.

Esta sección está basada en [21].

2.7.7. Librería D3.js.

D3.js (*Data-Driven Documents*), es una biblioteca de Javascript utilizada principalmente para crear gráficos y visualizaciones interactivas en un sitio web. Esta herramienta de visualización de datos hace uso principalmente de: gráficos vectoriales (SVG), Javascript, HTML5 y CSS3. D3.js permite usar grandes cantidades de datos en distintos formatos: JSON, CSV, GeoJSON, etc.

Esta biblioteca permite manipular documentos basados en datos usando estándares abiertos de la web y los navegadores pueden crear visualizaciones complejas sin depender de un software propietario. Siendo una biblioteca basada en estándares web, se adapta a

cualquiera de los navegadores modernos, permitiendo mantener su aspecto y diseño (Cross-Browsers). [22]

A través de HTML, CSS y SVG permite generar gráficas y animaciones. D3 permite manipular cualquier parte del DOM.

Por su flexibilidad, es utilizado por los desarrolladores web en sus proyectos. Por lo tanto, existen módulos de WordPress y Drupal. [23]

2.7.8. Otras herramientas.

Otra herramienta muy utilizada son los sistemas de control de versiones o SCM (*SourceCode Management*). Estos son programas para versionar y controlar el código fuente. Ejemplo de ellos son: SVN (Subversion) y GIT.

CAPITULO III.

3. MARCO METODOLOGICO.

El desarrollo de software no es una tarea fácil. Para poder abordar la complejidad del desarrollo de software, fueron creadas las metodologías. Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen planteando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

Además, las metodologías ingenieriles han estado presentes durante mucho tiempo. La crítica más frecuente a estas metodologías es que son burocráticas. Hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda. [24]

En la actualidad existen numerosas propuestas metodológicas que inciden, según sea el caso, en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales centradas específicamente en el control del proceso.

Por lineamientos de la organización y características como que en este proyecto los requisitos están bien definidos, es recomendable aplicar una metodología como MeRinde que evita forzar al cliente a tomar la mayoría de las decisiones al principio, y luego el coste de cambio de una decisión tomada pueda llegar a ser muy elevado. Evitando generar problemas como los que a continuación mencionamos:

- Retrasos en la planificación: llegada la fecha de entregar el software éste no está disponible.
- Sistemas deteriorados: el software se ha creado pero después de un par de años el coste de su mantenimiento es tan complicado que definitivamente se abandona su producción.
- Tasa de defectos: el software se pone en producción pero los defectos son tantos que nadie lo usa.

- Requisitos mal comprendidos: el software no resuelve los requisitos planificados inicialmente.
- Cambios de negocio: el problema que resolvía nuestro software ha cambiado y nuestro software no se ha adaptado.
- Falsa riqueza: el software hace muchas cosas técnicamente muy interesantes y divertidas, pero no resuelven el problema de nuestro cliente, ni hace que éste gane más dinero.
- Cambios de personal: después de unos años de trabajo los programadores comienzan a odiar el proyecto y lo abandonan.

Como respuesta a los problemas aplicando metodologías tradicionales surgieron otras metodologías que tratan de adaptarse a la realidad del desarrollo de software. [24]

3.1. Metodología de la Red Nacional de Integración y Desarrollo de Software Libre (MeRinde)

Esta sección está basada en la recopilación de información de [25].

MeRinde es un proyecto que propone un estándar abierto para el proceso de desarrollo de software orientado a planes que se estructura en dos dimensiones o ejes que serán explicados en detalle en la siguiente sección.

La Metodología MeRinde surge de la combinación y adaptación de modelos y metodologías ampliamente utilizadas para el desarrollo de software y la reingeniería de procesos del negocio. Esta metodología está fuertemente fundamentada en los requisitos del Centro Nacional de Tecnología de Información (CNTI) y en varias metodologías como el Proceso Unificado (UP) especialmente.

Pretende entre sus principales objetivos apoyar a las comunidades de desarrollo de software sin restricciones de licencia en sus proyectos, suministrando las herramientas necesarias para que estos cumplan con un proceso de desarrollo y documentación de sus sistemas.

MeRinde es concebida para abarcar el desarrollo completo de sistemas de software de diversa complejidad y magnitud, por lo cual su estructura responde a diversos desarrollos y deberá adaptarse y dimensionarse en cada momento de acuerdo a las características particulares de cada proyecto. Dada la adaptabilidad que puede sufrir la metodología, ésta puede llegarse a aplicar bajo un enfoque ágil. Así mismo, esta permite producir y mantener una biblioteca de plantillas reutilizables para ingeniería de software. Está basada en componentes, lo cual quiere decir que el sistema en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Con el proceso de desarrollo y con las plantillas de esta metodología se busca a su vez estimular la transferencia del conocimiento entre las comunidades desarrolladoras de software sin restricciones de licencia, con lo cual no solo se pretende que sea compartido los códigos de los sistemas sino que también se compartan la documentación como guía de referencia para mejoras por terceros al sistema o para que sirva como modelo a otras comunidades para el desarrollo de sus propios sistemas.

El ciclo de vida de un proyecto de software desarrollado por el CNTI se inspira en UP, motivo por el cual se descompone en el tiempo en cuatro fases secuenciales.

3.1.1. Estructura.

MeRinde propone un estándar abierto para el proceso de desarrollo de software orientado a planes, que como se muestra en la Figura N° 2, se estructura en dos (2) dimensiones o ejes (eje vertical y eje horizontal) donde el eje horizontal representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso e indica las características del ciclo de vida del proceso, expresado en términos de fases, iteraciones e hitos. Y el eje vertical que representa los aspectos estáticos del proceso y describe el proceso en términos de componentes, disciplinas, actividades, artefactos y roles. Cada barra representa una iteración por fase para un proyecto, adicionalmente el modelo enfatiza que durante cada iteración se recorren todas las disciplinas pero con diferente esfuerzo.

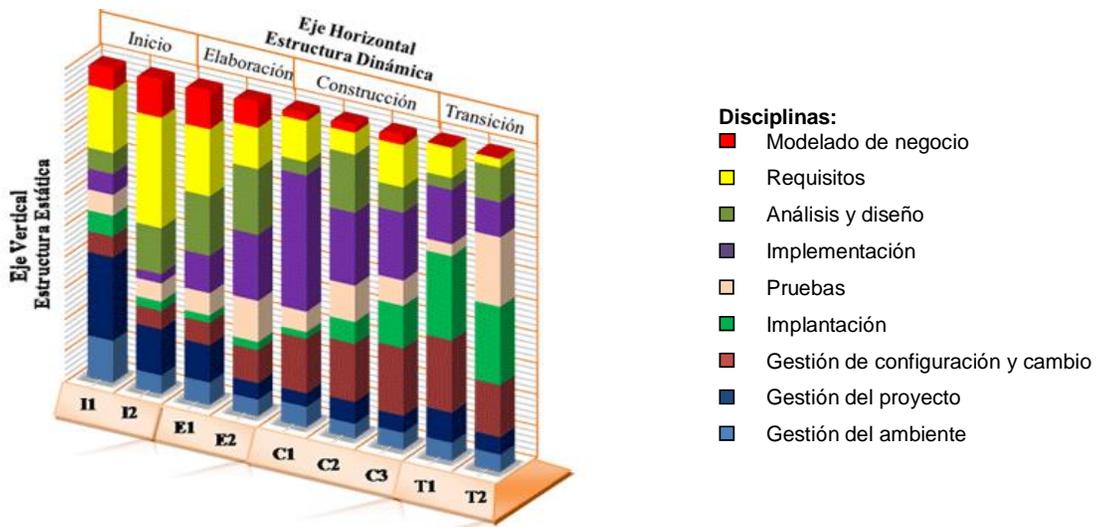


Figura N°2: Estructura MeRinde.

Fuente: [25]

A continuación serán explicadas las disciplinas que serán abarcadas en este proyecto como apoyo del equipo de trabajo de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT).

3.1.1.1. Requisitos.

El objetivo principal de esta disciplina es establecer las funciones que se quiere que satisfaga el sistema a construir. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. Para obtener los requisitos se deben aplicar prácticas de elicitación a los involucrados en el proyecto, anotar y validar todas sus solicitudes.

Los requisitos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario. Los requisitos pueden ser divididos en dos grupos: los requisitos funcionales, los cuales describen las funciones que el software va a ejecutar; por ejemplo, ajustarse a un formato de texto o modular una señal, y los requisitos no funcionales, los cuales especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus funciones específicas; por ejemplo, seguridad, eficiencia y usabilidad.

3.1.1.2. Análisis y diseño.

El objetivo principal de la disciplina análisis y diseño es transformar los requisitos a una especificación que describa cómo implementar el sistema. El análisis fundamentalmente consiste en ver qué hace el sistema de software a desarrollar, por tal motivo se enfoca en los requisitos funcionales. Por otro lado el diseño es un refinamiento que abarca los requisitos no funcionales, por lo cual se centra en cómo el sistema cumple sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes.

3.1.1.3. Implementación.

El objetivo principal de la disciplina implementación es convertir los elementos del diseño en elementos de implementación, estos elementos son: códigos fuentes, ejecutables y binarios. Otra parte de esta disciplina son las pruebas unitarias, las cuales se limitan a los componentes de software implementados. De esta disciplina se obtiene un sistema ejecutable estable, constituido de los resultados producidos por los programadores individuales.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

3.1.1.4. Pruebas.

El objetivo principal objetivo de esta disciplina es el de evaluar la calidad del producto que se está desarrollando a través de las diferentes fases por las cuales este pasa, mediante la aplicación de pruebas concretas para validar que las suposiciones hechas en el diseño y los requisitos se estén cumpliendo satisfactoriamente, esto quiere decir que se verifica que el producto funcione como se diseñó y que los requisitos son satisfechos cabalmente. Esta disciplina brinda soporte para encontrar y documentar (y solucionar) defectos en la calidad del sistema a las otras disciplinas. Esta disciplina debe estar presente en todo el ciclo de vida del desarrollo del sistema para ir refinándolo y no al final del mismo.

3.1.2. Fases.

El ciclo de vida de un proyecto de software desarrollado por el CNTI se inspira en UP, motivo por el cual se descompone en el tiempo en cuatro fases secuenciales, que son:

- Inicio
- Elaboración
- Construcción
- Transición

- **Inicio.**

Su propósito general es establecer los objetivos para el ciclo de vida del producto. Durante esta fase se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y casos de uso.

El hito en esta fase finaliza con el establecimiento del alcance del proyecto y su viabilidad.

- **Elaboración.**

Su objetivo general es plantear la arquitectura para el ciclo de vida del producto. Se construye un modelo de la arquitectura, que se desarrolla en iteraciones sucesivas

hasta obtener el producto final, este prototipo debe contener los casos de uso críticos que fueron identificados en la fase de inicio. En esta fase se realiza la captura de la mayor parte de los requisitos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

El hito en la fase de elaboración finaliza con la obtención de una línea base de la arquitectura del sistema, la captura de la mayoría de los requisitos y la reducción de los riesgos importantes así como permitir la escalabilidad del equipo del proyecto durante la fase de construcción.

- **Construcción.**

El objetivo general de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. En esta fase todas las características, componentes, y requisitos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión aceptable del producto comúnmente llamada versión beta. Se hace énfasis en controlar las operaciones realizadas, administrando los recursos eficientemente, de tal forma que se optimicen los costos, los calendarios y la calidad.

El hito en esta fase culmina con el desarrollo del sistema con calidad de producción y la preparación para la entrega al equipo de transición. Toda la funcionalidad debe haber sido implementada y las pruebas para el estado beta de la aplicación completadas. Si el proyecto no cumple con estos criterios de cierre, entonces la transición deberá posponerse una iteración.

- **Transición.**

Tiene como objetivo general entregar el producto funcional en manos de los usuarios finales una vez realizadas las pruebas de aceptación por un grupo especial de usuarios, para lo que se requerirá desarrollar nuevas versiones actualizadas del producto, entrenar a los usuarios en el manejo del sistema, completar la documentación, y en general tareas relacionadas con la configuración, instalación y usabilidad del producto.

3.1.3. Roles

La cantidad de roles a utilizar para el desarrollo de un proyecto de software a realizar con esta metodología depende de la magnitud del proyecto. Mientras más grande y complejo sea el proyecto requerirá de una mayor cantidad de participantes para su elaboración y más roles especializados. Otro factor importante a considerar para elegir los roles a participar en el proyecto es el tiempo asignado al desarrollo del proyecto.

Esta metodología más que proponer una serie de roles estáticos para un proyecto establece que se pueden utilizar los roles que se consideren necesarios para realizar el proyecto según las características y el tiempo requerido por éste.

Una de las razones principales de la adopción de esta metodología para el desarrollo de software consiste en la definición de las tareas que serán llevadas a cabo por los individuos que participan en un proyecto. En MeRinde un rol define las responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Éste se encarga de la realización de tareas, las cuales generan artefactos.

La metodología del CNTI propone ocho (8) roles básicos que deben tomarse en cuenta para la elaboración de software como son:

- Analista de Calidad
- Analista de Producto
- Arquitecto de Software
- Desarrollador
- Involucrado
- Líder del Proyecto
- Mentor
- Probador

Es importante destacar que todos los proyectos pequeños o grandes que utilicen esta metodología en su proceso de desarrollo, deben considerar estos ocho (8) roles entre los roles definidos para el proyecto. Esta metodología señala una serie de roles recomendados pero cabe destacar que un rol puede ser desempeñado por varias personas y una persona

puede representar varios roles, es por ello que esta metodología brinda la oportunidad de incorporar un número indefinido de personas a los proyectos de desarrollo.

CAPITULO IV.

4. DESARROLLO DE LA APLICACIÓN.

4.1. Plataforma utilizada en el desarrollo.

Debido a que este proyecto es desarrollado para PDVSA-INTEVEP S.A. siguiendo los lineamientos de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT), se utilizaron las siguientes herramientas de software: Drupal como CMS base, PostgreSQL como sistema gestor de base de datos, D3.js como biblioteca javascript para la generación de gráficos, SVN para el control de versiones, apache como servidor y *sublime text* como editor, para facilitar el desarrollo del código fuente. Dado que las plataformas actuales de PDVSA INTEVEP S.A. son Windows XP y Linux, estas herramientas se ejecutaron bajo estos sistemas operativos.

4.2. Plataforma necesaria para la ejecución de la aplicación.

El usuario deberá contar con las siguientes versiones de navegadores para utilizar la aplicación:

- Internet Explorer 6.x o superior.
- Firefox 5.x o superior.
- Opera 12 o superior.
- Safari 5.x o superior.
- Google Chrome.

4.3. Metodología.

En la sección 3, se explica la metodología MeRinde, como una metodología iterativa e incremental, que divide el desarrollo en cuatro (4) fases. Este trabajo, por ser apoyo al

equipo de trabajo de la gerencia AIT, abarcará tres (3) fases: Inicio, elaboración y construcción.

En la misma idea, dado que el aporte de este trabajo se enfoca en el desarrollo de dos de los módulos del Sistema de Información de Gestión de la Calidad (SIGC), este proyecto no abarca todas las disciplinas de la metodología MeRinde mostradas en la Figura N° 2, sólo se enfoca en las disciplinas: requisitos, análisis y diseño, implementación, y pruebas.

4.3.1. Fase - Inicio.

En esta fase se establecieron los límites del proyecto, así como el establecimiento de los casos de uso del sistema. Además, se muestra una arquitectura candidata para los escenarios principales.

4.3.1.1. 1ra Iteración.

En esta iteración se establecen los requisitos funcionales y no funcionales. De igual manera se muestra la arquitectura de la organización.

- **Requisitos.**

A continuación se describen los requisitos no funcionales y funcionales de los módulos abarcados en este trabajo para el Sistema de Información de Gestión de la Calidad (SIGC); estos fueron obtenidos a través del trabajo previo ya referenciado [1]:

- **Requisitos no funcionales**

Para el desarrollo del SIGC, durante el levantamiento de los requisitos, se plantearon los siguientes requisitos no funcionales:

- Usabilidad: el software será de fácil uso, ya que las interfaces estarán diseñadas pensando en las necesidades del usuario. El software contará con

iconos alusivos y textos de ayuda que faciliten la interacción con el programa, por lo que el tiempo de aprendizaje será mínimo.

- **Confiabilidad:** el software será confiable, ya que, asegura un nivel de funcionamiento adecuado bajo condiciones normales, y se recupera manteniendo su nivel de funcionamiento en caso de errores.
- **Mantenimiento y actualización:** el sistema será construido de forma modular, con alta cohesión y bajo acoplamiento entre módulos, por lo que el mantenimiento y actualización será una tarea sencilla ya que los cambios realizados en un módulo no afectaran a otro.

- **Requisitos funcionales**

Para el desarrollo del proyecto, se extrajeron los requisitos funcionales que serán abarcados:

- **Iniciar sesión:** el usuario se identifica en el sistema con su indicador y contraseña para tener acceso a las funciones correspondientes.
- **Gestionar procesos:** permite definir la estructura de la organización, los roles y las responsabilidades (permisos) inherentes a cada rol.
- **Crear proceso:** permite definir los procesos de la organización. Establecer el tipo de proceso.
- **Configurar proceso:** permite establecer los datos relacionados con los procesos, niveles de aprobación, POA y su sistema de gestión de calidad asociado.
- **Modificar proceso:** permite la modificación de la definición de los procesos de la organización dejando establecido el estado anterior.
- **Proceso obsoleto:** permite definir procesos activo como obsoleto, al igual que todo su sistema de gestión.
- **Seguimiento:** permite aplicar métodos apropiados para el seguimiento de los procesos del sistema de gestión de la calidad.
- **Gestionar objetivos:** permite asociar los diferentes niveles del POA a cada nivel según corresponda en la estructura organizacional.
- **Crear objetivo:** permite definir los diferentes objetivos.
- **Editar objetivo:** permite la modificación de la definición de los objetivos.

- Gestionar indicadores: permite conocer los diferentes indicadores.
- Crear indicadores: permite definir los diferentes indicadores.
- Editar indicadores: permite la modificación de la definición de los indicadores.
- Administración: permite la administración del sistema. Permite la visualización de los registro de modificación de información así como de obsolescencia, y la modificación de listas del sistema sin necesidad de un especialista en bases de datos.

La descripción de los casos de uso así como los diferentes niveles, se encuentran en el documento Especificación de Requisitos de Software (ERS) [1]. Especificando que el ERS controla la evolución del sistema durante todo el ciclo de desarrollo del proyecto, cuando las nuevas características son añadidas o modificadas, son aclarados dentro del artefacto ERS.

A través del ERS se filtraron los casos de usos abarcados por este proyecto los cuales se pueden observar en las Figura N° 3 y N° 4.

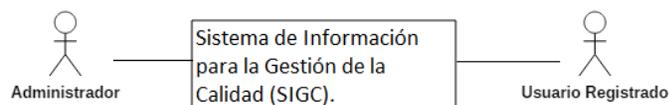


Figura N° 3: Caso de uso Nivel 0.



Figura N° 4: Caso de uso Nivel 1.

- **Análisis y diseño.**

Para el ciclo de vida del desarrollo del producto se planteó el esquema mostrado en la Figura N° 5, la cual muestra una arquitectura basada en la filosofía Cliente/Servidor de la organización, que consiste en 3 capas: interfaz o cliente quien es el que realiza las peticiones; objetos de negocios como encargada de ofrecer servicios y responder las peticiones solicitada; y objetos de acceso a bases de datos que permiten el almacenamiento de grandes cantidades de datos. Adicional, un corta fuegos (*firewall*) diseñado para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

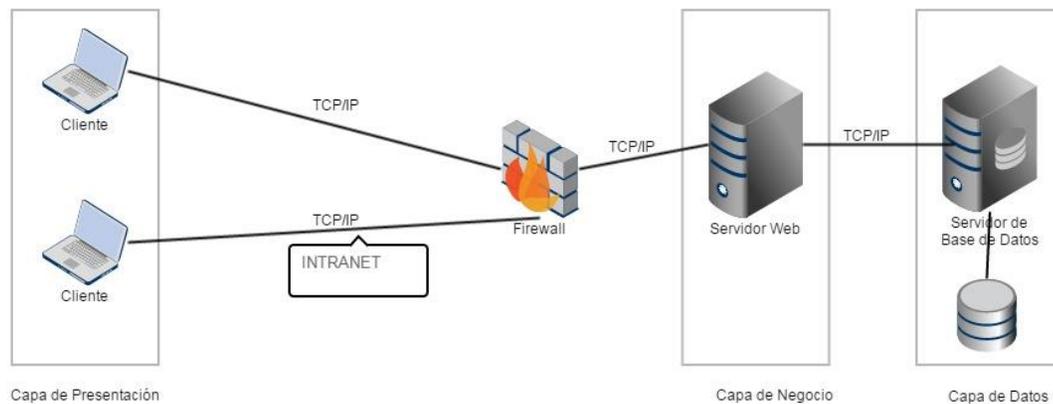


Figura N° 5: Arquitectura Cliente/Servidor.

De esta manera, estableciendo los límites del proyecto a través de los requisitos funcionales, no funcionales y los casos de usos críticos del proyecto, así también la arquitectura de desarrollo para los escenarios principales, la gerencia determinó el proyecto viable para desarrollo.

4.3.2. Fase - Elaboración.

En esta fase se definió, validó y estableció la arquitectura.

4.3.2.1. 1ra Iteración.

En esta iteración se definió la arquitectura del sistema.

- **Análisis y diseño.**

Como se puede observar en la figura N° 6, se establece la arquitectura sistema de información de gestión de la calidad, la cual consiste en tres (3) capas, que separan los datos de la aplicación en módulos transaccionales, módulos de inteligencia de negocios y el módulo de gestión de documentos. Esto adaptado a la arquitectura de hardware y software presente en la organización.

Los módulos abarcados en este proyecto, llamados *Gestión de Procesos y Seguimiento*, (enmarcados en rojo en la figura n° 6) pertenecen a la capa de módulos transaccionales. Los módulos restantes fueron contemplados para un desarrollo posterior, enfocando el módulo de *Control de Documentos* al servicio de la organización enfocado a esto, *Gestión de Carpetas, Revisión por la Dirección, Análisis y Mejora, Centro de Mensajes y Programa de Reuniones* son enfocados al módulo inteligencia de negocios en la plataforma de Pentaho. Los módulos *Recursos Humanos, Control de Interesados y Equipos* no están definidos en su totalidad por falta de información.

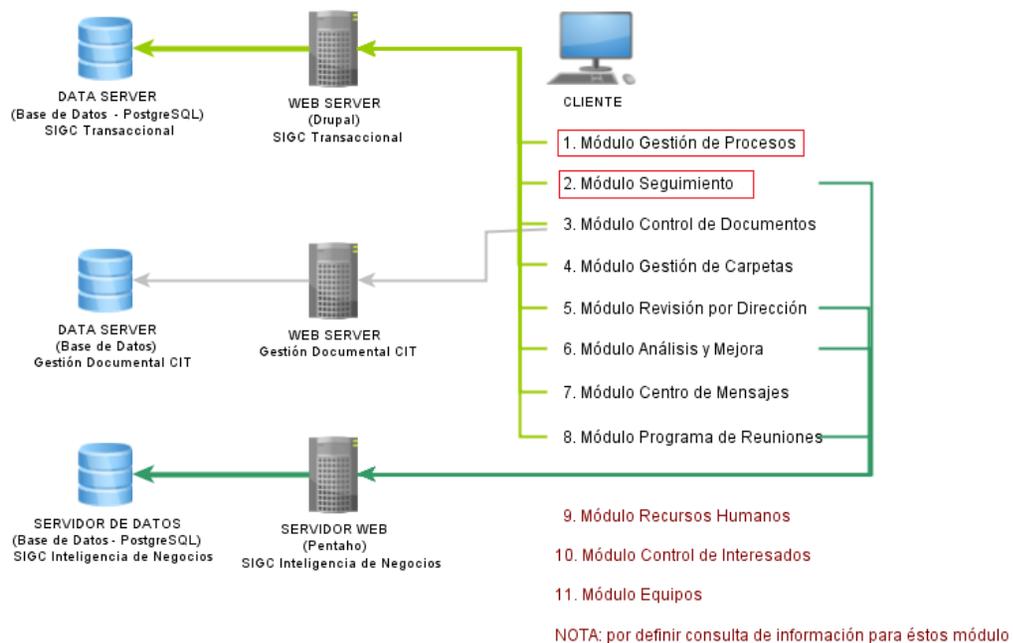


Figura N° 6: Arquitectura SIGC.

En la figura N° 7 se puede observar el diagrama de despliegue en el que estará basado la aplicación. Se tiene la interfaz de usuario a través del navegador web, también un servidor de aplicaciones con Apache, el cual tiene instalado Drupal como manejador de contenidos para el desarrollo de la aplicación, un servidor de base de datos con postgresql, un servidor SMTP para correo (que no es utilizado en lo que abarca este proyecto pero debió ser considerado) y el directorio activo de la organización para ser consultado para el inicio de sesión de los usuarios.

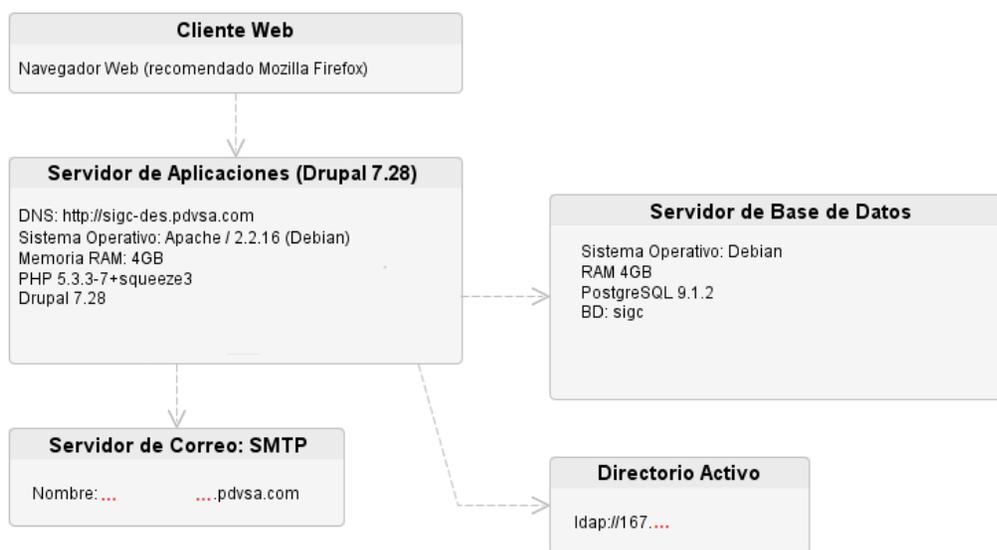


Figura N° 7: Diagrama de despliegue SIGC.

4.3.2.2. 2da Iteración.

Usualmente no se utilizan las plantillas por defecto del CMS, además que no se adaptan a los lineamientos de diseño de la organización, por lo cual se hace una particularización basado en temas.

- **Análisis y diseño.**

En la figura N° 8 se observa la definición de la estructura de los objetos que componen los módulos transaccionales a desarrollar. Las estructuras principales son las organizaciones, suborganizaciones, procesos y subprocesos (resaltados en recuadros de

color rojo), las cuales son base para estructuras que van asociadas a ellas como las entradas y salidas de los procesos, riesgos, retos, tipos, información documentada, registros, oportunidades). De igual manera, asociadas a estas estructuras bases, tenemos las estructuras que conforman el POA de la organización, estos son los objetivos estratégicos, objetivos tácticos estratégicos y tácticos operacionales, metas tácticas estratégicas y tácticas operacionales, indicadores, objetivos de la calidad, metas de la calidad e indicadores de la calidad.

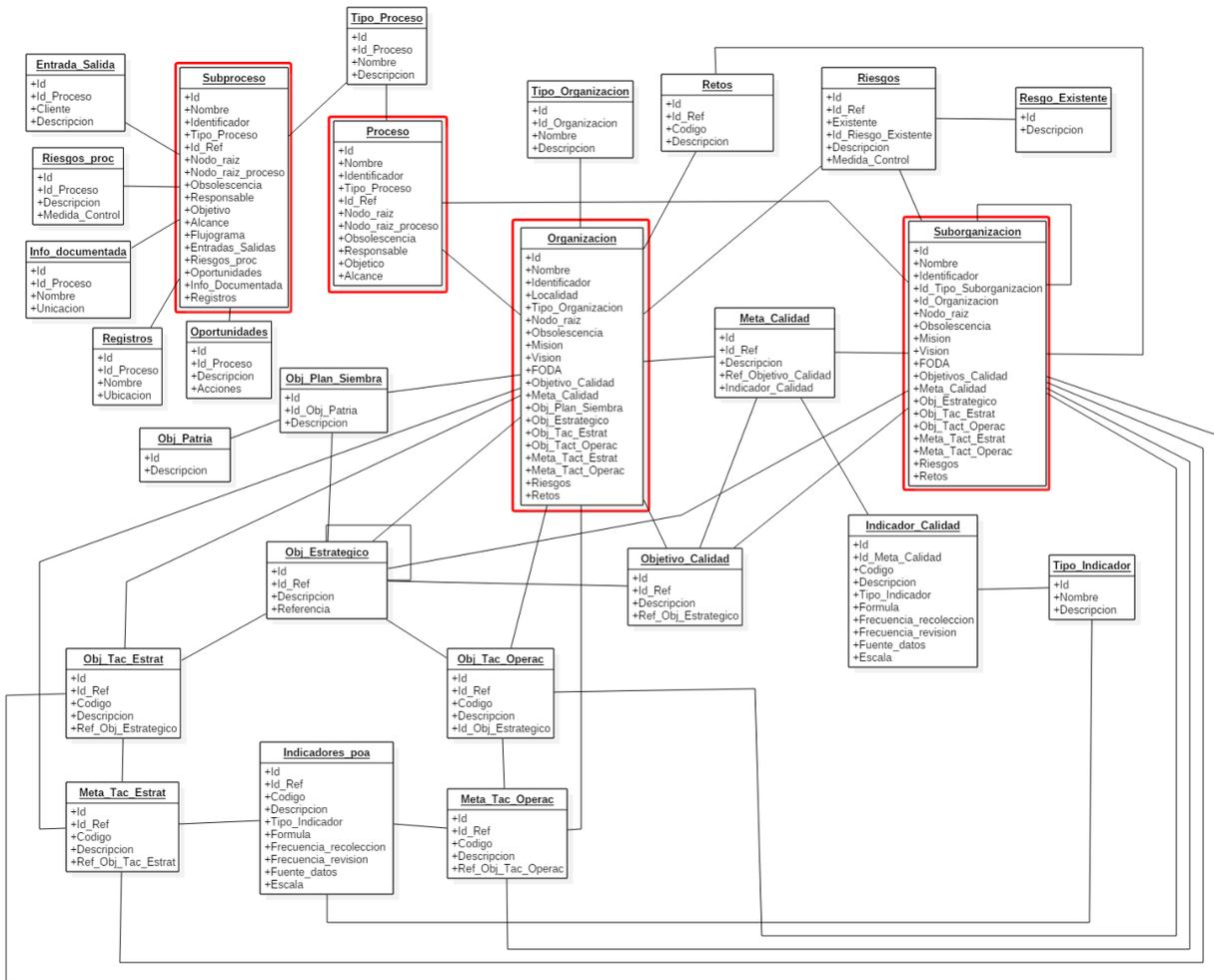


Figura N° 8: Diagrama de objetos.

Es importante hacer notar que la estructura reflejada en el diagrama de objetos de la figura 8, no se corresponde necesariamente con el modelo de datos, esto debido a que Drupal proporciona una capa de abstracción de base de datos que hace de intermediario

entre el sistema y la base de datos en sí misma. En este modelo no se realizan llamadas directas a la base de datos, sino que se hacen a través de funciones genéricas definidas por la capa de abstracción de la base de datos. Sin embargo, los contenidos creados en la aplicación si estarán estructurados de esta manera.

- **Implementación.**

En la primera etapa de esta disciplina, se estableció el diseño de las interfaces de usuario del sistema, en ocasiones fue necesario realizar modificaciones al diseño preestablecido inicialmente.

Como base fue utilizado el tema *simplecorp* en el CMS, que provee la plantilla de interfaz que visualizará el usuario en toda la navegación, además provee características adaptadas a los lineamientos de la organización, permitiendo ahorrar costos de desarrollo. En el manejador de contenido (Drupal) se configuraron algunas características que permiten ocultar o mostrar algunos elementos que por defecto vienen con el tema, así como establecer el color del tema y sus botones. Como parte de la configuración, se integra el logo de la organización en la plantilla para cumplir con los lineamientos.

Las personalizaciones realizadas, las cuales se explicaron en el párrafo anterior, no fueron suficientes debido a que ciertos atributos no eran personalizables desde la configuración del manejador de contenidos. Es por esto que se tuvo que modificar directamente la hoja de estilos del tema como se puede observar en la figura N° 9, donde se muestra parcialmente las modificaciones realizadas a la plantilla básica, como fondos, colores y ajustes de tamaño.

```
body { background: #FFF; color: #898989; fo
-webkit-text-size-adjust: 100%; }

body.custom-background {background:#f2f2f2;

iframe {border: none;}

.front #first-time {
display: none;
}
```

Figura N° 9: Algunos cambios en hoja de estilos.

Este diseño se personalizó enmarcado en los lineamientos de diseño de software establecidos por la organización.

Por lineamientos de la organización no debe ser posible la creación de nuevas cuentas ni la solicitud de nuevas contraseñas a través del sistema para el inicio de sesión de los usuarios. Esto debido a que el sistema debe conectarse al directorio activo de la organización, por lo que los datos de inicio de sesión son consultados directamente en él y son comprobados con los datos de los usuarios al momento del inicio de sesión.

Después de aplicar todos los cambios en la hoja de estilos, y las configuraciones necesarias a través del manejador de contenidos, se puede observar en la figura N° 10, el diseño de la plantilla obtenida, adaptado a lo solicitado por la gerencia, cumpliendo con los colores, fuentes y dimensiones. Obteniendo un entorno base, que enmarca toda la aplicación.

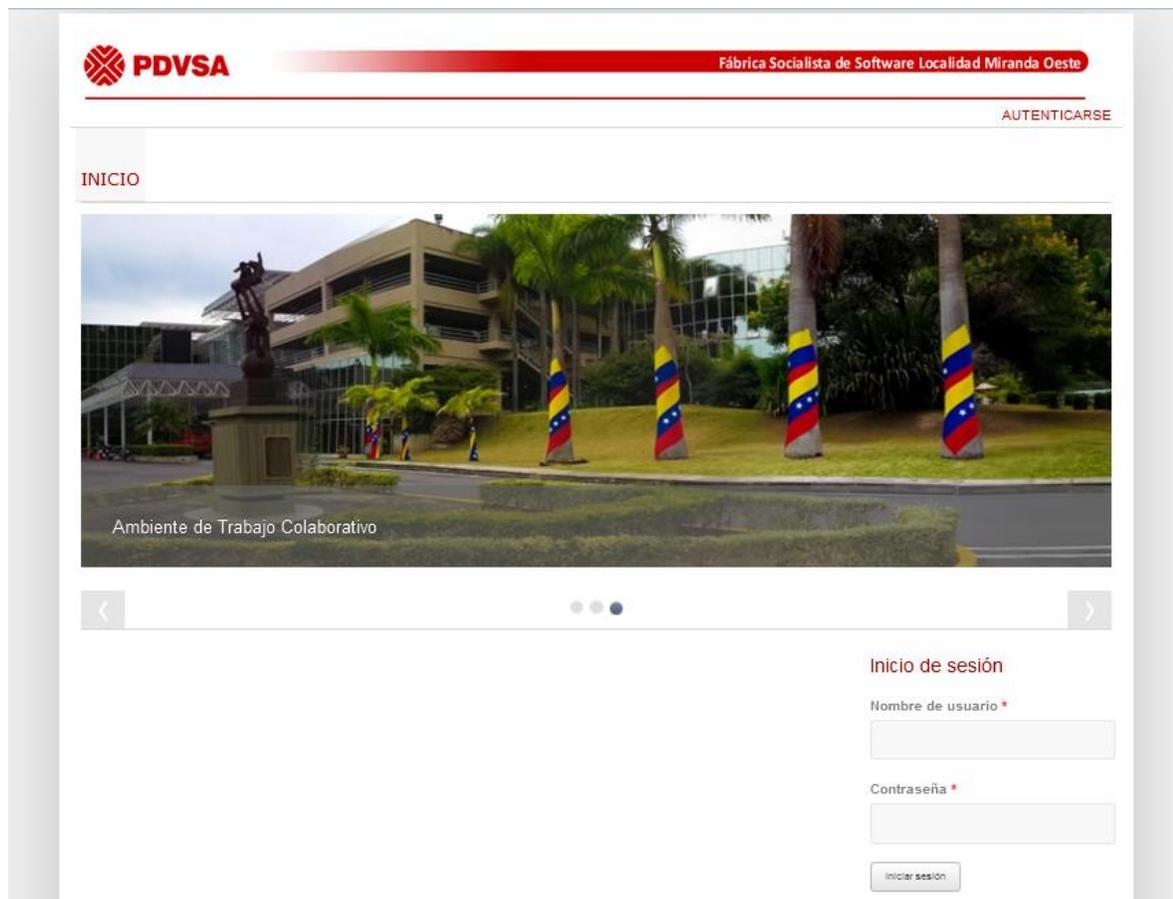


Figura N° 10: Página principal.

- **Pruebas.**

Mediante estas pruebas se revisaron los aspectos gráficos del sitio, determinando su correcto despliegue en las páginas. Dentro de los elementos más importantes a ser verificados, siguiendo los lineamientos de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste, se incluyen los siguientes:

- **Consistencia.**

Cada una de las páginas del sitio tiene elementos consistentes, con el fin de ofrecer al usuario una experiencia similar en cualquier área del sitio web. Entre algunos aspectos tenemos que los menús deben aparecer siempre en el mismo lugar; que los listados deben estar diseñados de similar manera en todo el sitio y que los colores y formas de uso de las interfaces deben ser similares a lo largo de las páginas.

- **Compatibilidad de navegadores.**

Aunque la codificación en los lenguajes soportados por los navegadores puede apegarse a los estándares, no todos muestran de la misma manera los sitios web. Dado esto, fue necesario revisar el sitio en diferentes navegadores, contemplando los más usados: Google Chrome, Microsoft Internet Explorer, Opera y Mozilla Firefox en las versiones mencionadas anteriormente en la sección 4.2. Se revisó en este caso el despliegue de todos los elementos que se muestran en la pantalla, para asegurar de que aparecen en las posiciones que se les han asignado en el diseño.

4.3.3. Fase – Construcción.

En esta fase, se minimizaron los costos de desarrollo, reutilizando código y módulos existentes y desarrollando nuevos en el caso que fueran necesarios, así evitando el tener que rehacer el trabajo o incluso desecharlo.

4.3.3.1. 1ra Iteración.

Se establecieron las estructuras bases que permitirán la creación de la estructura organizacional, así como la separación principal según el tipo de organización para facilitar al usuario la introducción al sistema.

- **Implementación.**

En esta primera iteración se construye la estructura definida en el diagrama de objetos, que permite establecer los tipos de organización existentes, divididos en 4 grupos: Gerencias corporativas, Filial, Empresas mixtas, y Negocios. Permitiendo dividir las organizaciones según su tipo, reduciendo carga al usuario al momento de localizar la organización que desea. A partir de estas, se integra la sección que permite ingresar a la navegación de la estructura organizacional llamada "Gestión de procesos" al menú principal de la aplicación.

Se crearon los botones correspondientes, así como una plantilla para la inserción de éstos. Para luego incorporar los botones referenciales a cada uno de los tipos de organización para que al ser accedidos, se ejecute la consulta que se observa en la figura N° 11 y liste las organizaciones asociadas. Dicha consulta extrae el identificador del contenido al igual que su título, de la tabla resultado de la unión de todos los contenidos que tienen cualquier "tipo de organización" asociado y su posterior filtrado según su estado y el tipo de organización que quiere visualizar el usuario.

```
SELECT node_taxonomy_index.title, node_taxonomy_index.nid
FROM taxonomy_term_data
LEFT JOIN taxonomy_index ON taxonomy_term_data.tid = taxonomy_index.tid
LEFT JOIN node node_taxonomy_index ON taxonomy_index.nid = node_taxonomy_index.nid
WHERE (( (node_taxonomy_index.status = '1') AND (node_taxonomy_index.type IN ('tipo_organizaciones')) ))
```

Figura N° 11: Consulta SQL.

En esta iteración se crean las estructuras establecidas en el diagrama de objetos, que permiten formar la estructura organizacional. Entre ellos tenemos: organizaciones, suborganizaciones, procesos y subprocesos.

Las organizaciones serán creadas solo por el administrador de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT), y debido a que el caso de estudio para este proyecto es la Gerencia Departamental del Sistema de Gestión de Calidad perteneciente a PDVSA-INTEVEP S.A., la organización base es "INTEVEP S.A., S.A.". Las estructuras restantes (suborganizaciones, procesos y subprocessos) serán creados por los usuarios con permisos correspondientes.

En la figura N° 12 se puede observar cómo se comporta la estructura organizacional. El primer nivel, que son las organizaciones, éstas pueden contener de 1 a N suborganizaciones y/o de 1 a N procesos. En el nivel siguiente tenemos las suborganizaciones, que a su vez pueden contener de 1 a N niveles de suborganizaciones; estas suborganizaciones pueden tener de 1 a N procesos. El siguiente nivel está formado por procesos, que pueden contener de 1 a N subprocessos. Y por último tenemos los subprocessos, que pueden contener de 1 a N niveles de subprocessos.

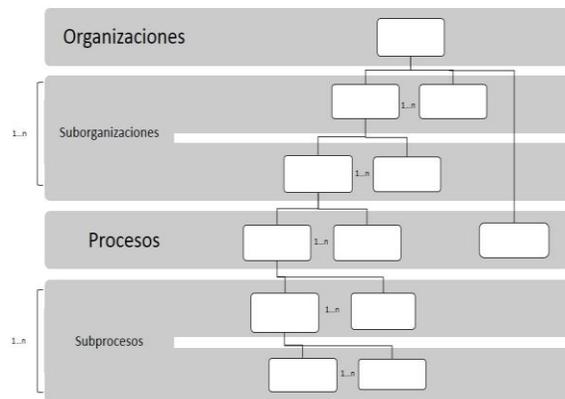


Figura N° 12: Estructura organizacional.

Las referencias serán creadas automáticamente por el sistema a través de los formularios, y solo podrán ser modificadas y en algunos casos visualizadas solo por el administrador de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT).

4.3.3.2. 2da Iteración.

- **Implementación.**

En esta iteración se crean las estructuras y sus respectivos campos, que permiten asociar la información de cada contenido y su sistema de gestión de la calidad a cada nivel de la estructura organizacional, incluyendo la estructura del Plan Operativo Anual (POA) de PDVSA-INTEVEP S.A.

Se organizan estas estructuras en los formularios de cada contenido. Cada formulario está dividido en tres (3) secciones: Datos, Información general y Mapa de proceso. En la sección "datos" se pueden observar los atributos básicos y obligatorios del contenido. En la sección "información general" se encuentran la estructura del Plan Operativo Anual así como otra información asociada a la calidad. Por último, en la sección "Mapa de proceso" se permite la creación y asociación de la estructura organizacional.

Dentro de cada una de estas secciones, los campos fueron agrupados por pestañas expansibles, de esta manera se reduce la sobrecarga visual a los usuarios, se permite tener un orden y escalabilidad en la carga de la información. En la figura N° 13 se puede observar el formulario de uno de los contenidos.

The image shows a vertical list of eight expandable tabs, each with a right-pointing chevron and blue text. The tabs are: 'OBJ. PLAN SIEMBRA', 'OBJETIVOS ESTRATÉGICOS', 'OBJ. TÁCTICOS ESTRATÉGICOS', 'OBJ. TÁCTICOS OPERACIONALES', 'METAS TÁCTICAS ESTRATÉGICAS', 'METAS TÁCTICAS OPERACIONALES', 'RETOS', and 'RIESGOS'. Below the tabs are two buttons: 'Anterior' on the left and 'Siguiete' on the right. The entire form is enclosed in a thin black border.

Figura N° 13: Vista de un formulario.

Para mayor información sobre los formularios y la carga de información, es recomendable consultar el manual de usuario [26].

4.3.3.3. 3ra Iteración.

Se establecieron mecanismos para permitir al usuario la asociación de contenido ya existente a través de los reportes en los campos de referencia.

- **Implementación.**

Para los campos de referencias de contenido surgió la necesidad de generar reportes en el sistema, pero debido a que estos campos son mostrados a través de una ventana emergente, se pierde el identificador del contenido que se encuentra visualizando el usuario, por lo que a través de las funciones proporcionadas por el API de Drupal no se pudo tener acceso a él.

Por esto se desarrolló un acceso a la cabecera HTTP denominada HTTP Referrer, que identifica la dirección de la página web que creó el vínculo con el recurso que está siendo solicitado (en este caso, la ventana emergente). En la figura N° 14 se observa un ejemplo de este acceso, permitiendo de esta manera obtener el URI (Identificador de Recursos Uniforme) que es un identificador más completo que el URL y luego, éste es parseado para obtener el identificador del contenido que está visualizando el usuario y, a través de las funciones proporcionadas por la API de Drupal, tener acceso a él. En el ejemplo observado en la figura, si el contenido es una suborganización, se retorna el identificador de la organización principal a la cual pertenece, pero sí el contenido es una organización, retorna su propio identificador.

```
$url= $_SERVER['HTTP_REFERER'];
$url= strstr($url,"node/");
$result = explode("/",$url);
$nid = (int)preg_replace("/[^0-9]/", "", $result[1]);
$node = node_load($nid);
if ($node->type == "organizacionesn") {
return($node->field_ref_organizacion['und'][0]['target_id']);
}
else{
return ($nid);
}
```

Figura N° 14: Código utilizando HTTP Referrer.

Este procedimiento es utilizado en todas las vistas que necesitan hacer referencias a contenido ya existente. En algunos casos, las referencias son a contenidos del mismo que está visualizando el usuario, o al que este le está haciendo referencia el mismo.

También se desarrolla el diseño de la ventana emergente presente en los campos de referencia. En la figura N° 15 se observa el estado original de la ventana y su posterior diseño, el cual, fue modificado a través de la modificación de la hoja de estilos CSS como se puede observar en la figura N° 16. Aquí, se establecen algunas características como los bordes, márgenes, alineaciones, entre otros. Cada ventana de referencia al ser desplegada, muestra una vista de una consulta configurada y asignada anteriormente.



Figura N° 15: Ventana emergente.

```
div.ctools-modal-content {  
  background: #fff;  
  color: #000;  
  padding: 0;  
  margin: 2px;  
  border: 1px solid #000;  
  width: 600px;  
  text-align: left;  
  border-radius: 15px 15px 15px 15px;
```

Figura N° 16: Hoja de estilos de la ventana emergente.

- **Pruebas.**

Para las pruebas funcionales, se realizó la evaluación de cada uno de los campos para verificar la correcta visualización de las vistas, como se puede observar en la figura N° 17.

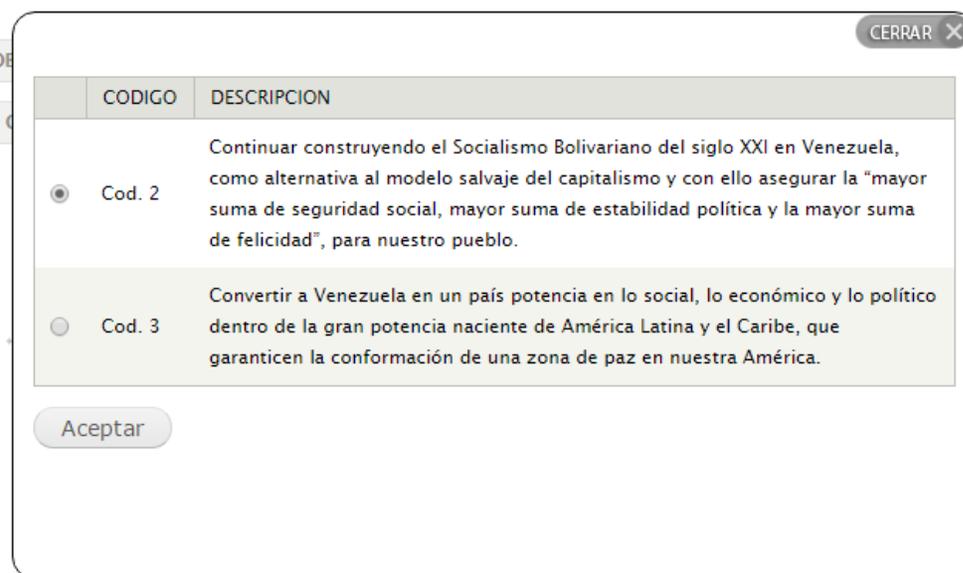


Figura N° 17: Visualización de las vistas.

4.3.3.4. 4ta Iteración.

Se establecieron mecanismos para lograr la asignación de referencias al momento de la creación de los contenidos.

- **Implementación.**

En este punto, dada la necesidad de crear la estructura organizacional desde el formulario del contenido que está visualizando el usuario, se agregó al manejador de contenidos un campo de creación que permite realizar dicha función. Dicho campo no permitía establecer las referencias por defecto, lo cual era necesario para evitar que el usuario realizara esta acción y para que a través de esta referencia se puedan consultar otros datos.

Dicho campo de creación permite crear de forma dinámica entidades relacionadas al contenido, ayudando a mejorar la experiencia del usuario. Pero adicional a esto, se requiere que el contenido creado (suborganizaciones, procesos y subprocesos) haga referencia a su creador. Para poder llevar a cabo esto es necesario que el formulario del contenido creador se almacene en caché, a continuación, se genera el nuevo formulario de creación, el cual permite la creación de contenido nuevo. Una vez que el nuevo

formulario se completa y se guarda, el sistema retorna al usuario al formulario original a través del identificador de cache. Para lograr las referencias, al identificador de cache donde se almacena el formulario, que es enviado a través de la URL, se le concatena el identificador del contenido que lo está creando (el cual es obtenido del HTTP Referrer), como se puede observar en la figura N° 18.

```

$url= $_SERVER['HTTP_REFERER'];
$url= strstr($url,"node/");
$result = explode("/", $url);
$url = preg_replace("/[^0-9]/", "", $result[1]);

$item['href'] = $item['href'] . ".$cache_id". "/"$url";
$content[$key] = $item;
}

```

Figura N° 18: Obtención y asignación de identificador.

De igual forma, siguiendo con el planteamiento anterior, se definieron acciones que son ejecutadas condicionalmente con base a los acontecimientos que ocurren, en este caso, luego de que un contenido nuevo es guardado. Esto puede ser observado en la figura N° 19 donde, según sea el tipo de contenido que se está creando, serán asignados y almacenados en base de datos las referencias correspondientes según el nivel de la estructura organizacional. Por ejemplo, como se observa en la figura N° 19 el caso de que el contenido nuevo sea una suborganización, se asignarán dos referencias, una referencia a su contenido creador y otra referencia a la organización a la que pertenece (la cual es obtenida a través de una función que consulta a que organización pertenece su contenido creador).

```

if(arg(2) == "procesos"){
$node->field_referencia['und'][0]['target_id'] = arg(4);
$node->field_nodo_raiz_ref_p['und'][0]['target_id'] = arg(4);
$node->field_nodo_raiz_ref['und'][0]['target_id'] = NodeRoot(arg(4));
}

if(arg(2) == "organizacionesn"){
$node->field_ref_organizacion['und'][0]['target_id']=arg(4);
$node->field_nodo_raiz_ref['und'][0]['target_id'] = NodeRoot(arg(4));
}

if(arg(2) == "procesosn"){
$node->field_referencia['und'][0]['target_id'] = arg(4);
$node->field_nodo_raiz_ref_p['und'][0]['target_id'] = NodeRootP(arg(4));
$node->field_nodo_raiz_ref['und'][0]['target_id'] = NodeRoot(arg(4));
}

function NodeRoot($node){
$result = node_load($node);
return $result->field_nodo_raiz_ref['und'][0]['target_id'];
}

function NodeRootP($node){
$result = node_load($node);
return $result->field_nodo_raiz_ref_p['und'][0]['target_id'];
}

```

Figura N° 19: Asignación de referencias.

- **Pruebas.**

Se comprueba la creación de referencias al crear la estructura organizacional en sus diferentes niveles. Verificando los identificadores de cache y el pase de los identificadores de los contenidos por la URL y su correcta asignación a los contenidos creados. Aquí también se observó que el campo mencionado permite la edición de la estructura una vez está creada, esto podría causar problema de nodos no referenciados, y fue abarcado en la siguiente iteración.

4.3.3.5. 5ta Iteración.

En esta iteración, se implementó el bloqueo del campo que permite crear la estructura organizacional y la creación de un módulo de Drupal que garantiza la integridad referencial de los términos utilizados en el sistema.

- **Implementación.**

Una vez que el campo de creación (el cual genero la inconsistencia en la iteración anterior) se encuentre lleno, el usuario no puede modificarlo, evitando contenidos sin referencias. Para ello se implementó en el módulo anterior de Drupal, que para cada uno de los campos, se desactive para evitar ser editado como se ve en la figura N° 20.

```
$( ':input[type=text]' ).each(function(){  
    if(this.name.indexOf("field_estruct_orga") >= 0 && this.value != ""){  
        this.disabled = true;  
    }  
})
```

Figura N° 20: Visualización de las vistas.

También, se crean las estructuras de términos que serán utilizadas por los contenidos, establecidas en el diagrama de objetos. Estas estructuras permitirán asociar términos a los contenidos, como por ejemplo los tipos de organización, tipos de procesos, frecuencia de revisión de los indicadores, entre otros.

Estas estructuras, deben evitar ser eliminados si están siendo utilizados por algún contenido, por lo que se desarrolló el modulo con el API de Drupal que evita dicha acción. Este módulo fue llamado "term_used".

Así, no se le permitirá al usuario eliminar un término que este siendo utilizado por algún contenido, manteniendo así la integridad referencial. En cambio, se mostrará un mensaje de aviso con la lista de los contenidos que están utilizando el término que intenta ser eliminado; además de una lista con enlaces a cada uno de los contenidos que lo está utilizando.

En la figura N° 21 se muestra parte del desarrollo donde se toma el formulario de eliminación de términos y se personaliza. Se realiza una consulta de los contenidos que estén utilizando el término, si el resultado arroja resultados, se muestra el aviso junto con la lista de contenidos. También se oculta el botón de confirmar la eliminación.

```

if (count($result) > 0) {
  $markup = t('Este término esta siendo usado y no puede ser borrado:') . '<ul>';

  foreach($result as $nid) {
    $node = node_load($nid);
    if (!$node)
      continue;
    $markup .= '<li>' . l($node->title, 'node/' . $node->nid, array('attributes' => array('target'=>'_blank'))) . '</li>';
  }

  if (count($result) >= $limit)
    $markup .= '<li>' . t("... solo @limit resultados han sido mostrados.", array('@limit' => $limit)) . '</li>';
  $markup .= '</ul>';

  $form['description']['#markup'] = $markup;
  $form['actions']['submit']['#access'] = FALSE;
}

```

Figura N° 21: Creación de formulario de eliminación de términos.

Estas estructuras también deben ser accedidas y alimentadas a través de una sección de administradores. Para se creó una vista que realiza una consulta que muestra todas las estructuras de términos existentes y crece dinámicamente, es decir, si se crea una nueva estructura de términos, automáticamente aparecerá en el módulo de administración para ser alimentada por el usuario con el rol correspondiente.

• Pruebas.

Para comprobar el funcionamiento del módulo de Drupal creado, se crearon términos y se asignaron a contenidos, así también términos que no estén en uso. Se logró la eliminación de los términos que no estén en uso y se observó el correcto funcionamiento del módulo en los términos que están siendo utilizados, como se observa en la figura N° 22.

Inicio » [Estratégicos](#)

Este término esta siendo usado y no puede ser borrado:

- [Proceso Refinacion](#)

[Cancelar](#)

Figura N° 22: Comprobación de advertencia cuando un término está siendo utilizado.

4.3.3.6. 6ta Iteración.

Se desarrolla el módulo de Drupal que permite la colocación en un estado de obsolescencia los contenidos.

- **Implementación.**

Con la finalidad de tener un histórico de las estructuras organizacionales y sus mapas de proceso, uno de los requisitos del usuario fue evitar la eliminación de contenido, para ello se estableció una función que permite a cada uno de los contenidos de las estructuras organizacionales que permite establecerlo como obsoleto. Permitiendo colocar un estado de obsolescencia a los contenidos, evitando la eliminación y dejando registro de que existió el contenido y puede ser visitado. El marcado se realiza a través de un campo en el formulario de cada contenido que puede ser visualizado sólo por el administrador.

La función se realiza a través de un módulo llamado "obsolescencia" que, al ser marcado el campo de obsolescencia en el formulario, se arroja a través de JavaScript una advertencia al usuario que indica que será modificada toda la estructura organizacional. En la figura N° 23 se observa la comprobación de cambiado de estado del campo, para luego realizar una consulta recursiva que arroja los identificadores de toda la estructura organizacional, comenzando en el nodo que el usuario está visualizando. Así las filas del resultado de la consulta, contienen los identificadores de cada contenido de la estructura organizacional, cargando cada uno de estos contenidos y actualizando en base de datos a través de las funciones proporcionadas por el API de Drupal, el estado a obsoleto, o según sea el caso, se le remueve el estado de obsolescencia, tanto al contenido como a toda su estructura.

```

if($form['field_obsolescencia']['und']['#value'] != $node->field_obsolescencia[LANGUAGE_NONE][0]['value']){
  $nodo = $node->nid;
  $result=db_query("WITH RECURSIVE path(id) AS (
    SELECT NP.NID
    FROM drupal_node NP
    WHERE NP.NID = $nodo
    UNION
    SELECT NP.NID
    FROM
    path as parentpath, drupal_node NP
    FULL JOIN (
      SELECT REF.ENTITY_ID, REF.FIELD_REFERENCIA_TARGET_ID AS REFERENCIA
      FROM drupal_field_data_field_referencia REF
      UNION
      SELECT REFE.ENTITY_ID, REFE.FIELD_REF_ORGANIZACION_TARGET_ID
      FROM drupal_field_data_field_ref_organizacion REFE
    ) REF
    ON NP.NID = REF.ENTITY_ID
    WHERE REF.REFERENCIA = parentpath.id) SELECT * FROM path;
  ");
  if($result->rowCount()){
    while( $fila = $result->fetchAssoc() ) {
      $nid=$fila['id'];
      $nodo= node_load($nid);
      if($form['field_obsolescencia']['und']['#value'] == 1){
        $nodo->field_obsolescencia[LANGUAGE_NONE][0]['value'] = 1;
        $nodo->status = 0;
      } else{

```

Figura N° 23: Funcionalidad de obsolescencia.

- **Pruebas.**

Para comprobar el funcionamiento del módulo, se crearon estructuras organizacionales y se establecieron como obsoletas en diferentes niveles de la estructura. Se comprobó que el modulo creado funciona correctamente y que puede establecer el contenido y su estructura como obsoleto al igual que quitar este estado.

4.3.3.7. 7ma Iteración.

En esta iteración, se establecieron los diseños de visualización de cada contenido, su organización y presentación para aliviar la carga visual al usuario y mantener uniformidad entre como son cargados los datos en el formulario y como son mostrados.

- **Implementación.**

Se modificó la estructura u organización con que se muestran los contenidos, implementándose una disposición que se adapta a los requisitos de diseño. Aunque cada entidad de la estructura organizacional está conformada por diferentes campos y relaciones, se realizó un diseño que fuera consistente para los contenidos. Se creó una

plantilla como se puede observar en la figura N° 24, la cual puede ser editada y configurada en un futuro.

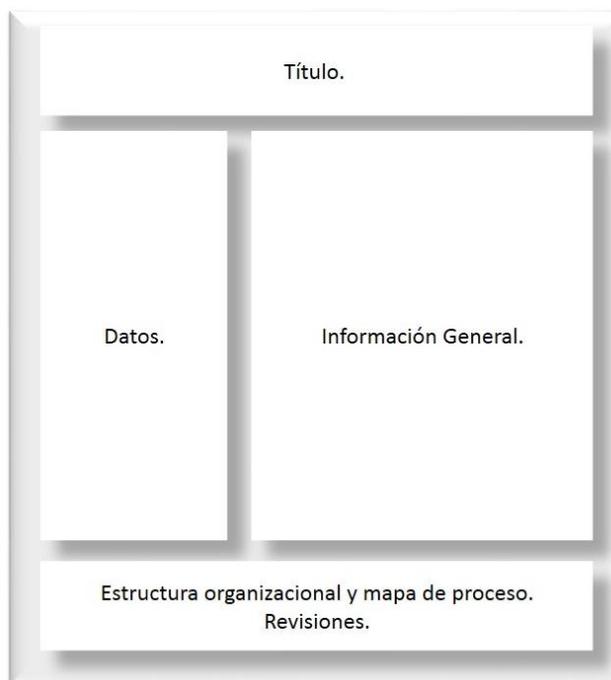


Figura N° 24: Modelo de plantilla de diseño para los contenidos.

Este diseño fue elaborado para lograr consistencia en la visualización de los diferentes tipos de contenidos, logrando que el usuario durante la navegación se encuentre con la misma estructura en los contenidos, reduciendo así el esfuerzo del usuario y moldeando la percepción que el usuario tiene del software.

A través de este diseño es posible indicar y distribuir los campos y estructuras pertenecientes a cada sección. Es importante también, que la distribución de las secciones corresponda a la distribución encontrada en los formularios de los contenidos.

Una vez establecida la plantilla, se asignaron los campos y estructuras a cada sección de la plantilla según correspondía. En la figura N° 25 se pueden observar parte de la plantilla con algunas secciones adecuadas al tipo de contenido "organizaciones". Cada recuadro representa un campo o una consulta que genera una vista, como por ejemplo el título, identificador, localidad, misión, visión, entre otros. Esto fue realizado para cada tipo de contenido de la estructura organizacional.

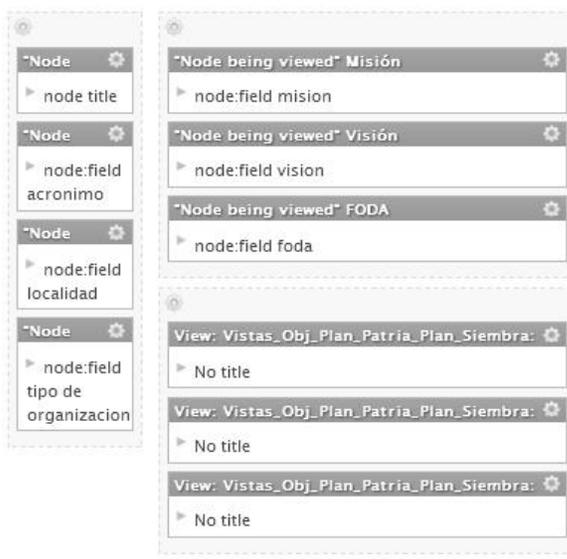


Figura N° 25: Plantilla de diseño para los contenidos establecida en la aplicación.

También se configura para que, según el caso, se muestren los campos por sección y por pestañas ya sean de manera horizontal o vertical.

Una vez establecida la plantilla con sus respectivos campos, el diseño muestra los elementos que hayan sido completados en el formulario como se puede observar en la figura N° 26.

Organización 1

Nombre de la organización
Organización 1

Identificador
Organización 1

Localidad
Región 1

Tipo de organización
Filial

Misión
Visión

Objetivos del Plan Patria
Objetivos del Plan Siembra Petrolera

Objetivos Estratégicos

Objetivos Tácticos Estratégicos
Metas Tácticas Estratégicas

Código	Descripción
1	Objetivo táctico estratégico 1 ⓘ
2	Objetivo táctico estratégico 2 ⓘ
3	Objetivo táctico estratégico 3 ⓘ

Objetivos Tácticos Operacionales
Metas Tácticas Operacionales

Política de la calidad
Objetivos de la calidad
Metas de la calidad

Retos
Riesgos

▶ Estructura organizacional

Figura N° 26: Visualización de plantilla de diseño.

- **Pruebas.**

Se realizaron pruebas para verificar la correcta visualización de los datos en los diferentes tipos de contenido. Así como el llenado de los campos en el formulario para su posterior aparición en la visualización del contenido. Se comprobó la correcta asignación de las plantillas a los diferentes tipos de contenido.

Sin embargo se observó que las pestañas horizontales, al visualizar el contenido, se desplegaban de manera automática y no permitían contraerse; similar las pestañas horizontales que no permitían contraerse una vez eran desplegadas. Lo cual fue abarcado en la siguiente iteración.

4.3.3.8. 8va Iteración.

Se establecieron comportamientos de algunos objetos de la interfaz de usuario.

- **Implementación.**

En esta iteración se implementan las funcionalidades que solucionen las inconformidades de las pestañas descritas en la iteración anterior.

En la figura N° 27 se observa la modificación de la propiedad *collapsible* que permite el despliegue y contracción de las pestañas, debido a que se encontraba deshabilitada.

```
he:false,cookie:null,collapsible:true,disable:nu
(/s/g, "_").replace(/[\w\u00c0-\uFFFF-]/g, "")|
.opacity&&f.opacity&&g[0].style.removeAttribute(
(i&&i!=="#"){d.data(f,"href.tabs",i);d.data(f,"l
ault ui-corner-top");this.panels.addClass("ui-ta
cted=-1;c.selected=c.selected>=0&&this.anchors[
```

Figura N° 27: Establecimiento de propiedad a pestañas.

También se observó que estaba siendo asignada por defecto la propiedad que despliega las pestañas, por lo cual se eliminó la propiedad como se puede observar en la figura N° 28, de manera que las pestañas no se desplegaran por defecto.

```
. _sanitizeSelector(1));else if(1&&i!=="#"){d.data(f,"href.tabs",i);d.data(f,"load.tab
this.lis.addClass("ui-state-default ui-corner-top");this.panels.addClass("ui-tabs-pane
else_if(c.selected===null)c.selected=-1;c.selected=c.selected>=0&&this.anchors[c.select
)) removeClass("ui-tabs-hide"); this.lis.eq(c.selected).addClass("ui-state-active");a.e
c.selected,c.cookie,b=0;for(var j;j=this.lis[b];b++)d(j).inArray(b,c.disabled):-1&&
action(){n("hover",d(this))};this.anchors.bind("focus.tabs",function(){k("focus",d(thi
("ui-tabs-selected ui-state-active");f.removeClass("ui-tabs-hide");a._trigger("show",n
("ui-tabs-hide")),l=a.element.find(a._sanitizeSelector(g.hash));if(f.hasClass("ui-tabs
```

Figura N° 28: Eliminación de propiedades.

También, debido a la necesidad de que el usuario durante toda la navegación y visualización del contenido este informado de en qué contenido está posicionado, a través de un módulo existente, se estableció un bloque que muestre el título del contenido a través del filtro PHP proporcionado por el sistema como se pueden ver en las figura N°

29, el cual se desplaza con la ventana del navegador. También se le asignaron propiedades CSS personalizadas para adaptarlo al diseño.

```
<h1 style="background-color: white; padding-top: 8px; padding-bottom: 8px; margin-top: 0px; border-bottom: 0.5px solid #F0F0F0">
  <?php print drupal_get_title(); ?>
</h1>
```

Figura N° 29: Establecimiento de bloque.

En la figura N° 30 se puede observar parte del CSS de las pestañas. La eliminación de la propiedad "position: relative" a las clases correspondientes de las pestañas horizontales y verticales, permitiendo que el resto de elementos continúen en su posición ignorando el bloque que se desplaza.

```
.ui-tabs { padding: .2em; zoom: 1; }
.ui-tabs .ui-tabs-nav { margin: 0; padding: .2em .2em 0; }
.ui-tabs .ui-tabs-nav li { list-style: none; float: left; top: 1px; margin: 0 .2em 0 0; }
.ui-tabs .ui-tabs-nav li a { float: left; padding: .5em 1em; text-decoration: none; }
```

Figura N° 30: Ajustes CSS.

También, se agregó la funcionalidad que permite la redimensión del ancho del bloque al redimensionar el navegador en el archivo JavaScript como se puede ver en la figura N° 31.

```
else {
  blockInfo.padding_bottom = 0;
}

blockInfo.width = $(block).width();

block.css({
  width: blockInfo.width + 'px',
  left: blockInfo.original_offset.left + 'px'
```

Figura N° 31: Redimensión de bloque.

- **Pruebas.**

En esta iteración se revisó el funcionamiento de las pestañas horizontales y verticales, pero en este caso comprobando las funcionalidades agregadas, las cuales, funcionaron como se esperaba permitiendo el despliegue y contracción de las mismas.

También se visitaron diferentes contenidos para verificar el comportamiento del bloque flotante con el título de contenido, el cual se desplazaba correctamente. Adicional, el bloque se ajusta al navegador al redimensionarlo.

Se verificaron las funcionalidades en diferentes navegadores, y su correcto funcionamiento como se puede observar en la figura N° 32.

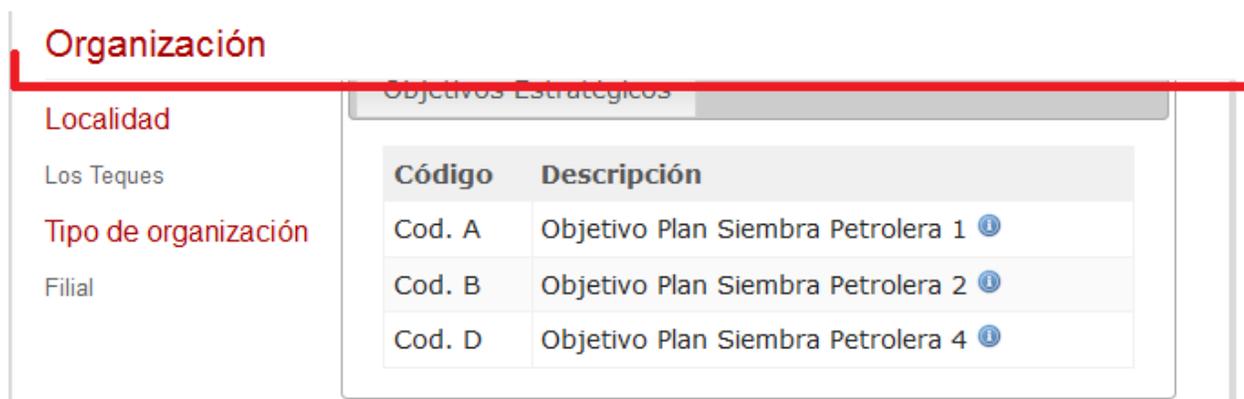


Figura N° 32: Bloque contenedor de título de contenido.

4.3.3.9. 9na Iteración.

Creación del módulo de Drupal para la carga de la librería d3.js y la creación del mapa de procesos dinámico.

- **Implementación.**

En esta iteración se comienza la implementación del mapa de proceso que permitirá al usuario navegar a través de él. Para ello se utilizó la biblioteca d3.js y su módulo para el

manejador de contenido. El cual fue adaptado para hacer un módulo instalable de la biblioteca y otro del mapa de proceso como se puede ver en la figura N° 33.

ACTIVADO	NOMBRE	VERSIÓN	DESCRIPCIÓN
<input checked="" type="checkbox"/>	D3	7.x-1.x-dev	An API for creating visualizations from D3 libraries Necesita: Libraries (activado), System (activado) Necesitado por: mapa (activado)
<input checked="" type="checkbox"/>	mapa	7.x-1.x-dev	API D3 Contiene Mapa de Procesos Necesita: D3 (activado), Libraries (activado), System (activado)

Figura N° 33: Instalación de módulos D3.

Una vez instalada la biblioteca en el manejador de contenido, se realizó la creación de un bloque, que es el espacio que contiene y permite la visualización del mapa de proceso. Este bloque se estableció dentro de una pestaña vertical, la cual fue asignada al espacio correspondiente de la plantilla de diseño de cada contenido como se observa en la figura N° 34.

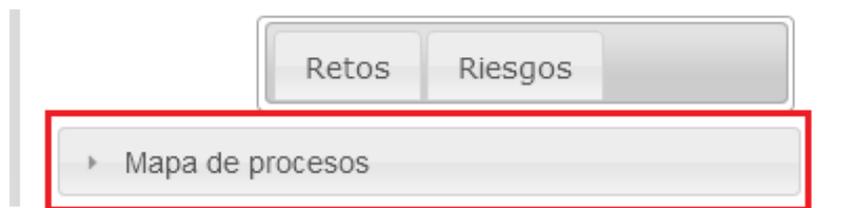


Figura N° 34: Visualización de pestaña contenedora del bloque.

Como puede observarse en la figura N° 35 el archivo que contiene el HTML del mapa de proceso, el cual contiene una etiqueta de imagen para el logo de PDVSA, un contenedor para el mapa de procesos, una etiqueta de imagen para la leyenda del mapa de procesos, un botón para exportar el mapa de procesos a una imagen descargable y un canvas oculto que nos servirá como estructura auxiliar para la exportación.

```
<div><img id="logo" ></img></div>
<div <?php print $attributes ?> class="<?php print implode(' ', $classes_array); ?>"></div>
<div style="vertical-align:middle; text-align:center; padding-top: 25px;"><img id="leyenda" ></img></div>
<p><button id="png">Exportar mapa de proceso</button></p>
<canvas id="canvas" style="display: none"></canvas>
```

Figura N° 35: Estructura del mapa de procesos.

En la figura N° 36 se observa la realización de un llamado desde el modulo a la biblioteca que contiene el mapa de proceso.

```
function d3_default_libraries() {
  return array(
    'd3.mapa',
  );
}
```

Figura N° 36: Llamada a la biblioteca constructora del mapa de procesos.

Pero, antes de comenzar con la construcción del grafico del mapa de procesos, se requirió extraer los datos de la base de datos. Se necesitó de cada contenido la siguiente información: identificador, nombre, acrónimo, tipo de proceso, referencia y prioridad.

Para la extracción de los datos, se estableció una consulta que devuelve el árbol de los procesos a través del formato JSON. Para lograr esto, se instaló en la base de datos la extensión plv8 de postgresql, que permite ejecutar funciones escritas en JavaScript en el servidor. De esta manera se creó la función recursiva en JavaScript que es ejecutada en el servidor de base de datos llamada "json_child_tree", que puede ser observada en la figura N° 37, la cual recibe como parámetro el identificador y el estatus (debido a que si se encuentra en estado de obsolescencia debe mostrar la estructura correspondiente a su estado) del contenido que se encuentra visualizando el usuario. Y retornando una cadena con formato JSON con toda la estructura organizacional.

```
CREATE OR REPLACE FUNCTION json_child_tree(parent integer, status integer)
  RETURNS json AS
$BODY$
var rows = plv8.execute("WITH RECURSIVE path(name, path, id, parent_id,
var all = {},
  out = [],
  top,r,i;
for(i=0; i<rows.length; i++){
  r = rows[i];
  r.children = [];
  all[r.id] = r;
  if(r.path == parent){
    out.push(r);
  }
}
for(i=0; i<rows.length; i++){
  r = rows[i];
  if(all[ r.parent_id ]){
    all[ r.parent_id ].children.push(r);
  }
}

return out;
$BODY$
LANGUAGE plv8 VOLATILE
```

Figura N° 37: Función que retorna JSON de la estructura organizacional.

La función "json_child_tree" contiene la consulta mostrada en la figura N° 38, que extrae de cada tabla los atributos antes mencionados y construye una tabla final con dichos datos, esto lo hace de manera recursiva, pero adicional, construye un campo que fue denominado "path", que contiene los identificadores de los padres representando una ruta de cada contenido en el árbol.

```
WITH RECURSIVE path(name, path, id, parent_id, lane, tid, peso, acronimo) AS (
  SELECT NP.TITLE, NP.NID || '', NP.NID,
  REF.FIELD_REFERENCIA_TARGET_ID, TP.NAME, TP.TID, PR.DELTA, AC.FIELD_ACRONIMO_VALUE
  FROM drupal_node NP
  FULL JOIN drupal_field_data_field_referencia REF
  ON NP.NID = REF.ENTITY_ID
  FULL JOIN
  (SELECT name, entity_id, tid
  FROM drupal_taxonomy_term_data, drupal_field_data_field_tipo_proceso
  WHERE tid = field_tipo_proceso_tid) TP
  ON
  NP.NID = TP.ENTITY_ID
  FULL JOIN drupal_field_data_field_procesos_relacionados PR
  ON
  NP.NID = PR.FIELD_PROCESOS_RELACIONADOS_TARGET_ID
  FULL JOIN drupal_field_data_field_acronimo AC
  ON
  NP.NID = AC.ENTITY_ID
  WHERE NP.NID = 35 AND NP.STATUS = 1
  UNION
  SELECT NP.TITLE, parentpath.path || '.' || NP.NID, NP.NID,
  REF.FIELD_REFERENCIA_TARGET_ID, TP.NAME, TP.TID, PR.DELTA, AC.FIELD_ACRONIMO_VALUE
  FROM
  path as parentpath, drupal_node NP
  FULL JOIN drupal_field_data_field_referencia REF
  ON NP.NID = REF.ENTITY_ID
  FULL JOIN
  (SELECT name, entity_id, tid
  FROM drupal_taxonomy_term_data, drupal_field_data_field_tipo_proceso
  WHERE tid = field_tipo_proceso_tid) TP
  ON NP.NID = TP.ENTITY_ID
  FULL JOIN drupal_field_data_field_procesos_relacionados PR
  ON
  NP.NID = PR.FIELD_PROCESOS_RELACIONADOS_TARGET_ID
  FULL JOIN drupal_field_data_field_acronimo AC
  ON NP.NID = AC.ENTITY_ID
  WHERE REF.FIELD_REFERENCIA_TARGET_ID = parentpath.id AND NP.STATUS = 1)
  SELECT * FROM path ORDER BY tid, peso;
```

Figura N° 38: Consulta de estructura organizacional.

- **Pruebas.**

Se realizaron pruebas para verificar la extracción correcta de la base de datos, para ello, se hicieron diferentes llamadas a la función "json_child_tree" con distintas estructuras organizacionales de prueba, un ejemplo de ella se puede observar en la figura N° 39.

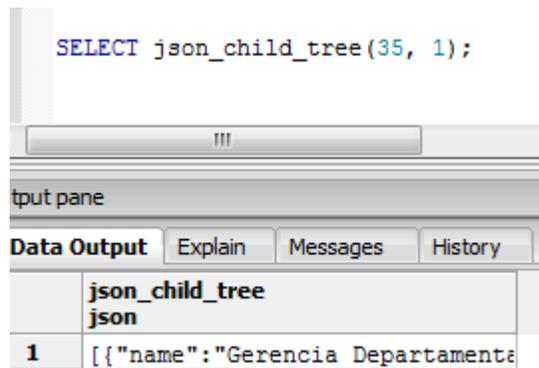


Figura N° 39: Prueba de llamada a función.

Los resultados obtenidos fueron los esperados, los cuales fueron exportados en formato CSV y estaban formados en el formato esperado, lo cual puede observarse en la figura N° 40.

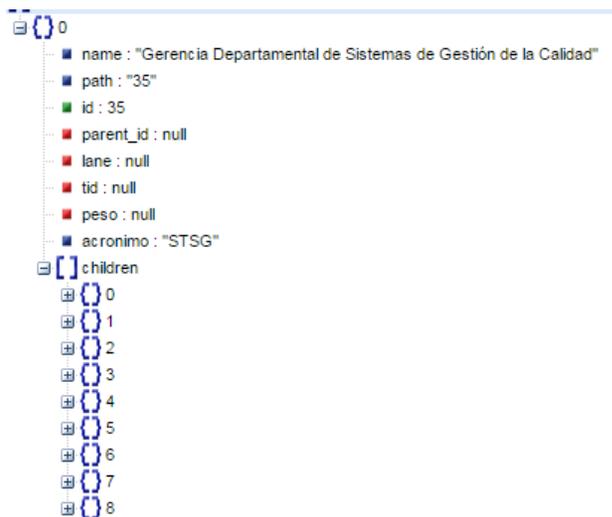


Figura N° 40: Estructura de los datos obtenidos a través de la consulta.

4.3.3.10. 10ma Iteración.

Se construye la estructura organizacional en formato JSON para la posterior construcción del mapa de procesos.

- **Implementación.**

En esta iteración se realiza la llamada desde el modulo creador de mapa de procesos a la función establecida en la base de datos explicada en la sección anterior (*json_child_tree*) para lograr extraer los datos, enviándole el identificador y el estatus del contenido, así permitirá filtrar el contenido. Luego creamos un arreglo asociativo de los resultados obtenidos. Como se puede observar en la figura N° 41.

```
$result=db_query("SELECT json_child_tree($nodo, $status)");

if($result->rowCount()){ //Verificamos que la consulta arroje resultados
  while( $fila = $result->fetchAssoc() ) { //Arreglo Asociativo
    $data = $fila['json_child_tree'];
  }
};
```

Figura N° 41: Llamada a la función en base de datos y construcción de arreglo asociativo.

Así, se obtienen los procesos principales. Ahora, se extraen los procesos internos y externos que fueron referenciados. Para ello se creó una consulta adicional se ejecuta en cada construcción. En la figura N° 42 se puede observar la consulta de los procesos estratégicos internos. Cabe destacar la consulta tiene la misma estructura para los procesos internos, externos y de apoyo. Se consultan datos como el identificador del contenido, el título y acrónimo de las tablas construidas por el manejador de contenido. Luego las filas del resultado devuelto por la consulta en transformado a un arreglo asociativo, el cual será enviado a la construcción del mapa de procesos. Dicho arreglo luego es completado por los resultados devueltos por las otras consultas mencionadas (procesos internos, externos y de apoyo).

```
$result=db_query("SELECT nid as id, title as name, field_acronimo_value as acronimo
FROM drupal_node DP,drupal_field_data_field_proceso_estrat_int PI, drupal_field_data_field_acronimo AC
WHERE PI.entity_id = $nodo AND
AC.entity_id = $nodo AND
DP.NID = PI.field_proceso_estrat_int_target_id");

if($result->rowCount()){
  $data2 = arreglo_asoc($result,"Estratégicos","Interno");
};
```

Figura N° 42: Ejemplo de consulta para procesos internos y externos.

Los resultados de todos los procesos internos y externos que fueron referenciados se fueron adjuntando en la misma variable, adicional tenemos la variable que contiene los procesos principales obtenidos en la consulta de la figura N° 41.

Para finalizar, se realiza la llamada a la función establecida en el módulo de d3, que realizara la construcción del gráfico, enviando el arreglo que contiene los datos de los procesos, el identificador de la etiqueta div que contendrá el mapa de procesos y el tipo de grafico que se va a crear, así como se puede observar en la figura N° 42.

```
$chart = array(
  'id' => 'visualization',
  'type' => 'mapa',
  'data' => $data,
  'data2' => $data2,
  );
return d3_draw($chart);
```

Figura N° 42: Creación de arreglo y llamado a la construcción.

Aquí, como se observa en la figura N° 43 se realizan llamadas a algunas bibliotecas y hojas de estilos, que serán utilizadas para completar algunas funciones del mapa de procesos, como *tooltips* y la exportación del mapa de procesos.

```
drupal_add_js('misc/lib_mapa/jquery.tipsy.js');
drupal_add_js('misc/lib_mapa/d3plus.js');
drupal_add_js('misc/lib_mapa/canvg.js');
drupal_add_js('misc/lib_mapa/rgbcolor.js');
drupal_add_js('misc/lib_mapa/StackBlur.js');

drupal_add_css('misc/lib_mapa/tipsy.css');
```

Figura N° 43: Carga de bibliotecas y hojas de estilos adicionales.

- **Pruebas.**

Se realizó la comprobación de la extracción de los datos de la base de datos, así también la carga correcta de las bibliotecas al momento de generar el grafico.

4.3.3.11. 11ma Iteración.

Se establecieron los elementos que van a permitir la construcción del mapa de procesos. Así como la conversión de los elementos svg en elementos contenedores de textos.

- **Implementación.**

En esta iteración se realiza la construcción del mapa de proceso. La normativa ISO no establece de manera explícita qué procesos o de qué tipo deben estar identificados, si bien induce a que la tipología de procesos puede ser de toda índole. La organización establece un mapa de procesos de tres (3) tipos: estratégicos, medulares y de apoyo.

Para la construcción se obtienen los datos de los procesos, se ordenan según el tipo de proceso y su prioridad, y luego son creados y asignados según su tipo.

Para los procesos estratégicos y de apoyo se utilizó el elemento de rectángulo con bordes curvados. Pero debido al catálogo de elementos preestablecidos en svg, los procesos del tipo medulares se realizaron a través del elemento "path", que permite a través del establecimiento de puntos, crear la forma deseada. Para ello se creó una función como se puede observar en la figura N° 44 que recibiendo la posición "x" y la posición "y", se logra construir la forma del proceso.

```
function fig_med(pos_x,pos_y) {
  return " M " + pos_x + " " + pos_y
    + " L " + (pos_x + 125) + " " + pos_y
    + " L " + (tam_proceso+pos_x) + " " + (pos_y+(alt_proceso/2))
    + " L " + (pos_x + 125) + " " + (alt_proceso+pos_y)
    + " L " + pos_x + " " + (alt_proceso+pos_y)
    + " L " + (pos_x+25) + " " + (pos_y+(alt_proceso/2))
    + "z";
}
```

Figura N° 44: Creación dinámica de procesos medulares.

Los elementos svg no permiten ser contenedores de texto, para ello se integró la biblioteca d3plus de JavaScript que permite esta funcionalidad, y a través de ella permitir la asignación del nombre de proceso a cada tipo de proceso como se observa en la figura N° 45.

```
function wrap(text) {
  text.each(function () {
    var text = d3.select(this);
    d3plus.textwrap()
      .container(d3.select(this))
      .valign("middle")
      .padding(9)
      .draw();
  })
}
```

Figura N° 45: Elementos svg contenedores de texto.

- **Pruebas.**

Se verificó la creación del mapa de procesos y todos los tipos de procesos, como se observa en la figura N° 46. Sin embargo se observó que los títulos extensos no eran abarcados dentro de los procesos como se puede observar en la Figura N° 47.

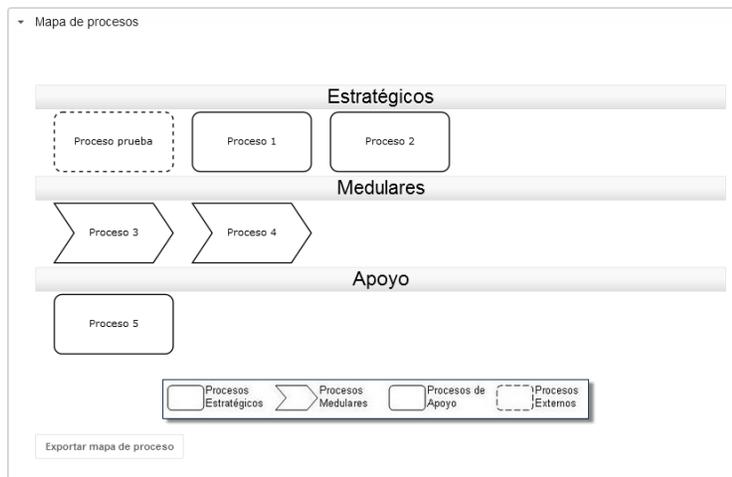


Figura N° 46: Mapa de procesos.

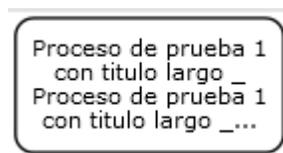


Figura N° 47: Proceso con texto largo.

Como se observa en la figura N° 48, la pestaña contenedora del mapa de procesos igual era creada aunque este no existiera. Lo cual fue solucionado en la iteración siguiente.



Figura N° 48: Pestaña sin mapa de procesos.

4.3.3.12. 12ma Iteración.

Se agregaron *tooltips* al mapa de procesos debido a que no muestra los textos largos y también se agregaron ayudas a los usuarios en los campos de los formularios. También se agregó la funcionalidad de exportar el mapa de procesos.

- **Implementación.**

Se implementó una acción condicional al inicio de la creación del mapa de procesos, que evita la creación de la pestaña contenedora del mapa de procesos si no hay datos para ser graficados.

Como se observa en la figura N° 49, se agrega el *tooltip* a cada proceso representado por el elemento *svg*, para que el usuario pueda observar el título completo del proceso.

```
$('.Mculares, .Interno, .Externo').not(':parent').tipsy({  
  gravity: 's',  
  title: function() {  
    var d = this.__data__;  
    return d.name;  
  }  
});
```

Figura N° 49: Adición de tooltip al mapa de proceso.

También se implementó la redimensión del mapa de procesos para que se adapte al navegador. De igual manera se implementó la validación que al no existir mapa de procesos, no exista la pestaña contenedora del mapa de procesos.

Para finalizar se implementa la funcionalidad para exportar el mapa de procesos agregando el logo de la empresa. Para ello se utiliza la estructura auxiliar canvas, que copia los elementos del svg y sus estilos CSS, y adicionalmente se agrega el logo de la organización para luego ser exportados.

También se agregan ayudas en marca de agua (*placeholder*) a cada uno de los campos de los formularios. También se agrega un *tooltip* en vistas que contienen referencias a otros campos.

- **Pruebas.**

Se verifico la visualización de los *tooltips*, al igual que la redimensión del mapa de procesos según corresponda con el navegador como se puede observar en la figura N° 50.

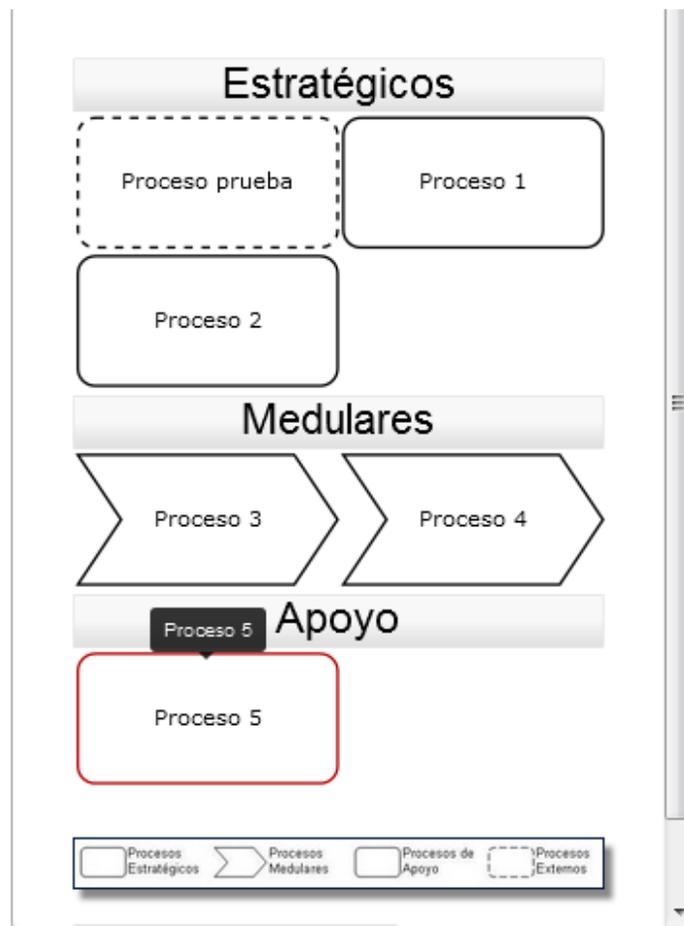


Figura N° 50: Redimensión del mapa de procesos y tooltip.

También se comprueba la funcionalidad de la exportación del mapa de procesos, obteniendo una imagen de formato “.png” con el logo de la organización como se puede observar en la figura N° 51.



Figura N° 51: Mapa de procesos exportado.

4.3.3.13. 13ma Iteración.

Se realizaron pruebas de carga a los servidores de aplicación y de base de datos.

- **Pruebas.**

Siguiendo los lineamientos de la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste, para esta sección se utilizó la herramienta de software Jmeter debido a que es un software sin restricciones de licencia y un proyecto proporcionado por Apache.

- **Prueba de carga al servidor de aplicaciones.**

La aplicación fue desarrollada en esta fase en el servidor de aplicaciones enfocado a desarrollo de la organización, con las características antes descritas en la figura N° 7 y configurado para un máximo de 50 usuarios concurrentes.

Se realizaron 2 pruebas al servidor para analizar el comportamiento. Una con 50 peticiones http, siendo este el tope de conexiones concurrentes, y otra con más de 50 peticiones.

Como se puede observar en la figura N° 52 se realizó una prueba inicial con un grupo de 50 hilos. En el campo de nombre establecemos "Grupo de Hilos" y no añadimos comentarios adicionales en el campo "Comentarios". Para la "Acción a tomar después de un error de Muestreador" seleccionamos "Continuar" para que la prueba continúe en caso de errores. En el campo "Numero de Hilos" establecemos 50 hilos como el grupo de peticiones concurrentes que se van a realizar. En los campos "Período de Subida (en segundos)" y "Contador del bucle" establecemos que en un periodo de 1 segundo y se ejecute una sola vez cada hilo.

Figura N° 52: Configuración Jmeter.

Se observa en la tabla N° 1, en el campo "%Error" que las peticiones no tuvieron porcentaje de error, es decir que el número de peticiones que se pueden observar en el campo "#Muestras" fueron atendidas.

Etiqueta	# Muestras	% Error
Petición HT...	50	0,00%
Total	50	0,00%

Tabla N° 1: Tabla de resultados en pruebas de carga.

Ahora, con una configuración similar a la establecida en la figura N° 52, pero con un grupo de 100 hilos, es decir, que sobrepasa lo establecido en la configuración del servidor, se pueden observar los resultados de la prueba realizada en la tabla N° 2,

arrojando como se puede observar en el campo "%Error" un 27% de peticiones descartadas.

Etiqueta	# Muestras	% Error
Petición HTTP	100	27,00%
Total	100	27,00%

Tabla N° 2: Tabla de resultados en pruebas de carga.

- Prueba de carga al servidor de base de datos.

El servidor de base de datos, con las características antes expuestas en la Figura N° 7, fue comprobado a través de pruebas de carga con una consulta de selección, debido a que es el tipo de consulta que más será utilizada en el sistema.

Al igual que las pruebas de carga al servidor de aplicaciones, las pruebas fueron realizadas con la aplicación Jmeter.

En la figura N° 53 se puede observar la configuración establecida para realizar la prueba, en el campo "Numero de Hilos" se establecen 100 hilos, que corresponden a peticiones concurrentes que se van a realizar, a través de los campos "Período de Subida (en segundos)" y "Contador del bucle" establecemos que en un periodo de 1 segundo y se ejecute una sola vez cada hilo.

Propiedades de Hilo

Número de Hilos: 100

Periodo de Subida (en segundos): 1

Contador del bucle: Sin fin 1

Figura N° 53: Configuración Jmeter.

Debido a que el sistema desarrollado es un sistema de consulta en su mayoría, más que de entrada de datos, la prueba se realizó con una consulta SQL extraída de un contenido de tipo "organización" que ya posee información relacionada. En la Figura N° 54 se observa dicha consulta, que selecciona una serie de campos de este

contenido y realiza uniones y comparaciones que serán común en el sistema. Dicha consulta fue seleccionada debido a que es la consulta más común por la cantidad de consultas a este tipo de contenidos.

```

Query:
1 SELECT node.title AS node_title, node.nid AS nid, node.language AS node_language, field_data_field_obj_estrat_o.delta AS
field_data_field_obj_estrat_o_delta, field_data_field_obj_plan_siembra_o.delta AS field_data_field_obj_plan_siembra_o_delta, node.
created AS node_created, 'node' AS field_data_field_foda_node_entity_type, 'node' AS field_data_field_acronimo_node_entity_type,
'node' AS field_data_field_localidad_node_entity_type, 'node' AS field_data_field_metastact_estrat_o_node_entity_type, 'node' AS
field_data_field_metastact_operac_o_node_entity_type, 'node' AS field_data_field_mision_node_entity_type, 'node' AS
field_data_field_nodo_raiz_ref_node_entity_type, 'node' AS field_data_field_obj_estrat_o_node_entity_type, 'node' AS
field_data_field_obj_plan_siembra_o_node_entity_type, 'node' AS field_data_field_obj_tac_estrat_o_node_entity_type, 'node' AS
field_data_field_obj_tac_operac_o_node_entity_type, 'node' AS field_data_field_objetivos_calidad_node_entity_type, 'node' AS
field_data_field_obsolescencia_node_entity_type, 'node' AS field_data_field_politica_calidad_node_entity_type, 'node' AS
field_data_field_proceso_apoyo_ext_node_entity_type, 'node' AS field_data_field_proceso_estrat_ext_node_entity_type, 'node' AS
field_data_field_ref_organizacion_node_entity_type, 'node' AS field_data_field_retos_o_node_entity_type, 'node' AS
field_data_field_riesgos_foda_node_entity_type, 'node' AS field_data_field_tipo_de_organizacion_node_entity_type
2 FROM
3 drupal_node node
4 LEFT JOIN drupal_field_data_field_obj_estrat_o field_data_field_obj_estrat_o ON node.nid = field_data_field_obj_estrat_o.entity_id
AND (field_data_field_obj_estrat_o.entity_type = 'node' AND field_data_field_obj_estrat_o.deleted = '0')
5 LEFT JOIN drupal_field_data_field_obj_plan_siembra_o field_data_field_obj_plan_siembra_o ON node.nid =
field_data_field_obj_plan_siembra_o.entity_id AND (field_data_field_obj_plan_siembra_o.entity_type = 'node' AND
field_data_field_obj_plan_siembra_o.deleted = '0')
6 WHERE 0((node.status = '1') and (node.nid = 215))
    
```

Figura N° 54: Consulta SQL.

Así, se puede observar en la tabla N° 3, a través del campo “%Error”, que el número de peticiones que se puede observar en el campo “#Muestras” fueron atendidas y que no hubo peticiones descartadas.

Etiqueta	# Muestras	% Error
Petición BD	100	0,00%
Total	100	0,00%

Tabla N° 3: Tabla de resultados en pruebas de concurrencia.

De esta manera, observando un buen rendimiento de los servidores a través de las pruebas realizadas, quedó validada la arquitectura de hardware y software del proyecto.

4.3.4. Roles.

Para este proyecto se abarcaron los siguientes roles:

- **Arquitecto de Software.**

Definió la arquitectura que guió el desarrollo, y la continua refinación de la misma en cada iteración. Definió los lineamientos generales del diseño y la implementación.

- **Desarrollador.**

Tuvo como cargo la codificación de los componentes en código fuente en algún lenguaje de alto nivel a desarrollar en la iteración; elaboró y ejecutó las pruebas unitarias realizadas sobre el código desarrollado.

- **Probador.**

Realizo las pruebas identificadas y definidas previamente, utilizando las instrucciones, métodos y herramientas necesarias. Obtuvo los resultados de las mismas.

Los roles restantes enumerados en la sección 3.1.3, fueron cumplidos por el equipo de trabajo perteneciente a la Gerencia de Automatización, Informática y Telecomunicaciones (AIT) localidad Miranda Oeste.

CONCLUSIONES.

Aplicando la metodología de desarrollo software libre (MeRinde) se logró obtener una aplicación que cumple con los requisitos definidos.

Se realizó la definición de las estructuras que conformaron el establecimiento de los dos módulos desarrollados, a través del diagrama de objetos.

El sistema desarrollado permite crear la estructura organizacional con todo su sistema de gestión de la calidad relacionada, de igual manera la asociación del POA a cada uno de los niveles de la estructura organizacional, de tal manera que los módulos desarrollados sirven de base a los módulos de próximo desarrollo.

La biblioteca D3.js a través de la flexibilidad que brinda, permitió realizar la construcción del mapa de proceso de la estructura organizacional, cumpliendo con el diseño establecido en la organización que es formado por tres tipos de procesos y la normativa ISO 9001.

La creación de módulo de administración permitió el acceso de los usuarios al llenado de las estructuras de listas de términos del sistema, y su continua actualización.

Entre los retos en el desarrollo de esta aplicación se encuentra el uso del CMS Drupal. Se puede decir que la curva de aprendizaje fue elevada; cabe acotar que una vez familiarizado, se facilita el desarrollo de aplicaciones. Entre las dificultades encontradas, se tuvo el aprendizaje de desarrollo de módulos con el API del manejador; sin embargo permite definir el desarrollo de forma modular, logrando con esto tener una mejor organización, también ofrece una gran flexibilidad y escalabilidad en los proyectos.

Se consiguió un sistema que cubre con los lineamientos de diseño de la organización, además de cumplir los requisitos establecidos. Permite al usuario navegar en el sistema a través de la estructura organizacional conociendo todo el sistema de gestión de la calidad de la organización.

REFERENCIAS BIBLIOGRÁFICAS Y DÍGITALES.

- [1] Arturo Jiménez (2014). Especificación de desarrollos de software.
- [2] ISO. (2005) Norma Internacional ISO 9000. Traducción oficial, Sistemas de gestión de la calidad — Fundamentos y vocabulario (4ta Edición). Ginebra, Suiza: Autor.
- [3] López, C. (2001). Las normas ISO 9000. Consultado el 01 de febrero del 2016, de <http://www.gestiopolis.com/canales/gerencial/articulos/27/ISO.htm>.
- [4] <http://www.normas9000.com/que-es-iso-9000.html> consultado el 05 de Febrero de 2016.
- [5] López, M. (2011) "Propuesta de Mejoramiento de los Procesos Operativos de la Empresa bajo el enfoque de Calidad." Informe Final de Pasantía. Universidad Simón Bolívar. Caracas.
- [6] Carbellido, V. (2009) "ISO 9001:2008 Elementos para conocer e implantar la norma de calidad para la mejora continua." Editorial Limusa. México.
- [7] Marcela Acosta (2013). Disponible en URL <https://misindicadores.co/blog/mediciones-que-exige-iso-9001/>
- [8] ISO 9001 calidad. Sistemas de Gestión de Calidad según ISO 9000. Consultado el 02 de febrero de 2016. <http://iso9001calidad.com/la-norma-y-los-indicadores-242.html>
- [9] Corporation, C. (2012). Taller Integral de Procesos. Lima, Perú: Cuality Corporation.
- [10] Pemex. (2013). Guía de aprendizaje de ingeniería de confiabilidad. México: Aprendizaje sin fronteras.
- [11] Nacional, I. P. (2013). Taller de Mapeo y documentación de procesos. México: Departamento de Gestión de la Calidad y Normalización.
- [12] <http://www.ecuadorelectronico.com/CMS.htm>, consultado el 02 de Febrero de 2016.
- [13] Anahy Avalos, (2002). Componentes de una red de cableado estructurado.

- [14] Álvarez Miguel Ángel (01 de enero de 2001) Qué es HTML. Disponible en URL: <http://www.desarrolloweb.com/articulos/que-es-html.html>
- [15] Eguiluz Javier. (s. f.) Introducción a CCS. Disponible en URL: <http://librosweb.es/css>
- [16] Eguiluz Javier (s. f.) Introducción a JavaScript. Disponible en URL: http://librosweb.es/libro/javascript/capitulo_1.html
- [17] Eguiluz Javier (s. f.) Introducción a AJAX. Disponible en URL: http://librosweb.es/libro/ajax/capitulo_1.html
- [18] Álvarez Miguel Ángel (01 de enero de 2001) Qué es PHP. Disponible en URL: <http://www.desarrolloweb.com/articulos/392.php>
- [19] <http://www.json.org/json-es.html> consultado el 05 de Junio de 2015.
- [20] <http://emiliocobos.net/basicos-de-javascript4-cosas-que-no/> consultado el 05 de Junio de 2015.
- [21] Jorge ATGU (05 de febrero de 2014) Qué es SVG. Disponible en URL: <http://www.desarrolloweb.com/articulos/que-es-svg.html>
- [22] <http://d3js.org/> consultado el 04 de Junio de 2015.
- [23] <http://inusual.com/articulos/visualizacion-de-datos-d3-js/> consultado el 04 de Junio de 2015.
- [24] Delgado Expósito Ery (2008). Metodologías de desarrollo de software. ¿Cuál es el camino? Revista de Arquitectura e Ingeniería.
- [25] <http://www.merinde.net/> consultado el 05 de Junio de 2015.
- [26] Arturo Jiménez (2016). Manual y ayuda de usuario para el Sistema de Información para la Gestión de la Calidad (SIGC).