

Universidad de las Ciencias Informáticas

Facultad 1

**“Desarrollo de la versión 2.0 del componente
Estructura y composición del Sistema de Gestión
Universitaria”**

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

Autores:

Julio César Hernández González

Elaimy Arias Lismé

Tutores:

Ing. Alexander Rodríguez Mompié

Ing. Yoan Carlos Machado Espinosa

Ing. Yasmany Tellez Collazo

Ing. Javier Heredia Ruíz

La Habana, Cuba

Junio del 2014

“La felicidad humana generalmente no se logra con grandes golpes de suerte, que pueden ocurrir pocas veces, sino con pequeñas cosas que ocurren todos los días.”

Benjamín Franklin



DECLARACIÓN DE AUTORÍA •



Declaración de autoría

Declaramos ser los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2014.

Julio César Hernández González

Firma del autor

Elaimy Arias Lismé

Firma del autor

Ing. Alexander Rodríguez Mompíe

Firma del tutor

Ing. Yoan Carlos Machado Espinosa

Firma del tutor

Ing. Yasmany Tellez Collazo

Firma del tutor

Ing. Javier Heredia Ruíz

Firma del tutor



Datos de contacto

- **Autores**

Nombre y apellidos: Julio César Hernández González

Dirección: Calle Oscar Acuña No. 58, Manatí, Las Tunas

E-mail: jchdez@estudiantes.uci.cu

Nombre y apellidos: Elaimy Arias Lismé

Dirección: Calle 9na. e/ 8va y Final No. 27, Gaspar, Baraguá, Ciego de Ávila

E-mail: earia@estudiantes.uci.cu

- **Tutores**

Ing. Alexander Rodríguez Mompíe:

Ingeniero en Ciencias Informáticas. Profesor con categoría docente de instructor. Con 8 años de experiencia en el desarrollo de sistemas para la gestión universitaria. Jefe del Departamento de Desarrollo, Dirección de Informatización, Vicerrectoría de Tecnología.

Ing. Yoan Carlos Machado Espinosa:

Ingeniero en Ciencias Informáticas. Especialista de la producción. Con 3 años de experiencia en el desarrollo de sistemas para la gestión universitaria. Especialista de la producción, arquitecto de bases de datos del Departamento de Desarrollo, Dirección de Informatización, Vicerrectoría de Tecnología.

Ing. Yasmany Tellez Collazo:

Ingeniero en Ciencias Informáticas. Especialista de la producción, jefe del proyecto Núcleo del Sistema de Gestión Universitaria. Con 3 años de experiencia laboral. Actualmente trabaja en el Departamento de Desarrollo de la Dirección de Informatización, Vicerrectoría de Tecnología.

Ing. Javier Heredia Ruíz:

Ingeniero en Ciencias Informáticas. Especialista de calidad del proyecto ERP-Cuba, analista en el antiguo centro de desarrollo GEITEL de la Facultad 10. Cuenta con 6 años y medio de experiencia laboral. Actualmente trabaja como profesor del Departamento de Programación y Sistemas Digitales de la Facultad.



A mis padres por creer en mí, por apoyarme en todo momento, por ser guías de mi vida y por darme el honor de ser su hija. Por hacer de mí una mejor persona a través de su ejemplo de honestidad. A mi hermanito Elier, por los ejemplos de perseverancia y constancia que lo caracterizan y el valor mostrado para salir adelante y por su amor.

A mi gran familia por confiar en mí y apoyarme cuando más los necesité, en especial a mi pri Mairelys por mostrarme el camino a seguir y por compartir conmigo excelentes amigos tales como Noel y Alién. A mi tía Zena por educarme y ser otra madre para mí.

A mi compañero de tesis, Julio César por ser tan perfeccionista y difícil, por hacerme ver luz incluso donde todo estaba encendido y por exigirme tanto cuando todo estaba hecho.

A mis compañeros y amigos de toda la carrera, por ser mi familia en estos 5 años, a los que por algún motivo ya no están aquí y los que aún soportan mis defectos. Principalmente a Jorge Luis, Harold, las chulis (Eli, Patry), los chuchitos (Yusel y Carlos), mi esposo (El Yury), Mariely, Yasiel, Froilan, Yordy, Rosa, Yoan, Mario, Betty, Armando en fin a todos, sin importar que mencione su nombre quiero darles las gracias por formar parte de mí y por todo lo que me han brindado.

A mi novio Gabriel (mi nene) por siempre aliviarme en los excesos de preocupación, nerviosismo y por su paciencia y amor.

A mis tutores, Yoan Carlos, Alexander, Yasmany y Javier por su apoyo incondicional y confianza en la elaboración de este trabajo.

Un millón de palabras o lágrimas, no pueden agradecerles a todos los que de una forma u otra me apoyaron. Cultivar verdaderos amigos requiere dedicación y tiempo. No puedo escribir otra cosa que gracias y justamente, agradecer a todos por ser capaces de apoyarme, por callar, por hablar, por ser mis amigos.





A mi mamita que sin ella no sería quien soy en la vida, que nunca dudó de mí aún cuando fallé. Te amo con todo mi corazón. Eres más que importante para mí.

A mi papá que en los momentos importantes de mi vida siempre me dio todo su apoyo.

Y a mi hermanito ya que ellos tres, conforman la piedra angular de mi futuro y las manos que me sostienen e impulsan por esta vida.



Resumen

La Universidad de las Ciencias Informáticas desarrolla una herramienta de apoyo personalizada denominada Sistema de Gestión Universitaria cuyo propósito es automatizar la gestión de varios de sus procesos. Esta herramienta cuenta con un componente denominado Estructura y composición el cual permite representar de forma jerárquica la estructura administrativa de la entidad y asignar responsabilidades a personas pertenecientes a dicha estructura. La Universidad tiene una organización compleja por lo que no es posible representar algunas de sus estructuras con sus responsabilidades asociadas dentro del componente mencionado anteriormente.

En la presente investigación se propone desarrollar una nueva versión del componente Estructura y composición de forma que aumente su nivel funcional para representar estructuras organizativas correspondientes a las organizaciones como políticas, de masas, entre otras. Además, la representación de las estructuras debe ser más simple mejorando su comprensión y permitiendo combinar las unidades organizativas administrativas y sus responsabilidades asociadas con el resto de los tipos de estructuras.

En el desarrollo de la aplicación se respetó la arquitectura definida por la Dirección de Informatización para todos los componentes diseñados para la gestión universitaria, las cuales cumplen con las bases de la migración a software libre.

Palabras clave: estructura organizacional, unidades organizativas.

Índice

INTRODUCCIÓN	1
1 FUNDAMENTACIÓN TEÓRICA	6
1.1 Conceptos asociados al objeto de estudio	6
1.2 Análisis de sistemas informáticos para gestionar la estructura organizacional de una entidad	8
1.2.1 Sistemas para el trabajo con organigramas	8
1.2.2 Sistemas de planificación de recursos empresariales	11
1.3 Resultados del análisis a sistemas informáticos para gestionar la estructura organizacional de una entidad.....	12
1.4 Tecnologías, lenguajes y herramientas a utilizar para la implementación de la solución.	12
1.4.1 Proceso de desarrollo de software DAC.....	12
1.4.2 Lenguajes de programación.....	15
1.4.3 Lenguaje de modelado.....	16
1.4.4 Herramientas	17
1.5 Conclusiones parciales	20
2 CONCEPCIÓN DE LA PROPUESTA DE SOLUCIÓN	22
2.1 Modelo conceptual.....	22
2.2 Descripción de las reglas del negocio.....	23
2.3 Requerimientos del sistema.....	24
2.3.1 Definición de las estrategias de obtención de requisitos	24
2.3.2 Requisitos funcionales	25
2.3.3 Requisitos no funcionales	27
2.4 Descripción de requisitos	28
2.5 Descripción de la propuesta de solución.....	33
2.6 Descripción de la arquitectura.....	35
2.7 Modelo de la base de datos.....	40
2.8 Diagrama de despliegue	43
2.9 Conclusiones parciales	44
3 CONSTRUCCIÓN Y VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN	45
3.1 Paradigmas de programación	45

3.2	Estándares de codificación	47
3.2.1	Identación, tamaño de las líneas, llaves de apertura y cierre	47
3.2.2	Convención de nomenclatura.....	48
3.2.3	Estructuras de control	49
3.2.4	Documentación	49
3.3	Algoritmo para la generación del organigrama.....	50
3.4	Alternativa para el patrón entidad-atributo-valor.....	53
3.5	Estrategia de pruebas.....	54
3.6	Resultados de las pruebas	56
3.7	Conclusiones parciales	60
	CONCLUSIONES GENERALES	61
	RECOMENDACIONES.....	62
	BIBLIOGRAFÍA REFERENCIADA.....	63
	BIBLIOGRAFÍA CONSULTADA.....	65
	GLOSARIO DE TÉRMINOS	68
	ANEXOS	69
	Anexo 1: Descripción de requisitos	69
	Anexo 2 Diagrama Entidad–Relación de la propuesta de solución.....	155
	Anexo 3 Mapa conceptual	156
	Anexo 4: Casos de prueba de caja blanca	156

Índice de figuras

Figura 1. Modelo del proceso DAC.	14
Figura 2. Modelo conceptual.	22
Figura 3. Mapa de navegación.	34
Figura 4. Arquitectura Cliente-servidor.	35
Figura 5. Funcionamiento del MVC en GUUD.	37
Figura 6. Modelo de datos físico (Parte1).	40
Figura 7. Modelo de datos físico (Parte 2).	41
Figura 8. Modelo de datos físico (Parte 3).	42
Figura 9. Diagrama de despliegue.	43
Figura 10. Identación.	48
Figura 11. Variables.	48
Figura 12. Clases.	48
Figura 13. Funciones.	48
Figura 14. Estructuras de control.	49
Figura 15. Documentación de una clase.	50
Figura 16. Documentación de funciones.	50
Figura 17. Representación de un árbol.	51
Figura 18. Representación de un organigrama.	51
Figura 19. Representación de un organigrama. Hijos representados horizontalmente.	51
Figura 20. Representación de un organigrama. Hijos representados verticalmente.	51
Figura 21. Representación de un árbol con nodos por niveles de prioridad (A ((B), (C ((D), E))))).	51
Figura 22. Construcción del organigrama.	53
Figura 23. Usos de hstore.	54
Figura 24. Ejecución de una prueba estructural.	56
Figura 25. Modelo de datos lógico.	155
Figura 26. Mapa Conceptual.	156

Índice de tablas

Tabla 1. Áreas de procesos del nivel 2 de CMMI.....	13
Tabla 2. Listado de requisitos funcionales de la propuesta de solución.....	27
Tabla 3. Descripción del requisito Crear estructura.	32
Tabla 4. Elementos del diagrama de despliegue.	44
Tabla 5. Elementos de la programación orientada a objetos.	46
Tabla 6. Pasos para realizar pre-ajuste.	52
Tabla 7. Pasos para realizar el ajuste.....	53
Tabla 8. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst30_CrearEstructura.	58
Tabla 9. Caso de prueba de caja negra.....	59
Tabla 10. Relación de no conformidades por iteración.	59
Tabla 11. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst29_Listar árbol de estructura.....	159
Tabla 12. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst35_Generar organigrama.	161
Tabla 13. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst30_CrearEstructura.	163
Tabla 14. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst2_Crear categoría estructura.....	165



Introducción

La gestión de entidades y organizaciones se centra cada vez más en las personas. A través de la organización se coordinan los elementos que la componen y sus relaciones. A su vez, a la unidad organizativa se le asocian sus funciones bases correspondientes posibilitando la organización de la estructura. Una estructura organizacional establece una jerarquía de subordinación dentro de una entidad para contribuir a un objetivo común. Además, permite la asignación de responsabilidades de las diferentes funciones y procesos a distintas personas o departamentos.

La gestión de la estructura organizacional y los procesos de una entidad es una tarea compleja. A nivel mundial son utilizados sistemas de gestión que facilitan esta actividad, estos proporcionan nuevas ventajas competitivas mejorando la toma de decisiones, favoreciendo el trabajo en grupo, la rapidez y fiabilidad en la gestión de procesos mejorando el desarrollo de la entidad.

Cuba no se encuentra aislada de este proceso. Entre las acciones que realiza se encuentra la actualización de la estructura organizacional de las entidades con el objetivo de cumplir las metas trazadas por las mismas utilizando soluciones informáticas. Entre las instituciones más destacadas en el desarrollo y uso de sistemas informáticos se encuentra la Universidad de las Ciencias Informáticas (UCI).

La UCI presenta una estructura compleja, determinada por la relación existente entre la estructura administrativa, docente y diversas organizaciones sociales, políticas y de masas que intervienen en ella, como lo son el Partido Comunista de Cuba (PCC), la Unión de Jóvenes Comunistas (UJC), la Federación Estudiantil Universitaria (FEU), el Movimiento de Alumnos Ayudantes (MAA), la estructura investigativa, la organización de los proyectos de producción y desarrollo, entre otras.

En aras de alcanzar las metas de la entidad y mejorar su funcionamiento, servicios y procesos, la Universidad cuenta con una herramienta de apoyo personalizada denominada Sistema de Gestión Universitaria (SGU) que automatiza la gestión de varios procesos de la Universidad como son: pregrado, residencia, postgrado, cooperación, entre otros. Cada área de proceso está vinculada a la estructura administrativa de la institución y tienen sus propias unidades organizativas que le permiten cumplir con sus funciones y actividades.

El SGU cuenta con un componente denominado Estructura y composición el cual gestiona la información referente a la estructura administrativa y su jerarquía. A partir de la información brindada por dicho componente se puede definir a que información se tiene acceso en dependencia de la estructura que se gestione. También posibilita la asignación de las responsabilidades que pueden ser ocupadas en una



estructura, lo que mejora la organización y el acceso a la información del sistema logrando acercar más a la realidad los procesos que se automatizan.

Actualmente dicho componente tiene varios inconvenientes que atentan contra la calidad del sistema. Por una parte tiene un enfoque administrativo, impidiendo la representación de otras estructuras que pueden coexistir en la entidad o diferentes enfoques en la organización, ejemplo de ello son el MAA y el PCC. Esta restricción provoca que en el sistema no se puedan gestionar correctamente los procesos vinculados a estas organizaciones. Además, el componente no gestiona la relación que existe entre distintas unidades organizativas.

La composición de una estructura está formada por un conjunto de responsabilidades que tienen asociadas una cantidad de plazas fijas, por lo que no permite la asignación de plazas temporales o miembros de una organización a dicha estructura. Esto se evidencia al no poder definir una plaza para las personas en adiestramiento. Por otra parte el componente no cuenta con un registro de los diferentes cambios que pueden ocurrir en la estructura organizacional de la entidad. Esto es muy importante para poder realizar un análisis del comportamiento histórico en un período determinado y para conocer la información de las diferentes estructuras a las que ha pertenecido una persona. Además, no permite configurar el orden de la información en dependencia de las categorías y estructuras que se visualizan en el árbol que las representa. Dicho árbol es difícil de comprender y dificulta el trabajo para realizar análisis sobre la estructura por el poco nivel de información que brinda.

El componente no permite la generación de organigrama. Estos elementos son necesarios para facilitar la comprensión de la organización y sus relaciones jerárquicas. Además, el algoritmo utilizado para obtener los elementos de un subárbol de la estructura es recursivo, lo cual influye en el tiempo de respuesta del sistema.

Estas deficiencias atentan contra la calidad del sistema, por lo cual es necesario aumentar las funcionalidades del componente de Estructura y composición del SGU permitiendo definir diferentes dimensiones estructurales que converjan en la entidad, mejorando la gestión de sus procesos.

A partir de la situación existente se plantea el siguiente **problema de investigación**: ¿Cómo dimensionar diferentes tipos de estructuras y sus responsabilidades asociadas, aumentando el nivel funcional del componente Estructura y composición del SGU, en la UCI?

Para dar solución al problema en cuestión se propone como **objeto de estudio** sistemas informáticos que gestionen estructuras organizativas.



El **campo de acción** se centra en la gestión de la estructura organizativa en el componente Estructura y composición del SGU en la UCI.

En consecuencia se define como **objetivo general** desarrollar la versión 2.0 del componente Estructura y composición que permita dimensionar diferentes tipos de estructuras y sus responsabilidades asociadas, aumentando el nivel funcional en el Sistema Gestión Universitaria haciendo uso de tecnologías libres.

Para lograr dicho resultado se definieron las siguientes **tareas de investigación**:

- Análisis de sistemas que informaticen la estructura y organización de las entidades para obtener elementos reutilizables de estas en la solución que se implemente.
- Caracterización del proceso de desarrollo que se aplicará definiendo las fases y principales artefactos a generar.
- Análisis de técnicas de ingeniería de requisitos para aplicar las más idóneas en el desarrollo de la propuesta de solución.
- Definición de pruebas que permitan medir la calidad de la solución para su posterior aplicación.
- Obtención de los artefactos generados según el proceso de desarrollo aplicado para la realización de una propuesta de solución de un producto de software.
- Definición de los patrones de diseño más adecuados para la construcción del sistema propuesto.
- Elaboración del marco teórico conceptual asociado al objeto de estudio y campo de acción con el fin de comprender el mismo.
- Caracterización de las herramientas que se utilizarán en el entorno de desarrollo para obtener un producto basado en tecnologías libres.
- Modelado de la base de datos para definir de manera gráfica el diseño de la estructura de datos de la solución.
- Descripción de la arquitectura para caracterizar los componentes que conforman la solución.
- Descripción del estándar de codificación que regirá la implementación del componente.
- Descripción de las técnicas de programación a utilizar para la correcta implementación de la propuesta.
- Implementación de los requisitos especificados para obtener un software que cumpla con los objetivos planteados.
- Validación de la solución implementada mediante pruebas de funcionalidad para comprobar su correcto funcionamiento.



Para dar cumplimiento a las tareas de investigación se emplearon métodos científicos los cuales se pueden clasificar en teóricos y empíricos entre ellos se encuentran:

Método Teórico:

Histórico-Lógico: este método se utiliza en la fundamentación teórica para conocer las tendencias más relevantes que existen en cuanto a la gestión de estructuras organizativas del componente de Estructura y composición en la UCI, así como en el análisis de la base teórica durante el estudio de las tecnologías.

Métodos Empíricos:

Observación: permite mediante la observación visual de la realidad de los sistemas de planificación de recursos empresariales (ERP) existentes en Cuba, realizar una valoración de la situación actual observando diferentes peculiaridades en la gestión de estructuras, el tiempo de búsqueda de algunas informaciones y la imposibilidad del intercambio de información entre las estructuras organizativas de la Universidad.

Entrevista: permite realizar conversaciones planificadas entre los especialistas funcionales, analista principal y jefe de línea. Fue utilizado para la captura de datos que serviría para la especificación de los requisitos del componente. Se utiliza para precisar el problema a resolver, los problemas existentes y los servicios que brinda actualmente el componente Estructura y composición del SGU de la UCI.

Posible resultado:

Se espera como resultado de la presente investigación la versión 2.0 del componente Estructura y composición que permita dimensionar diferentes tipos de estructuras y sus responsabilidades asociadas en el SGU teniendo en cuenta las nuevas resoluciones, así como los artefactos y validaciones que requiera el componente.

La investigación está estructurada en tres capítulos:

Capítulo 1: Fundamentación teórica: se analizan los principales conceptos relacionados con el objeto de estudio. Se realiza una descripción de los sistemas existentes y una explicación de las herramientas y el proceso de desarrollo a utilizar en la construcción de la propuesta de solución.

Capítulo 2: Concepción de la propuesta de solución: se especifican los requerimientos del sistema. Por otra parte se crean los prototipos de interfaz. A su vez se muestra la utilización de los patrones de diseño y el patrón de arquitectura en la aplicación.

Capítulo 3: Construcción y validación de la propuesta de solución: se describe la construcción de la propuesta de solución teniendo en cuenta el algoritmo para la generación del organigrama. Se realizan las



pruebas a la propuesta de solución a través de la estrategia de pruebas seleccionada y se valoran los resultados de la investigación.

Además, el documento incluye las conclusiones generales y bibliografías consultadas que sirven de soporte a la investigación, así como los anexos donde se encuentran los artefactos generados a lo largo del desarrollo de la investigación.



1 Fundamentación teórica

Introducción

En el capítulo se describen los elementos principales que fundamentan el contenido de este trabajo. Se realiza un estudio de diversos sistemas informáticos que llevan a cabo la representación o gestión de la estructura organizacional de una entidad y la ERP. Además, se detallan las tecnologías, lenguajes y herramientas utilizadas para la implementación del componente Estructura y composición.

1.1 Conceptos asociados al objeto de estudio

Para poder lograr un mejor entendimiento del tema se considera imprescindible definir los siguientes conceptos que giran alrededor del objeto de estudio y del campo de acción:

Organización

El concepto organización en los últimos tiempos ocupa un espacio importante en las agendas de los investigadores, por lo que se considera importante exponer aquellos que más tributan a la construcción teórica del componente a desarrollar.

La organización es la acción y el efecto de articular, disponer y hacer operativos un conjunto de medios, factores o elementos para la consecución de un fin concreto (Andrade, 2005).

Otra definición encontrada en la bibliografía consultada es que la organización es, a un mismo tiempo, acción y objeto. Como acción, se entiende en el sentido de actividad destinado a coordinar el trabajo de varias personas mediante el establecimiento de tareas, roles o labores definidas para cada una de ellas, así como la estructura o maneras en que se relacionarán en la consecución de un objetivo o meta. Como objeto, la organización supone la realidad resultante de la acción anterior; esto es, el espacio, ámbito relativamente permanente en el tiempo bajo el cual las personas alcanzan un objetivo preestablecido (Guerra, 2007).

La organización está compuesta por un conjunto de personas capaces de comunicarse, recursos económicos, materiales y de información que son fusionados para satisfacer un objetivo o diferentes metas. Además, funciona mediante normas y reglas establecidas, que todos sus miembros deben respetar para su correcto funcionamiento.



Estructura y composición

Estructura es la coordinación de una serie de partes o elementos dispuestos en un cierto orden y con determinadas relaciones entre ellos. Orden que ha de ser relativamente duradero. La estructura de la organización es la suma total de los modos en que esta divide su trabajo en distintas tareas y los mecanismos a través de los cuales consigue la coordinación entre ellas. Se trata de un modelo relativamente estable de la organización que no puede identificarse totalmente con ella. Elementos estructurales como la división de funciones, la distribución de puestos y el orden de los distintos niveles de toma de decisiones; es decir, todo lo relativo a relaciones, actividades, derechos y obligaciones que es preciso fijar mediante reglas y ordenanzas (Psicología Online, 2014).

La estructura representa la percepción que tienen los miembros de la organización acerca de la cantidad de reglas, procedimientos, trámites y otras limitaciones a que se ven enfrentados en el desarrollo de su trabajo. La medida en que la organización pone el énfasis en la burocracia contra el énfasis puesto en un ambiente de trabajo libre, informal y no estructurado. Define cómo se dividen, agrupan y coordinan formalmente las tareas de trabajo. Existen 6 elementos claves a los que necesitan enfocarse los directivos cuando diseñan la estructura de su organización estos son: especialización del trabajo, departamentalización, cadena de mando, tramo de control, centralización y descentralización y formación (Carrasco, y otros, 2009).

La composición de una organización está definida por individuos que pueden formar, a su vez, grupos. Estos individuos y grupos funcionan a modo de elementos de un sistema. Para que exista una organización, no existe un número mínimo necesario de individuos, de manera que hay autores que consideran que desde que dos individuos se asocian para conseguir un objetivo, dividiéndose el trabajo, se puede hablar ya de organización (Díaz, 1998).

La estructura y composición de una organización es la forma en que esta se va a gestionar. La estructura permite la representación de los distintos departamentos de una organización. La composición es la disposición de los diferentes cargos o responsabilidades asociadas a una estructura y que pueden ocupar las personas que integran una organización.



Organigrama

El organigrama es a su vez un concepto necesario para comprender mejor la propuesta de solución.

Según Ferrel, Hirt, Adriaenséns, Flores y Ramos, autores del libro "Introducción a los Negocios en un Mundo Cambiante", el organigrama es una representación visual de la estructura organizacional, líneas de autoridad (cadena de mando), relaciones de personal, comités permanentes y líneas de comunicación (Ramos, y otros, 2004).

Para Enrique B. Franklin, autor del libro "Organización de Empresas", el organigrama es la representación gráfica de la estructura orgánica de una institución o de una de sus áreas, en la que se muestran las relaciones que guardan entre sí los órganos que la componen (Franklin, 2004).

Simón Andrade, autor del libro "Diccionario de Economía", proporciona la siguiente definición de organigrama: Expresión gráfica o esquemática de la estructura organizativa de una empresa, o de cualquier entidad productiva, comercial, administrativa, política, entre otros (Andrade, 2005).

Un organigrama es una representación gráfica de las áreas de una empresa u entidad. Además, permite visualizar la estructura y composición de una organización ofreciendo datos generales de sus características. Un organigrama debe ser flexible en cuanto al cambio, ya que debe reflejar todas las transformaciones estructurales de una organización.

1.2 Análisis de sistemas informáticos para gestionar la estructura organizacional de una entidad

Definir la forma más adecuada para gestionar la estructura organizacional de una entidad haciendo uso de un sistema informático es una decisión compleja. Entre los sistemas relacionados con el tema se encuentran los sistemas para el diseño y trabajo con organigramas, usados principalmente para la representación y análisis de la estructura y composición de una entidad; y los sistemas de ERP para la gestión y planificación empresarial basado en la estructura organizativa de una entidad. Con el objetivo de aumentar el nivel funcional del componente Estructura y composición se realiza un estudio de estos sistemas para buscar nuevas características que puedan ser incluidas en la nueva versión, lo planteado anteriormente se puede evidenciar a continuación.

1.2.1 Sistemas para el trabajo con organigramas

***OrgPlus.**

OrgPlus constituye una de las herramientas más intuitivas y potentes para producir organigramas, permite actualizar la disposición del organigrama de forma automática para obtener una óptima organización y



elimina los pasos de agregar o eliminar empleados manualmente. También permite mapear datos existentes a campos de OrgPlus para importarlos con rapidez desde una variedad de fuentes así como sincronizar los datos automáticamente de forma que los organigramas permanezcan siempre actualizados (Insperity Business Services, 2014).

Características:

- Alto rendimiento en el manejo de grandes conjuntos de datos.
- Capacidades de formato avanzadas.
- Planificación de cambios organizativos.
- Opciones avanzadas de exportación.
- Fácil navegación en organigramas complejos.

***SmartDraw.**

Para aprender a emplear el software gráfico tradicional se necesita mucho tiempo y requiere tanto habilidades de dibujo como de formato. SmartDraw es diferente, permite crear representaciones visuales profesionalmente con formato de cualquier clase, incluyendo diagramas de flujo, organigramas, planos, diagramas de red, mapas mentales, diagramas de Gantt, líneas de tiempo, mapas del territorio de ventas, entre otros. Además, permite crear rápidamente un diagrama, o manejar un proyecto, haciendo uso de las plantillas de diseño profesional de SmartDraw, o personalizar dicha plantilla. SmartDraw a medida que crea las imágenes las formatea automáticamente, al igual que un procesador de textos establece formato automáticamente a sus documentos (SmartDraw, 2014).

Características:

- Permite ahorrar tiempo mediante el formato automático de las imágenes.
- Generar organigramas con asistentes y control automático.
- Permite exportar a PDF.

***Microsoft Office Visio**

Microsoft Office Visio es un programa para dibujar diagramas inteligentes integrados en los programas de Microsoft Office desde la versión del 2003. Por otra parte incluye funcionalidades para hacer los diagramas más significativos, más flexibles y mejor adaptados a las necesidades del cliente. También permite copiar dibujos a través de otros programas (como Microsoft Word, Microsoft Excel) (Microsoft, 2014).

Las herramientas de diagramación avanzada de Visio permiten simplificar las tareas complejas con elementos visuales dinámicos basados en datos y nuevas maneras de compartir en la web en tiempo real.



Además, permite crear un diagrama con plantillas de apariencia profesional y modernas formas prediseñadas. Puede vincular fácilmente el diagrama con orígenes de datos comunes (como Excel) y observar la actualización automática de los datos dentro del diagrama reflejada en elementos visuales llamativos como íconos, símbolos, colores y gráficos de barras. También permite publicar el diagrama de datos vinculados en SharePoint y proporciona a otros usuarios acceso a estos datos en la web, aunque dichos usuarios no posean Visio (Microsoft, 2014).

La simplicidad, las formas basadas en datos y el uso compartido en la web convierten a Visio en una de las formas más eficaces de ver y comprender información importante. Utilizando Microsoft Visio se pueden crear gráficos relacionados con el trabajo tales como diagramas de flujo (organigramas), carta de la organización (organigramas) y el calendario del proyecto (programación de proyectos) (Microsoft, 2014).

Características:

- Conexión automática entre nodos del organigrama.
- Uso de efectos como sombra, bisel, iluminación y otros.
- Cuadros de mando sin código.
- Fácil personalización.

***Lucid Chart**

LucidChart es una diagramación en línea, diagrama de flujo y el software de dibujo que apoya la colaboración en tiempo real. Usando solo los estándares web, LucidChart es más rápido y más estable que otras aplicaciones de creación de diagramas en línea y se puede ver en cualquier dispositivo habilitado para la web. LucidChart se basa en estándares web como HTML5 y JavaScript y proporciona una plataforma de comunicación visual para las personas y los equipos virtuales. Se puede utilizar para crear diagramas informativos, diagramas de flujo, organigramas, diagramas funcionales de sitios web, diseño UML, mapas mentales, diseños de habitaciones, planos, maquetas de interfaz de usuario, diseños eléctricos y muchos otros tipos de diagramas. Además, permite exportar sus diagramas a PDF, JPG y PNG (Lucid Software, 2014).

***Creately**

Es una herramienta para la creación y mantenimiento de organigramas, cuenta con un sistema en línea y una versión de escritorio compatible con Microsoft Windows, Mac OS y Linux. Asimismo, todos los organigramas pueden ser exportados como SVG, un tipo de archivo común, que se puede ver con cualquier navegador y muchos otros programas de diagramación compatibles con SVG. Para el diseño cuenta con



filtrado permitiendo que sea fácil de añadir o quitar campos de todo el organigrama con un clic. Esto significa que se pueden mostrar campos para un ambiente de trabajo y ocultarlos del resto de la entidad con solo clic. Ayuda a mostrar u ocultar detalles como sea necesario al distribuirlo. Creately cuenta con un gran conjunto de plantillas de diseño profesional adaptables a las necesidades de la organización. Creately es más que una herramienta para dibujar organigramas en línea. Permite el uso de conectores inteligentes que ajustan automáticamente un objeto. No solo proporciona una manera de dibujar organigramas sino que permite compartirlo, colaborar y publicarlo (Cinergix, 2013).

1.2.2 Sistemas de planificación de recursos empresariales

Subsistema Estructura y composición del Sistema Integral de Gestión Cedrux

El Sistema Integral de Gestión Cedrux para la planificación de los recursos empresariales fue desarrollado en la UCI en Cuba. Es una aplicación web basada en Zend Framework, marco de trabajo para el desarrollo de servicios y aplicaciones web utilizando PHP5, Doctrine Framework mapeador relacional de objetos, ExtJS Framework, biblioteca de JavaScript para el trabajo con el DOM, AJAX, DHTML. Utiliza arquitectura cliente-servidor.

A diferencia de los sistemas ERP existentes en el mundo, los cuales son utilizados para gestionar los procesos en una entidad, el Sistema Integral de Gestión Cedrux ha sido desarrollado en base a las necesidades de Cuba. Brinda la posibilidad de ser multientidad, lo que significa que incluye un componente de Estructura y composición que se encarga de definir la estructura funcional de la entidad que se esté gestionando. Además, puede especificar a quien se subordina dichas entidades y su estructura.

Este sistema tiene entre sus responsabilidades fundamentales:

- Brindar la posibilidad al usuario de definir la estructura organizativa en la cual se va a ubicar la entidad y su estructura, así como permitir que se pueda especificar los cargos por los cuales van a estar compuestas las diferentes áreas dentro de las unidades organizativas.
- Ser configurable para que los usuarios no estén obligado a usar los mismos conceptos estructurales.
- Brinda la posibilidad de asignar subordinaciones a las estructuras conformadas por entidades.
- También posee otras importantes funcionalidades necesarias para cumplir con sus responsabilidades como son gestionar nomencladores, el cual permite al usuario gestionar los elementos o estructuras que luego se utilizarán para realizar la estructura general organizativa de la entidad donde el software esté implementado, por ejemplo los órganos, especialidades, cargos y categorías que pueda tener o no una entidad.



Componente Estructura y composición del SGU versión 1.0

El componente Estructura y composición del SGU en su versión 1.0 pertenece a los componentes del núcleo del sistema y está encargado de representar la estructura administrativa de la institución así como brindar a los demás subsistemas los servicios para poder acceder a dicha información. Entre las principales funcionalidades con las que cuenta cabe destacar la creación de tipos de entidades y entidades, además, permite la creación de la estructura administrativa, categorías de estructuras, asignación de responsabilidades y gestión de atributos.

1.3 Resultados del análisis a sistemas informáticos para gestionar la estructura organizacional de una entidad

Los sistemas para el trabajo con organigramas lograron reflejar que la mayoría de los sistemas para las comunicaciones de negocios y representación de estructuras organizacionales utilizan elementos visuales dado que esto mejora la comprensión y la gestión de la información analizada. Entre las principales características que pueden ser utilizadas en la solución se encuentra la utilización de organigramas, su forma de navegación para casos complejos (expansión y contracción de los nodos), formato automático y generación de organigramas con asistentes y control automático, forma de enlace del organigrama y los datos, efectos (sombra, bisel, iluminación), exportar a PDF y como imagen, el uso de conectores inteligentes y la generación de reportes de la composición de una estructura.

De los sistemas ERP estudiados el subsistema de estructura y composición del Sistema Integral de Gestión Cedrux brindó la posibilidad de definir las subordinaciones entre las estructuras, la gestión de nomencladores. En el caso del SGU el componente de Estructura y composición brinda las funcionalidades más básicas para gestionar la estructura administrativa de la entidad, permitiendo reutilizar y configurar las funcionalidades existentes, así como enriquecerlas con las características aportadas por los sistemas estudiados para aumentar el nivel funcional del componente.

1.4 Tecnologías, lenguajes y herramientas a utilizar para la implementación de la solución.

1.4.1 Proceso de desarrollo de software DAC

Como guía para el proceso de construcción del software se utilizó la metodología Desarrollo Ágil con Calidad (DAC) ya que fue elaborada por el Departamento de Desarrollo de la Dirección de Informatización de la UCI de la cual surge la presente investigación. La metodología DAC es un proceso de desarrollo de software



que combina las metas y prácticas de las áreas de procesos del nivel 2 de CMMI (*Capability Maturity Model Integration*) con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software. Es un proceso colaborativo, recursivo-iterativo, incremental y guiado por procesos y requisitos. Su modelo del proceso es una adaptación del modelo en Cascada a los modelos Programación Extrema y Desarrollo Concurrente. Está enfocado a proyectos pequeños o proyectos grandes divididos en sub-proyectos que desarrollan software de gestión basado en componentes (Sánchez, 2013).

Para una mejor comprensión de la metodología DAC se muestran a continuación las áreas de procesos del nivel 2 de CMMI y los postulados del manifiesto ágil.

Áreas de procesos del nivel 2 de CMMI	Categorías
REQM-Gestión de requerimientos	Ingeniería
PP-Planificación de proyectos	Gestión de proyectos
PMC-Monitorización y control de proyecto	Gestión de proyectos
SAM-Gestión de acuerdos con proveedores	Gestión de proyectos
MA-Medición y análisis	Soporte
PPQA-Aseguramiento de la calidad de proceso y de producto	Soporte
CM-Gestión de la configuración	Soporte

Tabla 1. Áreas de procesos del nivel 2 de CMMI.

Postulados del Manifiesto Ágil:

- Valorar más a los individuos y su interacción que a los procesos y las herramientas.
- Valorar más el software que funciona que la documentación exhaustiva.
- Valorar más la colaboración con el cliente que la negociación contractual.
- Valorar más la respuesta al cambio que el seguimiento de un plan.

DAC plantea que el problema una vez identificado y definido debe ser descompuesto en problemas más pequeños y si es necesario realizar con estos la misma operación. Cada sub-problema es resuelto mediante un componente y el problema resuelto es el software o producto final; por lo que las entregas en DAC son a nivel de iteración en la que hay obligatoriamente un incremento del producto a partir de la solución de un componente del mismo. Además, en cada iteración se define como mínimo un hito a cumplir por cada fase del proceso. Al finalizar cada iteración se realiza la integración del componente al producto obtenido hasta el momento realizando pruebas de integración. Al finalizar las iteraciones se pueden realizar liberaciones del componente y transición del mismo dentro de la fase Cierre de iteración. Las iteraciones no tienen que



desarrollarse todas al mismo tiempo sino que al contar con un equipo pequeño este se va a ir moviendo de una iteración a otra a medida que estas vayan terminando de acuerdo a un orden de prioridad establecido en el plan del proyecto (Sánchez, 2013).

Este proceso tiene 8 actividades del marco de trabajo del proceso común llamadas Fases o Procesos del Ciclo de Vida: Inicio, Análisis y Diseño Arquitectónico, Requisitos, Construcción, Cierre de iteración (opcional), Liberación, Transición y Cierre, ocurriendo las iteraciones concurrentes entre las fases de Requisitos, Construcción y Cierre de iteración. Además, entre las fases de Requisitos y Construcción puede ocurrir un ciclo pues a medida que los requisitos son especificados estos pueden ir entrando a la fase de Construcción. El proceso tiene también dos Áreas de Procesos de Protección: Gestión de proyectos y Soporte así como dos Fases o Procesos Horizontales cuyas tareas están presentes en varias de las fases del proceso común en forma de subprocesos: Arquitectura y Planificación. En la Figura 1 se puede ver el modelo del proceso DAC.

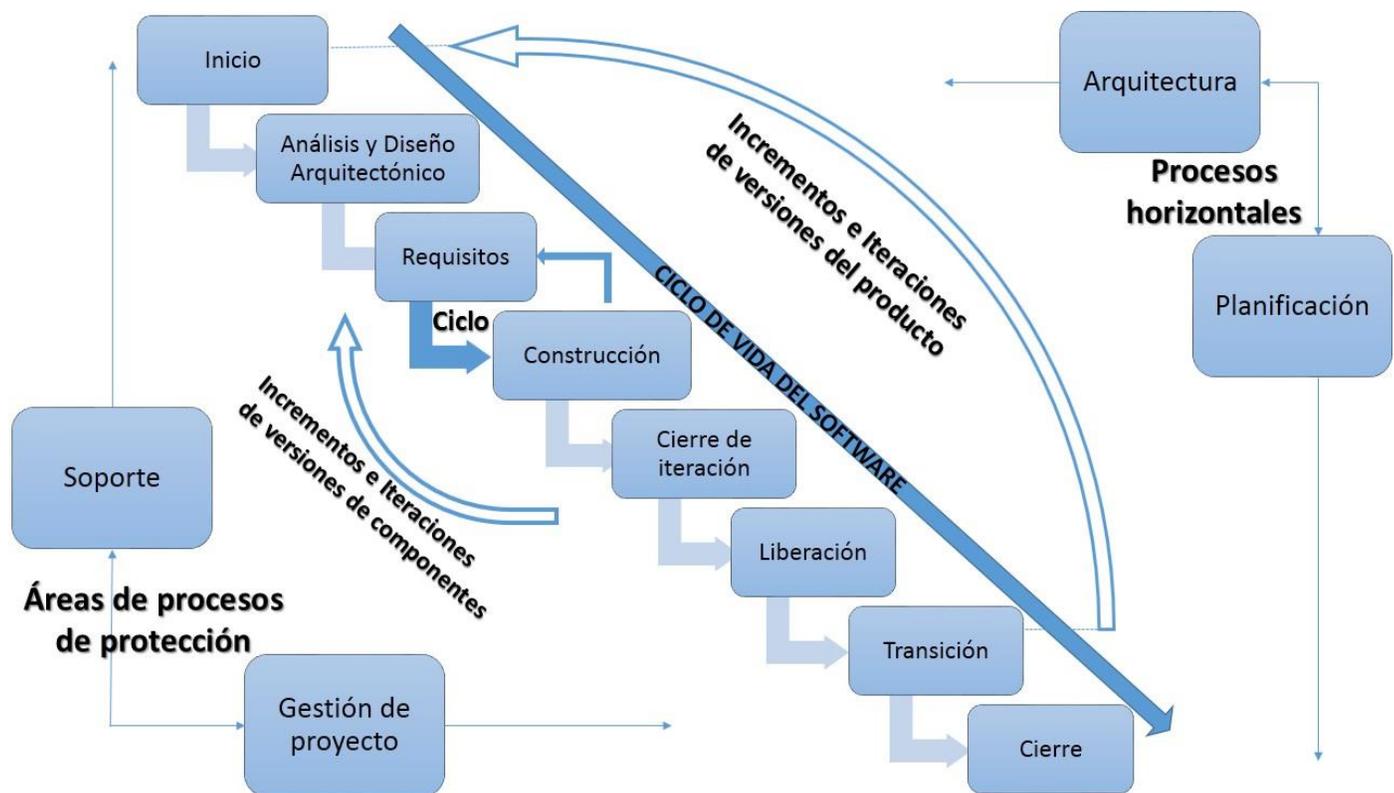


Figura 1. Modelo del proceso DAC.



1.4.2 Lenguajes de programación

Nombre del lenguaje: HTML 4

Descripción de su uso:

Lenguaje de marcas de hipertexto por sus siglas en inglés HTML, es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir hasta un cierto punto, la apariencia de un documento. HTML es el lenguaje de publicación del *World Wide Web* (WWW) (W3C, 2013).

Nombre del lenguaje: PHP 5.3.6

Descripción de su uso:

Preprocesador de hipertexto (*hypertext preprocessor*, por sus siglas en inglés PHP) es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Con PHP no se encuentra limitado a resultados en HTML. Entre las habilidades de PHP se incluyen: creación de imágenes, archivos PDF y películas Flash (usando *libswf* y *Ming*) sobre la marcha. También puede presentar otros resultados, como XHTML y archivos XML. Asimismo, puede autogenerar estos archivos y almacenarlos en el sistema de archivos en vez de presentarlos en la pantalla. Quizás la característica más potente y destacable es su soporte para una gran cantidad de bases de datos. Escribir una interfaz vía web para una base de datos es una tarea simple con PHP (The PHP Group, 2013).

Nombre del lenguaje: JavaScript 1.8

Descripción de su uso:

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, es utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y lenguaje C. Es orientado a objetos, ya que dispone de herencia, la cual se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. JavaScript se ejecuta en el cliente al mismo tiempo que las sentencias van descargándose junto con el código HTML (WEB, 2007).



Nombre del lenguaje: CSS 3

Descripción de su uso:

Las hojas de estilo en cascada (*cascading style sheets*, por sus siglas en inglés CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML. El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación (LibrosWeb, 2014).

Nombre del lenguaje: XML 1.0

Descripción de su uso:

Lenguaje de marcas extensibles (*eXtensible markup language*, por sus siglas en inglés XML). XML es una versión del lenguaje de marcas generalizado (*standard generalized markup language*, por sus siglas en inglés SGML) diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y organizaciones. Entre sus beneficios se encuentran: ser extensible (se pueden añadir nuevas etiquetas tras el diseño del documento), su analizador es estándar (no requiere de cambios para cada versión del metalenguaje), facilita el análisis y procesamiento de los documentos XML creados por terceros (García, y otros, 2013).

1.4.3 Lenguaje de modelado

Nombre de la herramienta: UML 8.0

Descripción de su uso:

Lenguaje unificado de modelado (*unified modeling language*, por sus siglas en inglés UML) es un lenguaje para especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado del negocio y otros sistemas. Un modelo UML está compuesto por tres clases de bloques de construcción; los elementos que son abstracciones de cosas reales o ficticias (objetos, acciones, entre otros) de los cuales existen cuatro tipos de elementos, estructurales, ambientales, grupales y de anotación; las relaciones que son las que relacionan los elementos entre sí, ejemplo: dependencia, asociación, generalización y comprensión; y los diagramas que son las colecciones de elementos con sus relaciones (Craig, 2003).



1.4.4 Herramientas

Nombre de la herramienta: IDE NetBeans IDE 7.4

Licencia del producto: Doble licencia; licencia de desarrollo y distribución común, por sus siglas en inglés, *common development and distribution license (CDDL)* y licencia pública general de GNU versión 2 con excepción de rutas de clases, por sus siglas en inglés *GNU general public license version 2 with classpath exception (GPL2)*.

Descripción de su uso:

NetBeans es un entorno de desarrollo integrado (*integrated development environment*, por sus siglas en inglés IDE) escrito en Java, de código abierto (*OpenSource*), gratuito para desarrolladores de software. Ofrece todas las herramientas necesarias para crear aplicaciones profesionales, empresariales, web y móviles con el lenguaje Java, JavaFX, C/C ++ y lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby. NetBeans es fácil de instalar y se puede ejecutar tanto en Windows, como en Linux, Mac OS X y Solaris. La plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio de gran tamaño. Ofrece servicios comunes permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Es el primer IDE que soporta totalmente Java EE6 y Sun GlassFish Enterprise Server v3. Mejora el soporte para PHP, JavaFX y C/C++ (Corporation, 2013).

Nombre de la herramienta: Pencil 1.3.5

Descripción de su uso:

Pencil es una herramienta para elaborar prototipos de código abierto disponible para todas las plataformas, construida con el propósito de ofrecer una herramienta de creación de prototipos, libre y de código abierto de interfaz gráfica de usuario que la gente pueda fácilmente instalar y utilizar para crear maquetas de plataformas de escritorio populares. Tiene entre sus características principales que los proyectos pueden ser exportados en los formatos HTML, PNG, documento Openoffice.org, Word y PDF, además, permite la instalación de plantillas definidas por el usuario, las operaciones de dibujo estándar: alinear, escalar y rotar y la adición de los objetos externos (Home - Pencil Project, 2012).



Nombre de la herramienta: PgAdmin III 1.14.0

Licencia del producto: Licencia BSD

PgAdmin III es una herramienta de código abierto para la administración de bases de datos PostgreSQL, cuenta con:

- Herramienta de consulta SQL.
- Interfaz administrativa gráfica.
- Editor de código procedural.
- Agente de planificación SQL/shell/batch.

Es considerada la más completa y popular de código abierto. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas (PostgreSQL administration and management tools, 2013).

Nombre de la herramienta: GUUD 1.0

Descripción de su uso:

Marco de trabajo desarrollado por el grupo de arquitectura del Departamento de Desarrollo de la Dirección de Informatización para la creación de aplicaciones web escritas en PHP. Constituye un híbrido entre el *framework* de PHP CodeIgniter y la librería de Javascript, jQuery. Implementa el patrón Modelo-Vista-Controlador (MVC) así como la programación orientada a aspectos. Realiza un manejo de excepciones y mensajes. Modulariza el marco de trabajo CodeIgniter e implementa como estrategia de comunicación entre componentes la Inversión de Control (IoC). Hace uso de *templates* para el renderizado de las vistas. Contiene componentes visuales que permiten la fácil interacción del usuario con la aplicación, tales como el calendario, el *grid*, el *tree*, entre otros (García, y otros, 2013).

Novedades que incorpora el GUUD

A continuación se muestra una relación de las principales mejoras y modificaciones que incorpora el GUUD en su infraestructura.



Del lado del cliente:

- Se implementaron *widgets*¹ para utilizarlos de interfaz de algunos de los *widgets* base de *jquery-ui* como por ejemplo el *date*, el *tab* (ambos son interfaces de los *widgets* de mismo nombre de *jquery-ui*) y el *popup* (interfaz del *dialog* de *jquery-ui*). Además de los ya mencionados, se implementaron otros nuevos entre los que se encuentran: *attach*, *menu*, *message*, *tooltip*, *form* (se construyó con la unión de los *plugins form* de jQuery el cual se utiliza para el envío de formularios la tecnología AJAX y el *validate* utilizado para validar formularios), *grid* (utiliza como plugin el *jqgrid*), *multiselect* (para hacer selecciones múltiples), *navbar* (para la creación de barras de navegación), *tree* (para la creación de árboles) y el *graph* (utiliza la librería Highchart para la generación de gráficas).
- Se le implementó un plugin a jQuery para el manejo de espacios de nombres e internacionalización.
- Se implementaron funciones comunes para todo el sistema (contenidas en los archivos *core.js* y *common.js*) entre las que se destacan: *loadIn*, *getDataJson*, *createSelect*, *isArray*, *isFunction*, *site_url*.

Del lado del servidor (hechas a CodeIgniter):

- Se le implementó la programación orientada a aspectos.
- Se le agregó manejo de excepciones y mensajes.
- Se le implementó el IoC para la interacción entre componentes.
- Se le añadió la característica de la modularidad, o sea que una aplicación pueda dividirse en componentes. CodeIgniter no cuenta con esta posibilidad.
- Se añadieron, modificaron y extendieron los *helpers* o asistentes entre los que se encuentran:
 - Añadidos: *template* (brinda la posibilidad de usar plantillas, característica que no posee CodeIgniter. Para esto se añadió también la librería *template*), *assets* (utilizado para la integración en las vistas de javascript, css, imágenes y el *template*), *grid*, *json*.
 - Modificados: *form*, *array*, *security*.
- Se añadieron los plugins *export_pi* (permite exportar a los formatos: pdf, csv y xls) e *import_pi* (permite importar desde archivos en formatos csv o xls).

¹ Un *widget* es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.



Nombre de la herramienta: Apache 2.2.22

Descripción de su uso:

Apache es un servidor web altamente configurable, robusto y estable. Es un sistema de código abierto para plataformas Windows, Unix, Macintosh y otras que implementa el protocolo HTTP (Ciberaula, 2010).

Características principales:

- Es una tecnología gratuita de código fuente abierta.
- Es personalizable, la arquitectura modular de Apache permite construir un servidor hecho a la medida y posibilita la implementación de los últimos y nuevos protocolos.
- Es un servidor altamente configurable de diseño modular. Permite aumentar fácilmente su capacidad e instalar cualquier componente para cumplir una función específica.
- Permite personalizar la respuesta ante posibles errores, ya que posibilita configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Permite la creación de ficheros de log facilitando, de este modo, el control de las acciones realizadas en el servidor.

Nombre de la herramienta: PostgreSQL 9.1

Licencia del producto: Distribuida bajo la licencia de postgresql, licencia de código abierto, similar a las licencias BSD o MIT.

Descripción de su uso:

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacional. Pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, PostgreSQL incluye características de la orientación a objetos como son: la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, además de otras específicas del gestor, como lo son: un mejor soporte para *sub-selects*, *triggers*, vistas y procedimientos almacenados (The PostgreSQL Global Development Group, 2014).

1.5 Conclusiones parciales

Los conceptos asociados al objeto de estudio permitieron obtener un mejor entendimiento del proceso, adquiriendo y fomentando los conocimientos necesarios para el desarrollo de la propuesta de solución. Después de realizar un análisis de sistemas para el diseño, trabajo con organigramas y de ERP se concluyó que estos aportaron ideas y características para realizar nuevas funcionalidades en la propuesta de solución. Por otro lado la aplicación de la metodología DAC permitió guiar el proceso de desarrollo de la aplicación para la gestión de la estructura organizacional. Las herramientas establecidas por el



Departamento de Desarrollo de la Dirección de Informatización permiten la elaboración de una solución libre, multiplataforma e integrada al SGU.



2 Concepción de la propuesta de solución

Introducción

En el presente capítulo se propone modelar la propuesta de solución a partir de artefactos que tendrán un valor significativo en la etapa de implementación. Se definen las reglas del negocio que deben cumplirse, se identifican y especifican los requisitos funcionales y no funcionales y la elaboración de una arquitectura que dé solución al objetivo planteado.

2.1 Modelo conceptual

En la presente investigación se representa un modelo conceptual que es un diagrama que refleja la organización conceptual de la situación problemática existente. Se realiza con el objetivo de representar mediante conceptos los problemas presentes en el componente Estructura y composición.

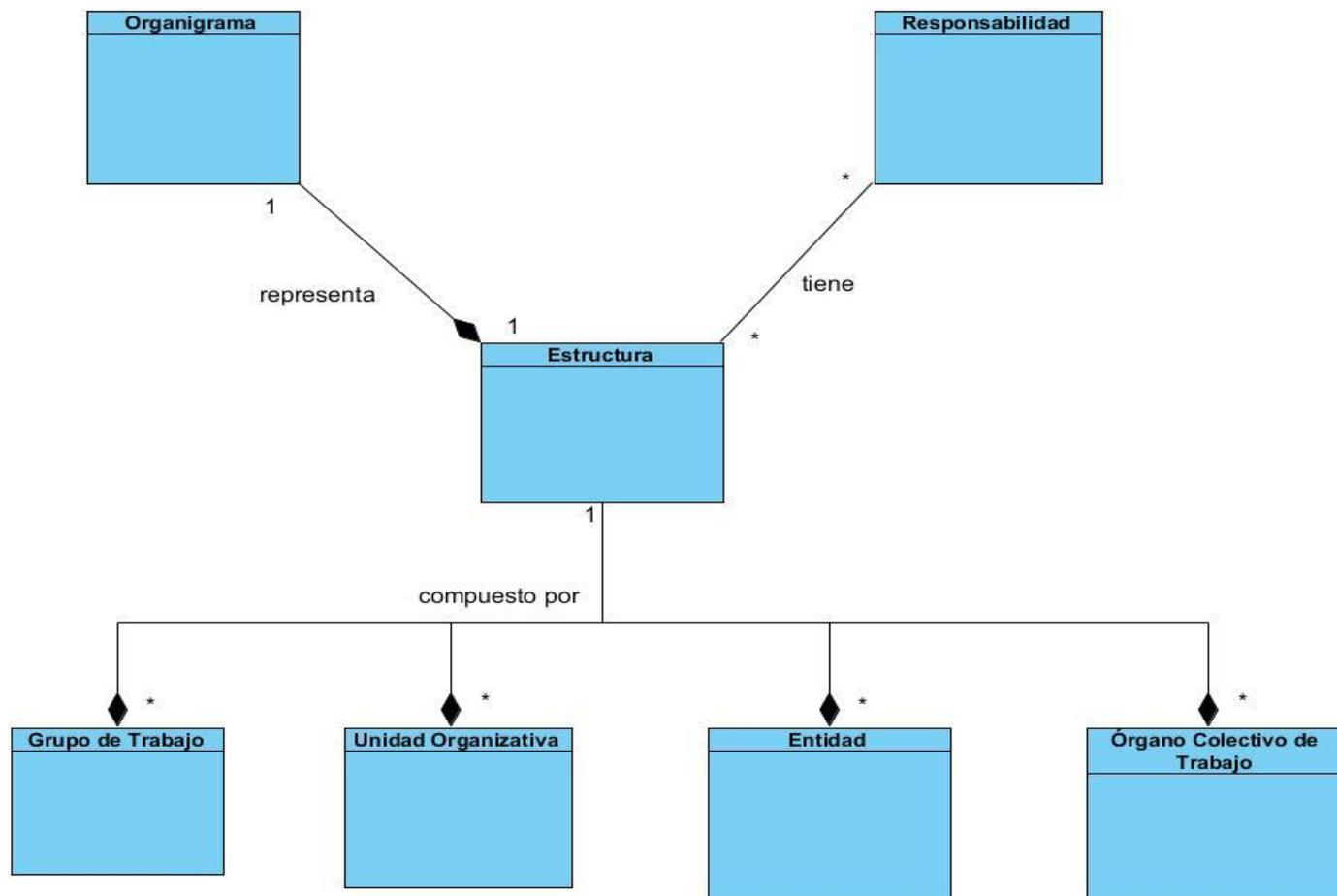


Figura 2. Modelo conceptual.



Para una mejor comprensión, a continuación se describen cada uno de los conceptos que intervienen en el modelo conceptual mostrado.

Estructura: es la forma en que las actividades de una organización se dividen, organizan y coordinan.

Organigrama: representa de forma gráfica la relación entre los elementos que conforman la estructura organizativa.

Responsabilidad: es el cargo que ocupa una persona en una estructura.

Grupo de trabajo: es un conjunto de personas asignadas o auto asignadas, de acuerdo a habilidades, conocimientos y competencias específicas para cumplir una determinada meta bajo la conducción de un coordinador.

Unidad organizativa: estructuras administrativas que tiene una plantilla definida.

Entidad: es una organización o institución formada por un grupo de personas, bienes materiales y financieros.

Órgano colectivo de trabajo: grupo de personas que organizan o dirigen una actividad.

2.2 Descripción de las reglas del negocio

Una regla de negocio es aquello que se usa para operar un negocio. Son las guías que determinan cómo se lleva el día a día de las operaciones. Una regla de negocio define o limita un aspecto del negocio con el objetivo de establecer una estructura o un grado de influencia que condiciona el comportamiento de los actores del negocio. El *Business Rules Group*, que es una organización cuyo propósito es fomentar el entendimiento y estandarización del concepto de reglas de negocio, se basa en dos perspectivas para definir una regla de negocio (Group, 2014):

- Desde la perspectiva del negocio, es una orientación en la cual hay una obligación conducida por una acción, práctica o proceso, dentro de una particular actividad o giro.
- Desde la perspectiva de sistemas de información, es una declaración que define o restringe algunos aspectos del negocio. Intenta hacer valer la estructura del negocio, o controlar o influir en la conducta del negocio (Group, 2014).

A partir del estudio y análisis realizado se identificaron las siguientes reglas del negocio que deben cumplirse para el correcto funcionamiento del sistema:

- Para definir una estructura es necesario establecer la entidad con la cual se trabajará en el sistema.
- Cada estructura tiene asociada una categoría.



- Cada estructura está relacionada con una estructura padre a menos que sea la entidad con la cual se trabajará en el sistema.
- Una estructura de mando está directamente relacionada con una responsabilidad.
- Para definir una composición es necesario crear la estructura.
- Una estructura no puede ser clonada en una de sus estructuras hijas.
- Una estructura solo puede ser combinada con otra estructura que sea solamente de la misma categoría.
- Para poder visualizar el organigrama de una estructura debe estar creada dicha estructura.
- Toda entidad tiene asociada un tipo de entidad.
- Toda organización tiene asociada un tipo de organización.
- Para validar los atributos se debe primeramente asociar una categoría de estructura o de responsabilidad.
- Todas las estructuras dentro de una entidad deben estar agrupadas en un concepto organizativo.
- La composición de una estructura está determinada por plazas asociadas a una responsabilidad.
- Para definir nivel de prioridad entre las categorías del árbol de estructuras se debe configurar el árbol para que muestre las categorías de estructuras.
- Cada estructura dentro del árbol de estructura tiene definido un nivel que no puede ser mayor que el mayor nivel registrado más uno dentro de su estructura padre.
- El organigrama debe mostrar para cada elemento el color de la categoría a la cual pertenece dicho elemento.
- El organigrama debe mostrar la jerarquía de las estructuras de forma vertical.

2.3 Requerimientos del sistema

2.3.1 Definición de las estrategias de obtención de requisitos

Para la recopilación y obtención de la información necesaria para la captura de requisitos existen varias técnicas que permiten establecer una buena comunicación con los interesados y su equipo de trabajo.



A continuación, se mencionan varias de las técnicas tradicionales para la captura de requisitos a partir de los trabajos de Sommerville² (2005) y Pressman³ (2005):

- Entrevistas
- Despliegue de la función de calidad (DFC)
- Sesiones de tormentas de ideas (*Brainstorming*)
- Cuestionarios
- Prototipado
- Casos de uso o escenarios

De las técnicas anteriormente mencionadas se utilizaron para la identificación de los requisitos de la propuesta de solución los siguientes:

- Entrevista: se utilizó con el fin de obtener un intercambio mediante preguntas con los especialistas para esclarecer con precisión el funcionamiento de todo el trabajo de Estructura y composición en la entidad.
- Prototipado: la elaboración de los prototipos permitió mostrar las funcionalidades de la propuesta de solución para su validación por parte de los interesados.

A partir de las técnicas de obtención de requisitos utilizadas se identificaron 44 requisitos funcionales los que se encuentran en las agrupaciones funcionales Estructura, Composición y Configuración del componente.

2.3.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (Sommerville, 2005). El sistema propuesto cuenta con los siguientes requisitos funcionales:

² Ian F. Sommerville, (1951) es un académico británico. Profesor de Ingeniería de software en la Universidad St Andrews, Escocia y autor de un popular libro de texto del estudiante en la ingeniería de software.

³ Roger S. Pressman es un ingeniero de software americano, autor y consultor, presidente de R.S. Pressman & Associates, tiene alrededor de 30 años de experiencia trabajado como ingeniero de software, administrador, profesor, autor y consultor, centrándose en temas de ingeniería de software.



Código	Descripción
RFestC1	Listar categorías de estructura
RFestC2	Crear categoría de estructura
RFestC3	Modificar categoría de estructura
RFestC4	Ver detalles de categoría de estructura
RFestC5	Listar entidades
RFestC6	Crear entidades
RFestC7	Modificar entidades
RFestC8	Establecer entidad activa
RFestC9	Ver detalles de entidad
RFestC10	Listar tipo de entidad
RFestC11	Crear tipo de entidad
RFestC12	Modificar tipo de entidad
RFestC13	Ver detalles de tipo de entidad
RFestC14	Listar organización
RFestC15	Crear organización
RFestC16	Modificar organización
RFestC17	Ver detalles de organización
RFestC18	Listar tipo de organización
RFestC19	Crear tipo de organización
RFestC20	Modificar tipo de organización
RFestC21	Ver detalles de tipo de organización
RFestC22	Listar atributos de estructura
RFestC23	Crear atributo de estructura
RFestC24	Modificar atributo de estructura
RFestC25	Ver detalles de atributo de estructura
RFestC26	Configurar visualización
RFestC27	Establecer nivel de prioridad de categoría de estructura
RFestC28	Asociar responsabilidades con cantidad de plazas variables
RFestC29	Listar árbol de estructura
RFestC30	Crear estructura
RFestC31	Clonar estructura
RFestC32	Combinar estructuras
RFestC33	Crear estructura de mando
RFestC34	Dimensionar en el tiempo el árbol de estructura
RFestC35	Generar organigrama
RFestC36	Mostrar datos de composición
RFestC37	Listar categoría de responsabilidad
RFestC38	Crear categoría de responsabilidad
RFestC39	Modificar categoría de responsabilidad
RFestC40	Ver detalles de categoría de estructura de responsabilidad



RFestC41	Listar responsabilidad
RFestC42	Crear responsabilidad
RFestC43	Modificar responsabilidad
RFestC44	Ver detalles de responsabilidad

Tabla 2. Listado de requisitos funcionales de la propuesta de solución.

2.3.3 Requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. Los requerimientos no funcionales como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema (Sommerville, 2005).

El sistema propuesto cuenta con los siguientes requisitos no funcionales:

Usabilidad

RnF_1. Requisito de usabilidad 1. Desarrollar una solución web integrada al SGU.

RnF_2. Requisito de usabilidad 2. Las vistas del sistema deben indicar en cada momento la acción que se está realizando así como los íconos deben estar representados por una imagen acorde a la acción que se realiza mediante el mismo.

Hardware y software

RnF_3. Requisito de hardware y software 3. Para el uso del sistema se requiere una PC cliente con el navegador web Mozilla Firefox 10 (o superior).

Seguridad

RnF_4. Requisito de seguridad 4. La información que se registra estará protegida de modificaciones no deseadas de acuerdo a la estrategia de seguridad definida.

Confiabilidad

RnF_5. Requisito de confiabilidad 5. Cuando ocurre una excepción el sistema mostrará un mensaje explicativo del error ocurrido y permanecerá en el mismo estado sin realizar ninguna otra operación.



Soporte

RnF_6. Requisito de soporte 6. El sistema cumplirá con las normas de codificación, conversiones para nomenclatura, bibliotecas de clase, normas de acceso y utilidades de mantenimiento definidas para el SGU en el documento Vista de Arquitectura de Datos e Integración.

RnF_7. Requisito de soporte 7. El sistema contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.

Restricciones de diseño

RnF_8. Restricción de diseño 8. El sistema cumplirá con la arquitectura de información definida para el SGU en el documento Vista de Arquitectura de Presentación.

RnF_9. Restricción de diseño 9. El sistema estará desarrollado con las herramientas definidas para el SGU en el documento Vista de Arquitectura de Tecnología e Infraestructura.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

RnF_10. Requisito para la documentación de usuarios en línea y ayuda del sistema 10. El sistema muestra un mensaje a los usuarios de confirmación o notifica acerca del resultado exitoso o no de las acciones realizadas.

Requisitos legales, de derecho de autor y otros

RnF_11. Requisito legal, de derecho de autor 11. Se procederá a una evaluación y certificación por parte del cliente del producto documentando en el expediente del proyecto.

Estándares aplicables

RNF_12. Requisito de Estándar aplicable 12. El sistema estará guiado por lo establecido en el documento Vista de Arquitectura Guía Base.

2.4 Descripción de requisitos

Con la descripción de los requisitos de software se describe cada detalle a tener en cuenta para el desarrollo del sistema lo que ayuda a evitar errores en posteriores etapas, por lo que la utilización de técnicas para la captura y validación asegura la calidad de estos requisitos. Todos los requisitos se describen mediante la técnica escenarios y prototipos. A continuación se presenta como ejemplo la descripción del requisito Crear estructura.



N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC30	Crear estructura	<ul style="list-style-type: none"> • El requisito permite crear estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Estructura del menú de funcionalidades Estructura. • El usuario selecciona la opción crear e introduce los datos: Tipo de estructura, Nombre, Hijo de, Siglas, Categoría de estructura, Nivel, Fecha de activación y Estado. • Cuando el usuario selecciona la categoría de la estructura el sistema muestra los datos a llenar según la selección que realice. • El usuario puede seleccionar vincular la estructura a otra estructura, donde deberá seleccionar en el árbol de estructura que estructura desea vincular a la estructura creada. • Si el usuario desactiva la estructura, no se mostrará el dato fecha de activación. 	Alta	Alta



- El usuario puede crear otra estructura o volver al listado el cual se actualizará.

Prototipo

Prototipo

Crear estructura

Tipo de estructura: *

Nombre: *

Hijo de: *

Siglas: *

Categoría de estructura: *

Nivel: *

Fecha de activación: *

Activo

Vincular a estructura



Crear estructura

Tipo de estructura: * Nombre: * Hijo de: *

Siglas: * Categoría de estructura: * Nivel: *

Fecha de activación: * Activo

Vincular a estructura

- Rectorado
- 100-002-Rectorado
 - Organizaciones
 - Secretaría
 - Departamento central
 - Dirección

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Único Alfanumérico, admite guiones bajos y espacios entre palabras.
<ul style="list-style-type: none"> Hijo de 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Siglas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y



			espacios entre palabras. <ul style="list-style-type: none"> Admite de 1 a 7 caracteres.
<ul style="list-style-type: none"> Categoría de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Alfanumérico, admite guiones bajos y espacios entre palabras. No se admiten los caracteres: %, comilla simple (') y espacios en los límites de la palabra. 	
<ul style="list-style-type: none"> Nivel 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Obligatorio Selección 	
<ul style="list-style-type: none"> Fecha de activación 	<ul style="list-style-type: none"> Timestamp 	<ul style="list-style-type: none"> Obligatorio Selección 	
<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección 	
<ul style="list-style-type: none"> Vincular a estructura 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Selección 	
<ul style="list-style-type: none"> Atributos 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección Admite solo letras. 	
Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente." En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". El estado se muestra activo por defecto. La entidad varía según el valor del tipo de estructura seleccionado. 		

Tabla 3. Descripción del requisito Crear estructura.



2.5 Descripción de la propuesta de solución

La propuesta de solución tiene entre sus principales objetivos garantizar una correcta definición de la estructura organizativa perteneciente a la entidad con la que se trabaja en el sistema.

El componente Estructura y composición cuenta con tres agrupaciones funcionales, de estas el Área de Estructura posibilita a los usuarios que tienen acceso a las funcionalidades del componente definir las categorías de estructuras que permiten agrupar bajo un mismo concepto diferentes elementos de la estructura organizativa así como asignarle un color que la identifique. Además, se le permite al usuario interactuar con el árbol de estructura y mostrar las estructuras asociadas. También brinda la posibilidad de registrar nuevos elementos a la estructura, así como definir su composición en la cual se establecen las responsabilidades y la cantidad de personas que pueden ocupar ese puesto, así como definir la estructura de mando interna de cada unidad. Asimismo, crea varias copias de una misma estructura con su composición a partir del uso de la funcionalidad clonar. Para unificar dos estructuras se utiliza la opción de combinar estructuras de una misma categoría entre sí con lo cual se organiza temporalmente de manera automática cada una de las unidades de la estructura y sus composiciones. Para poder comprender mejor la estructura organizativa el usuario interactúa con el organigrama de la entidad.

En la agrupación funcional Composición se definen las responsabilidades que puede tener una persona así como las categorías de responsabilidades a las que pertenece. En las funcionalidades de Configuración se define las entidades con las cuales se trabajará en el sistema, los tipos de entidades correspondientes y la entidad principal. Además, se establecen los distintos conceptos organizativos en los que se puede estructurar la entidad. También se registran los atributos dinámicos que se utilizan para nuevas responsabilidades o categorías. Para una mejor comprensión se muestra el mapa conceptual en el Anexo 3. A continuación se muestra en las Figura 3 el mapa de navegación para poder hacer uso de la propuesta de solución.

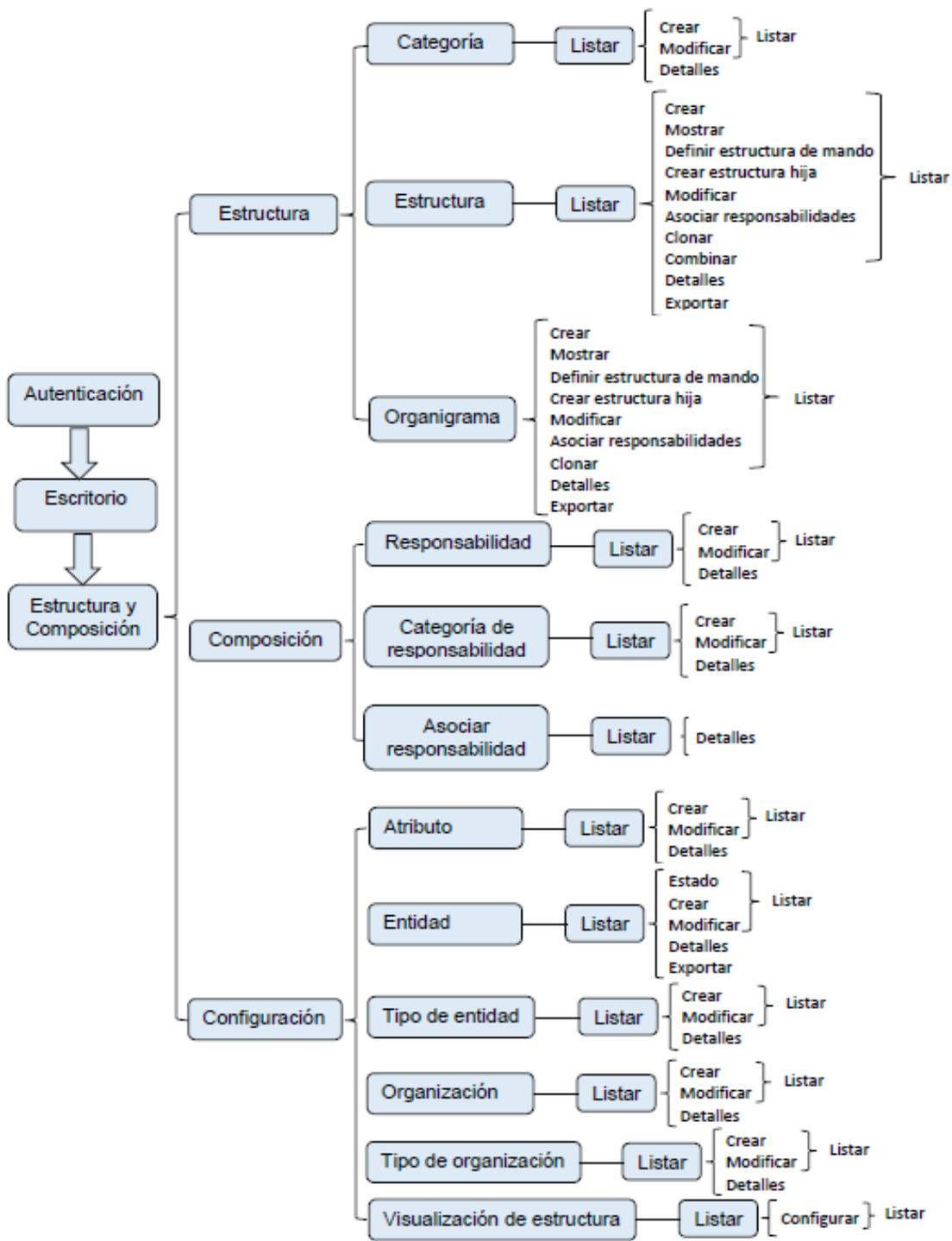


Figura 3. Mapa de navegación.



2.6 Descripción de la arquitectura

La propuesta de solución está concebida como un componente dentro del SGU, por esta razón se ajusta a la arquitectura definida para dicho sistema, la cual propone la arquitectura Cliente-Servidor haciendo uso del patrón Modelo-Vista-Controlador (MVC) que a su vez utiliza el marco de trabajo GUUD, estructura de soporte en la cual se está desarrollando este sistema.

La arquitectura Cliente-Servidor consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las que están contruidos los algoritmos distribuidos.

En la siguiente figura se muestra una representación de esta arquitectura:

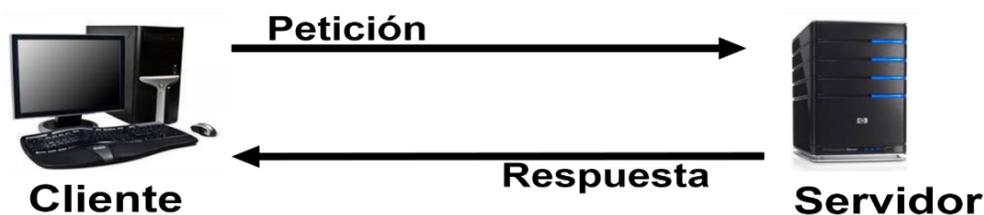


Figura 4. Arquitectura Cliente-servidor.

Como parte de la conformación de la arquitectura se encuentra el patrón arquitectónico Modelo-Vista-Controlador contenido dentro del estilo arquitectónico de llamada y retorno. Este separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Este patrón se ve frecuentemente en aplicaciones web donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio, mientras que el controlador es el responsable de recibir los eventos de entrada desde la vista (Alvarez, 2009).

Los elementos de este patrón son:

Modelo

- Representa a toda la información con la que opera la aplicación.
- Gestiona el comportamiento y los datos del dominio.
- Responde a las peticiones de información sobre el estado, que vienen de la Vista.
- Responde a instrucciones de cambio de estado, provenientes del Controlador.



Vista

- Gestiona la presentación de la información de la aplicación. Todo lo relativo a la interfaz de usuario, los datos de que dispone para seguir interactuando con la aplicación.
- Desde la interfaz gráfica a los estímulos que recibe del usuario, visual, auditiva o sensitivamente.

Controlador

- Respuesta a eventos invocados desde la Vista.
- Llama a la lógica de negocio para procesar y producir una respuesta.
- Interpreta las entradas del usuario informando al modelo y/o a la vista de los cambios que supongan esas entradas.

Implementación del MVC que realiza GUUD

En la Figura 5 se muestra la representación del patrón MVC evidenciado en el marco de trabajo GUUD el cual se describe a continuación:

El marco de trabajo GUUD cuenta con un controlador frontal que inicializa los recursos básicos necesarios para correr el CodeIgniter (parte estructural del GUUD). Una vez realizada una petición HTTP por un cliente, el controlador frontal se encarga de analizar la URI⁴ y a partir de esta determina cuál controlador de aplicación (controlador de un determinado componente) debe ser cargado para atender la petición realizada. Cada controlador de aplicación tiene asociada una o varias librerías responsables de procesar los datos e implementar la lógica del negocio inherente a las acciones relacionadas con dicho controlador. De manera similar cada librería tiene asociado uno o varios modelos encargados del acceso a los datos.

Cuando un controlador de aplicación es cargado, este examina la petición para determinar si solo debe cargar una vista determinada o si es necesario interactuar con la base de datos. En este último caso el controlador de aplicación envía los datos recibidos a la o las librerías. Estas a su vez cargan los modelos necesarios para obtener, registrar o actualizar en la base de datos la información solicitada o enviada. La sintaxis de la consulta es generada por cada adaptador de base de datos. Cuando los datos son obtenidos, se retornan al controlador de aplicación en un proceso inverso al descrito anteriormente. Posteriormente, el controlador carga estos datos a archivos escritos en HTML los cuales pueden incluir llamadas a archivos

⁴ Un Uniform Resource Identifier o URI (en español, Identificador Uniforme de Recurso) es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, entre otros). Su principal diferencia con el Uniform Resource Identifier o URL (en español, Localizador Uniforme de Recurso) es que permite especificar que parte del recurso se solicita (segmentos).



escritos en JavaScript para manejar dinámicamente su contenido. Finalmente, el resultado obtenido de todo este proceso es enviado al navegador web como respuesta a la petición inicial.

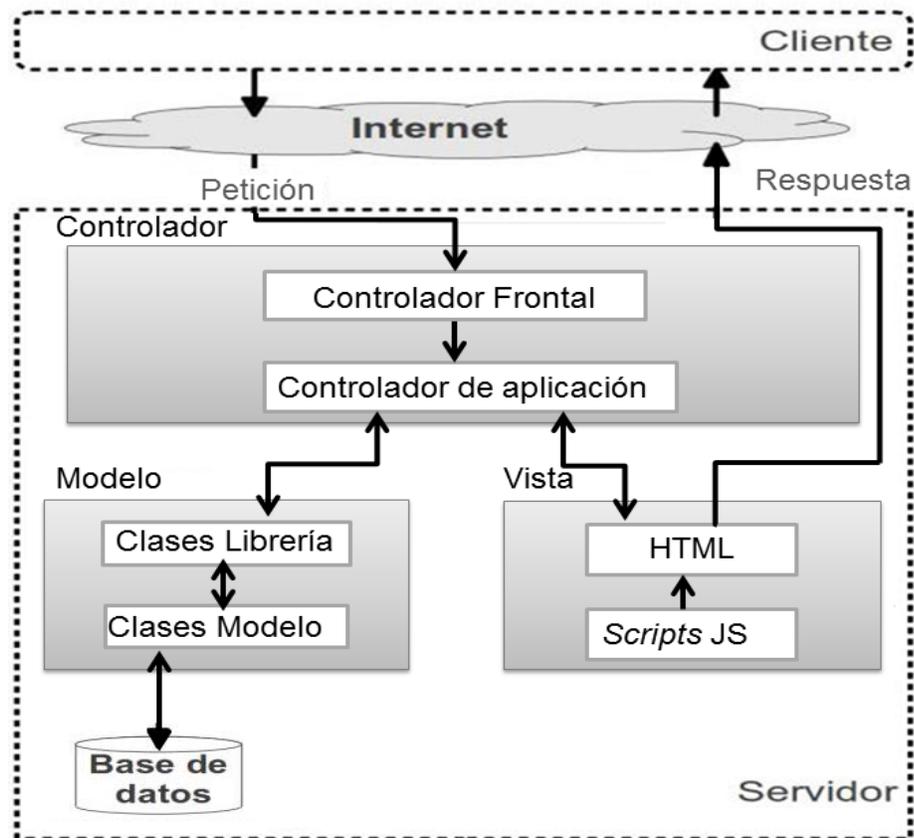


Figura 5. Funcionamiento del MVC en GUUD.

Ventajas de utilizar este patrón arquitectónico:

- Diseño modular y poco acoplado favoreciendo la reutilización.
- Mayor cohesión ya que cada elemento del patrón está altamente especializado en su tarea, la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo del negocio.
- Las vistas proveen mayor flexibilidad y agilidad pues se pueden crear múltiples vistas de un modelo, las vistas pueden anidarse y sincronizarse.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.



Patrones de diseño

Los patrones de diseño expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software. Existen varios tipos de patrones de diseños entre los cuales se encuentran los patrones GoF y GRASP los cuales se caracterizan a continuación:

Patrones de la "Pandilla de los Cuatro" (*Gang-of-Four*)

Los patrones que se presentan proceden de *Design Patterns*, un libro básico que presenta 23 patrones que son útiles durante el diseño de objetos. Puesto que el libro fue escrito por cuatro autores, estos patrones se conocen como los patrones de la "Pandilla de los Cuatro" o patrones "GoF" (Gamma, y otros, 1994).

Los patrones GoF describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases y la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudar a realizar tareas complejas. Existen tres tipos de patrones: de creación, estructurales y de comportamiento. Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas (Ingeniería del Software) (Craig, 2003).

A continuación se describen los patrones y donde se ponen de manifiestos en la solución:

- **Singleton (Instancia única):** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Este patrón se refleja en las clases controladoras que son instancias únicas. Un ejemplo de la misma se evidencia en la clase controladora Estructura la cual contiene un constructor (`__construct`) que obtiene una instancia del objeto o crea una si no existe.
- **Mediator (Mediador):** Define un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto. Se evidencia en las librerías las cuales son mediadoras entre las clases controladoras y los modelos o acceso a datos. Lo planteado anteriormente se puede reflejar a través de la librería `Estructura_lib` que es la mediadora entre la controladora `Estructura` y el modelo `Tb_destructura_md`.
- **Observer (Observador):** Define una dependencia de uno a muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él. Se refleja en la clase `loader` que es el objeto load de las clases controladoras cuya función es cargar los elementos del marco de trabajo dígase librerías, modelos y se encarga de actualizar la controladora instanciada.



GRASP: Patrones de Principios Generales para Asignar Responsabilidades

GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades).

Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. Es un sistema orientado a objetos se compone de objetos que envían mensajes a otros para que lleven a cabo las operaciones requeridas (Visconti, y otros, 2004).

Los patrones de asignación de responsabilidades GRASP dan la medida de un refinamiento del diseño.

- **Experto:** Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Se evidencia en la solución en las clases librerías, que son las que cuentan con la información necesaria para cumplir la responsabilidad sobre los elementos del negocio. Un ejemplo de ello queda demostrado en la librería Estructura_lib encargada de gestionar toda la lógica del negocio de las estructuras.
- **Creador:** el patrón creador permite identificar quién debe ser el responsable de la instanciación de nuevos objetos o clases. Este patrón se utilizó para identificar qué clase A debe crear elementos de una clase B, apoyándose en que la clase A debería: contener, agregar, registrar, utilizar y tener los datos de inicialización de la clase B. Se evidencia en la clase loader que es el objeto load de las clases controladoras, esta clase se encarga de cargar los elementos del marco de trabajo díganse librerías, modelos.
- **Controlador:** Permite asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Se evidencia en las clases controladoras que se encargan de obtener datos, enviarlos a las librerías y a las vistas. La clase Estructura es un ejemplo de controladora que obtiene información y los envía a la librería Estructura_lib y las vistas listar_estructura_view, detalles_estructura_view, entre otras.
- **Bajo acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. El grado de acoplamiento no puede considerarse aisladamente de otros principios como Experto y Alta Cohesión. Sin embargo, es un factor a considerar cuando se intente mejorar el diseño. Ejemplo de ello es la librería Estructura_lib la cual está encargada de gestionar la información referente a las estructuras.
- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. La propia implementación de CodeIgniter contiene los patrones bajo acoplamiento y alta cohesión nivelados pues permite el uso de los componentes de forma individual evidenciando el bajo acoplamiento, así



como la dependencia entre ellos o alta cohesión. Un ejemplo de ello se evidencia en la librería Estructura_lib, ya que hace uso de los métodos de la librería Categoría_estructura_lib para obtener las categorías a las que puede pertenecer una estructura.

2.7 Modelo de la base de datos

Los sistemas de bases de datos almacenan información y permiten a los usuarios recuperarla y actualizarla en base a sus peticiones, por lo que es de gran importancia el diseño de la base de datos. Un sistema de base de datos es un sistema computarizado para mantener información de un individuo o una organización y hacer que esté disponible cuando se solicite. Se puede considerar a la base de datos como una especie de archivo electrónico. Un modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Permite describir las estructuras de datos, las restricciones de integridad y las operaciones de manipulación de los datos (Date, y otros, 2001).

Para los sistemas informáticos siempre es de gran ventaja contar con una base de datos, ya que ofrece a los usuarios un control centralizado de sus datos. Para el componente de Estructura y composición del SGU la base de datos fue modelada con la herramienta Visual Paradigm 8.0. (Ver Figuras 6, 7 y 8).

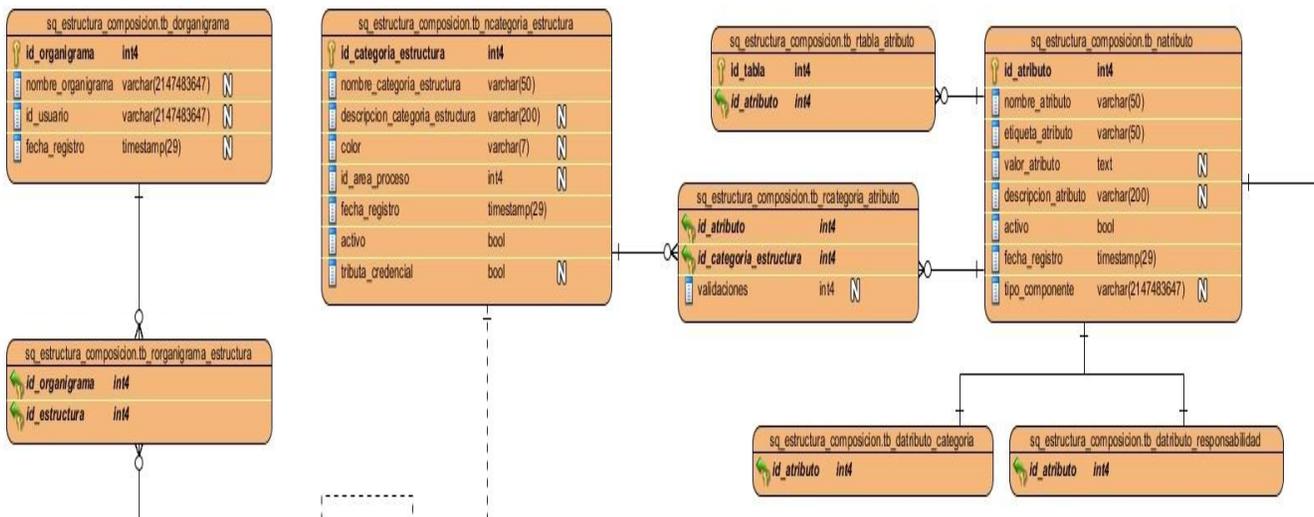


Figura 6. Modelo de datos físico (Parte1).

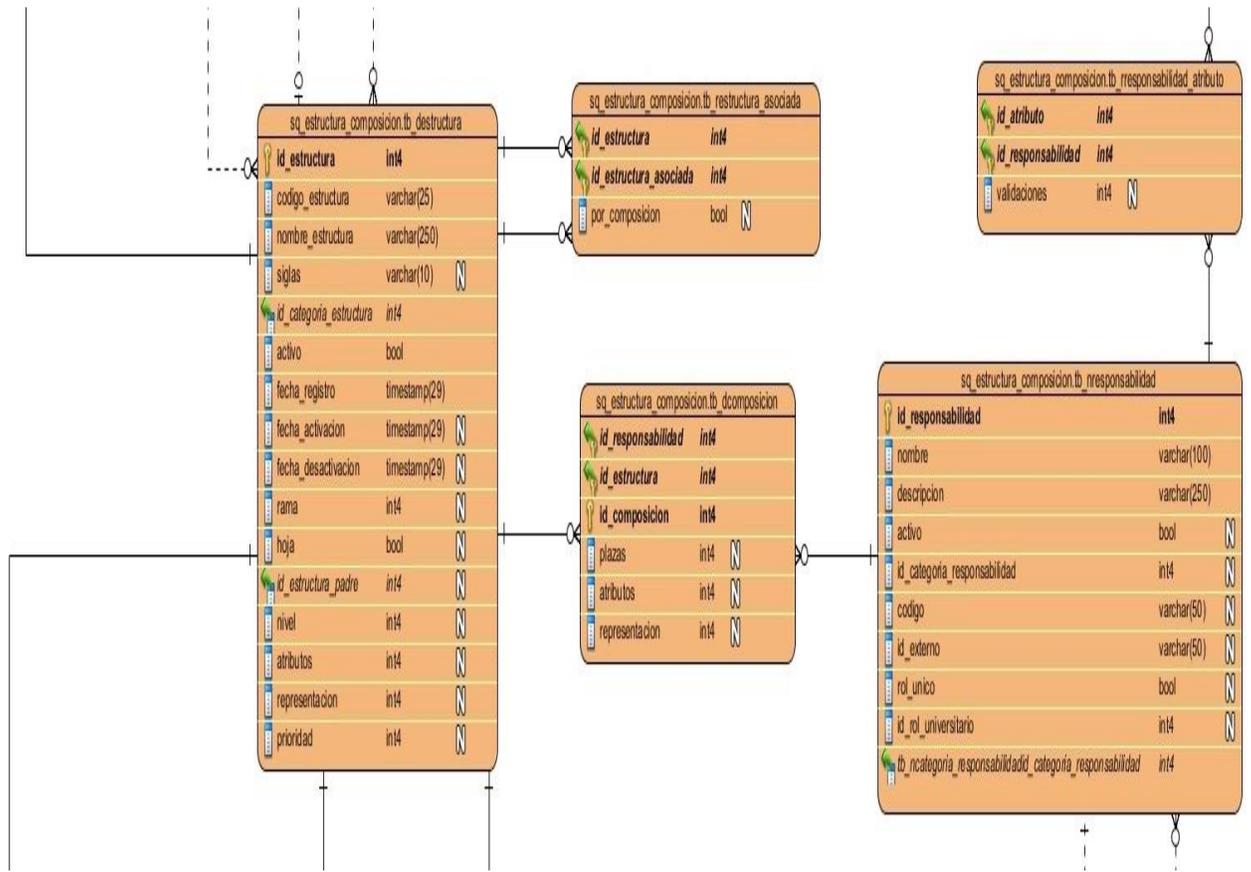


Figura 7. Modelo de datos físico (Parte 2).

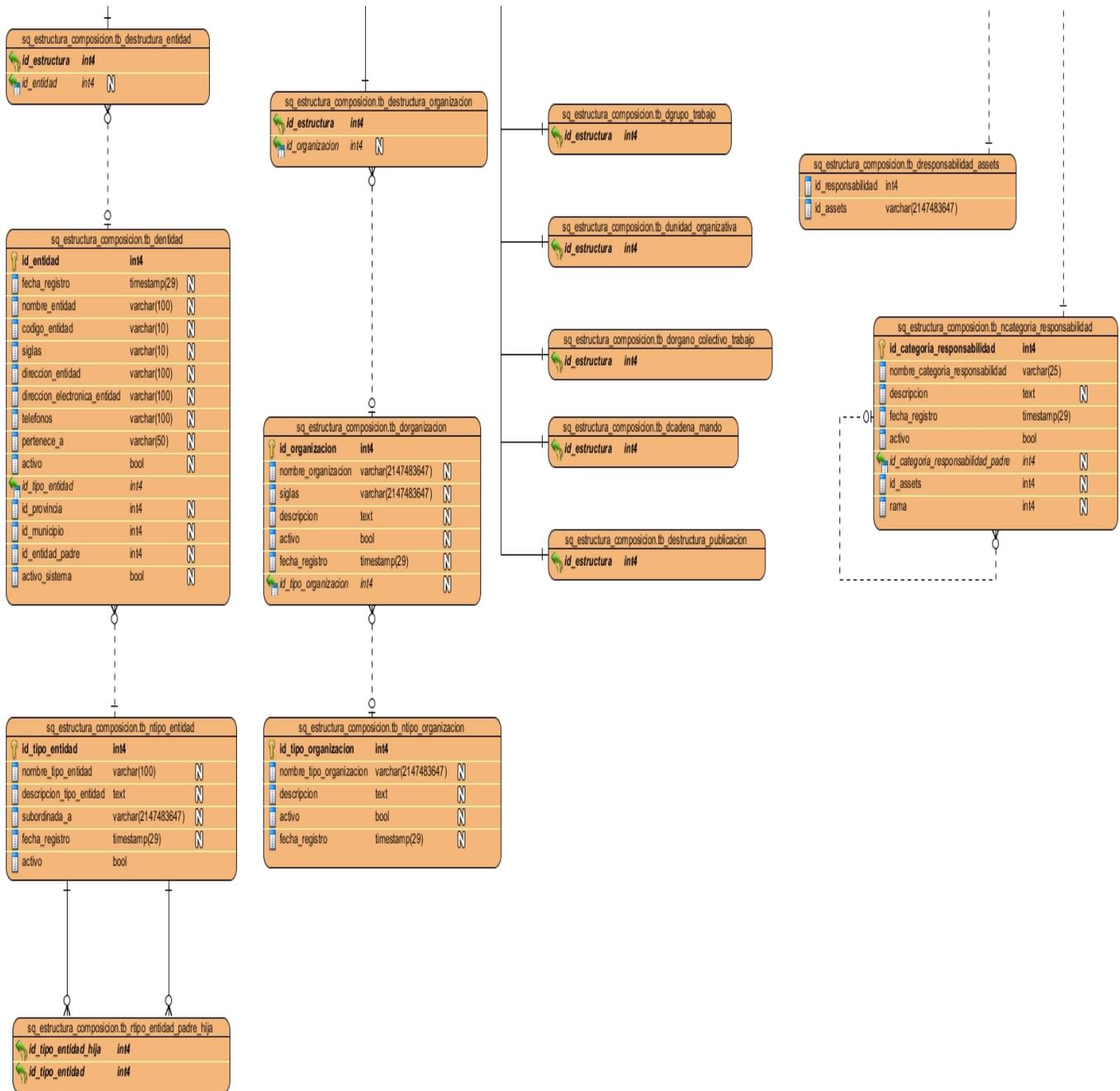


Figura 8. Modelo de datos físico (Parte 3).



2.8 Diagrama de despliegue

El diagrama de despliegue muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria (Grau, y otros, 2001).

A continuación se muestra el diagrama de despliegue correspondiente a la solución desarrollada.

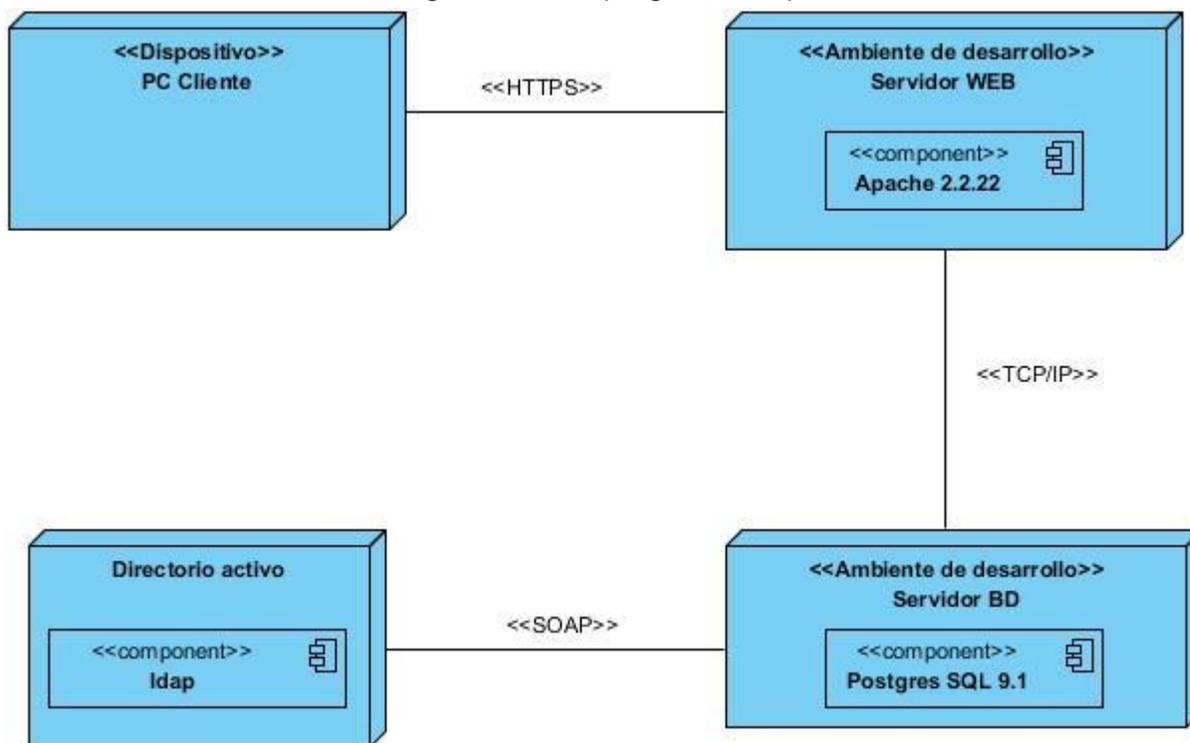


Figura 9. Diagrama de despliegue.

Nodos y enlaces de comunicación	Descripción
PC Cliente	Ordenador desde donde los usuarios acceden a la aplicación. Tiene como función principal acceder al sistema e interactuar con el mismo según sus necesidades.
Servidor Web	Servidor Web Apache es donde descansa la capa de presentación del sistema, la cual es accedida por los ordenadores clientes a través de un navegador web. Funciona como intermediario entre la



	PC cliente y el servidor que guarda la información (servidor de base de datos).
Servidor de BD	Servidor de base de datos es el encargado de almacenar toda la información generada por el sistema, en él se encuentra ubicada la base de datos de donde los servicios que brinda el componente se nutren de información.
Ldap	Permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
HTTPS	Protocolo de transferencia de hipertexto seguro, por sus siglas en inglés, <i>Hypertext Transfer Secure Protocol</i> (HTTPS), es un protocolo de red basado en HTTP por lo que está orientado a transacciones sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores y sigue el esquema petición-respuesta entre un cliente y un servidor.
TCP/IP	TCP/IP es la base de Internet y sirve para enlazar computadoras que utilizan diferentes sistemas operativos. Familia de protocolos utilizada para la conexión entre el servidor web y el servidor donde se encuentra ubicada la base de datos.
SOAP	Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Tabla 4. Elementos del diagrama de despliegue.

2.9 Conclusiones parciales

Las técnicas de obtención de requisitos permitieron la definición de los requisitos funcionales y no funcionales del sistema. Además, la obtención del modelo de la base de datos permitió conocer cómo está estructurada la información almacenada. Por otro lado la realización del modelo de despliegue ilustra la comunicación de los distintos componentes en los cuales se divide el sistema.



3 Construcción y validación de la propuesta de solución

Introducción

El presente capítulo demuestra los aspectos relacionados con la implementación y las pruebas realizadas al componente Estructura y composición. Se utilizan métodos y técnicas para la realización de las pruebas con el propósito de validar la propuesta de solución, la evaluación de su ejecución y los resultados obtenidos. El proceso de pruebas está dirigido a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos exigidos por el cliente y si posee la calidad requerida.

3.1 Paradigmas de programación

Un paradigma de programación define la forma de pensar y los métodos utilizados para construir un programa o subprograma. Ningún paradigma resuelve por sí solo todos los problemas que se presentan en el desarrollo de un sistema.

Programación orientada a objetos (POO):

Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Los objetos son entidades que combinan estado (atributo), comportamiento (método) e identidad (Wang, 2000).

La programación orientada a objetos es una forma de programar que trata de encontrar una solución a problemas. Introduce nuevos conceptos entre los que se destacan los siguientes:

Conceptos	Descripción
Clase	Definiciones de las propiedades y comportamiento de un tipo de objeto concreto.
Herencia	Es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C.
Abstracción	Denota las características esenciales de un objeto donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar, cambiar su estado y "comunicarse" con otros objetos en el sistema sin revelar cómo se implementan estas características. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella se puede llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.
Polimorfismo	Comportamientos diferentes asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener



	objetos de diferentes tipos y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado.
Encapsulamiento	Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción.

Tabla 5. Elementos de la programación orientada a objetos.

En la presente investigación la POO se evidencia en las clases controladoras, librerías y modelos. Lo dicho anteriormente se puede ejemplificar mediante la modelo Tb_destructura_md1 la cual hereda de la clase MY_Model. Además, contiene los atributos y métodos para interactuar con la entidad Estructura de la base de datos entre los que se encuentran obtenerEstructurasDadoldEstructuraPadre, obtenerTipoEstructuraDadoldEstructura y los métodos heredados obtenerPorPáginas, obtenerDadold, entre otros.

Programación orientada a eventos (POE):

La POE es un paradigma de programación en el que el flujo del programa está determinado por eventos o mensajes desde otros programas o hilos de ejecución. Es la base de lo que se denomina interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o componentes de núcleos. Un programa orientado a eventos debe haber sido creado en un lenguaje de programación orientado a objetos, ya que cada objeto espera algún evento que realice el usuario sobre él (Pavón, 2012).

El creador de un programa dirigido por eventos debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador del evento. Los eventos soportados estarán determinados por el lenguaje de programación utilizado por el sistema operativo e incluso por eventos creados por el mismo programador (Pavón, 2012).

La POE se evidencia en el marco de trabajo GUUD mediante ficheros JavaScript, estos a su vez requieren de la librería jQuery. Dichos ficheros son los encargados de configurar los componentes visuales y validar las acciones del usuario mediante la ejecución de funciones dada la ocurrencia de distintos eventos en las vistas del sistema. Ejemplo de ello lo constituye el fichero sgu/base/public/js/sistema/estructura_composicion/organigrama/organigrama.js el cual permite configurar la representación de la estructura a través de la función Organigrama y la manipulación de los eventos que realiza el usuario sobre el área de dibujo con el método iniciarEventos.



Programación Orientada a Aspectos (POA):

La POA es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Gracias a la POA se pueden capturar los diferentes conceptos que componen una aplicación en entidades bien definidas de manera apropiada en cada uno de los casos y eliminando las dependencias inherentes entre cada uno de los componentes. De esta forma se consigue razonar mejor sobre los conceptos, se elimina la dispersión del código y las implementaciones resultan más comprensibles, adaptables y reusables (Nieto, 2010).

"Un aspecto es una unidad modular que se disemina por la estructura de otras unidades funcionales. Los aspectos existen tanto en la etapa de diseño como en la de implementación. Un aspecto de diseño es una unidad modular del diseño que se entremezcla en la estructura de otras partes del diseño. Un aspecto de programa o de código es una unidad modular del programa que aparece en otras unidades modulares del programa." (Nieto, 2010).

El Sistema Gestión Universitaria define aspectos que pueden ser usados por todos sus sistemas como son: el aspecto auth para requerir que el usuario esté autenticado en el sistema cuando acceda a una funcionalidad, el aspecto beginTrans para crear una nueva transacción en el momento de registrar o modificar nuevos datos, el aspecto commitTrans que completa la operación y failedTrans para el caso en el que ocurre algún fallo.

3.2 Estándares de codificación

Usar estándares de codificación y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para el desarrollo del software. Un estándar de codificación permite conservar el código fuente entendible y fácil de mantener, además de mejorar la forma en la que se programa. En la construcción de la solución se utilizan los estándares definidos por el Departamento de Desarrollo de la Dirección de Informatización para el SGU.

3.2.1 Identación, tamaño de las líneas, llaves de apertura y cierre

Usar una indentación sin tabulaciones, con un equivalente a 4 espacios para mantener integridad en las revisiones svn⁵. El uso de las llaves "{" será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres para mantener la legibilidad del código. Un ejemplo del uso de estándar se puede observar en la clase Estructura_lib del fichero sgu/base/application/sistema/estructura_composicion/libraries/estructura_lib.php. Como se muestra a continuación:

⁵ Sistema de control de versiones.



```
....public function hijosArbol($idNodo, $idEstructura)
....{
    ....//
....}
```

Figura 10. Identación.

3.2.2 Convención de nomenclatura

Variables: se rigen por la nomenclatura **camelCase**. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. A continuación un ejemplo del uso de variables en la clase Estructura_lib.

```
....$ordenarPor
....$hijos
```

Figura 11. Variables.

Clases: siempre comienzan con mayúscula, en caso de nombre compuesto las palabras se separan con guión bajo “_” y el resto en minúscula. En la declaración de la clase Categoria_estructura_lib del fichero sgu/base/application/sistema/estructura_composicion/libraries/categoria_estructura_lib.php.

```
class Categoria_estructura_lib
{
    ....//
}
```

Figura 12. Clases.

Funciones: se rigen por la nomenclatura **camelCase**. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa. La función hijosArbol de la clase Estructura_lib es un ejemplo del uso de esta nomenclatura.

```
....public function hijosArbol($idNodo, $idEstructura)
```

Figura 13. Funciones.

Ficheros: todo siempre en minúscula y en caso de nombres compuestos se usa el guión bajo, seguido del sufijo definido para cada tipo de fichero.

- **Vistas:** intuitivo y relacionado con el formulario y/o vista que representa. Sufijo “_view”.
 - Ejemplo: registrar_estructura_view



- **Modelos:** mismo nombre de la tabla de la base de datos para la cual se crea. Sufijo “_mdl”.
 - Ejemplo: tb_destructura_md1
- **Librerías:** mismo nombre de la clase que representa. Sufijo “_lib”.
 - Ejemplo: estructura_lib

Controladoras: mismo nombre de la clase que representa.

3.2.3 Estructuras de control

Las estructuras de control incluyen **if**, **for**, **foreach**, **while**, **switch**. Entre las estructuras de control y los paréntesis debe de existir un espacio. Se recomienda utilizar siempre llaves de apertura y cierre, incluso en situaciones en las que técnicamente son opcionales. Con esto se aumenta la legibilidad del código y se disminuye la probabilidad de errores lógicos. A continuación un ejemplo de su uso en la clase Estructura_lib.

```
$estructurasPadre = array();

switch ($tipo) {
    case '1':
        foreach ($estructuras as $estructura)
        {
            if (!in_array($estructura->id_estructura, $oct)) {
                $valor = $estructura;
                $valor->codigo_nombre = $estructura->codigo_estructura;

                if ($activo == false
                    || ($estructura->activo == 't'
                        && $activo == true))
                {
                    $estructurasPadre[] = $valor;
                }
            }
        }
        break;
    case '2': {
    case '3': {
    case '4': {
    default: break;
}
}
```

Figura 14. Estructuras de control.

3.2.4 Documentación

Todos los archivos deben de tener la documentación asociada al mismo. Para esto debe de cumplir con el siguiente bloque al principio de cada clase:



```
/**
 * Clase librería Estructura_lib. Esta clase funciona como intermediaria
 * entre la controladora y las modelos
 *
 * PHP 5.3
 *
 * @category Librería
 * @package Estructura y composición
 * @author Julio César Hernández González, Elaimy Arias Lismé
 */
```

Figura 15. Documentación de una clase.

```
/**
 * Obtener el criterio por el cual se van a ordenar las estructuras
 *
 * @return string
 * @author Julio César Hernández González, Elaimy Arias Lismé
 */
public function obtenerCriterioDeOrden() {
```

Figura 16. Documentación de funciones.

3.3 Algoritmo para la generación del organigrama

La estructura de una organización, está definida por una relación jerárquica entre los elementos que la componen. Una de las formas más comunes para representar este tipo de conocimiento es haciendo uso de la estructura de datos árbol. Un árbol es un grafo conexo no dirigido que no contiene ciclos en el cuál se utilizan nodos para representar los elementos de la jerarquía y aristas para las relaciones, el mismo se construye a partir de un nodo raíz el cual puede tener cero o más nodos hijos, ver Figura 17.

Un organigrama es la representación visual de la estructura organizativa como se muestra en la Figura 18, en el cual se representan los elementos con rectángulos y las relaciones con líneas, cada elemento tiene un nivel de prioridad con respecto al padre, los hijos de un elemento en el organigrama pueden estar representados por niveles, de forma horizontal o vertical (Ver Figuras 18, 19 y 20). Por su semejanza con un árbol se utiliza la estructura de datos árbol para representar el organigrama teniendo en cuenta que es necesario diferenciar por niveles de prioridad los hijos de cada nodo, partiendo siempre del nivel 0. Para el caso del árbol de la Figura 17, si B tiene mayor prioridad que C y E con respecto a A se crea un nivel para cada sub-árbol dentro del nodo A, como se representa en la Figura 21.

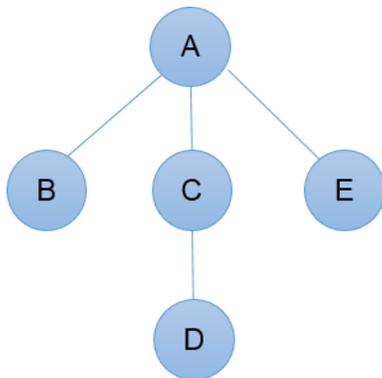


Figura 17. Representación de un árbol (A (B, C (D), E)).

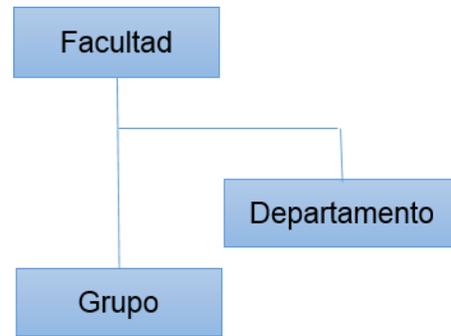


Figura 18. Representación de un organigrama. Hijos representados por niveles.



Figura 19. Representación de un organigrama. Hijos representados horizontalmente.

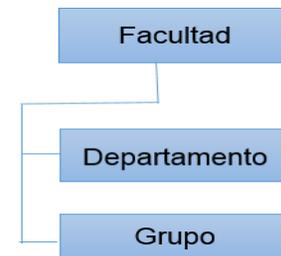


Figura 20. Representación de un organigrama. Hijos representados verticalmente.

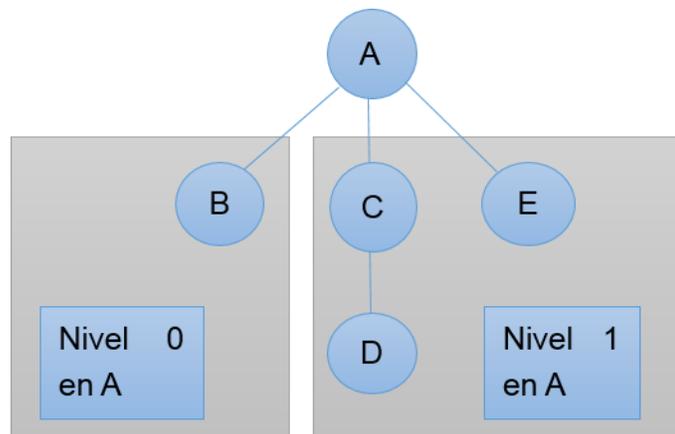


Figura 21. Representación de un árbol con nodos por niveles de prioridad (A ((B), (C ((D), E)))).

Una vez definida la estructura del árbol se utiliza una función para representar el organigrama en una tabla de N filas por M columnas, donde N es igual a la profundidad del organigrama y M a su ancho. Para lograrlo se realizan de forma general tres operaciones, pre-ajuste, ajuste y construcción.

El pre-ajuste consiste en determinar la fila en la que se representará cada nodo así como definir el nodo que se encuentra a la izquierda en su fila. Para ello se realiza un recorrido en profundidad visitando primero el nodo raíz y luego los hijos en dependencia del nivel al que pertenecen. La fila del nodo raíz es la primera, a partir de ello cada nodo hijo pertenece a la fila del nivel en que se encuentra, la fila de cada nivel comienza



en la fila que le continua a la última fila del nivel anterior por lo que si A en el nivel 0 termina en la fila X, entonces el nivel 1 de A comienza en la fila X+1, para el caso en la que los hijos se representen de forma vertical se define una nueva fila para cada uno de los hijos. Al mismo tiempo se determina si el nodo que se visita tiene algún nodo a su izquierda. A continuación se muestra en la Tabla 6 un ejemplo del procedimiento realizado para el árbol de la Figura 21.

Nodo visitado	Nivel	Fila nivel	Nodo padre	Fila	Nodo izquierdo
A	-	-	-	1	-
B	0 de A	2 (Fila del padre es 1)	A	2	-
C	1 de A	3 (Última fila del nivel anterior es 2)	A	3	-
D	0 de C	4 (Fila del padre es 3)	D	4	-
E	1 de A	3 (Última fila del nivel anterior es 2)	A	3	C

Tabla 6. Pasos para realizar pre-ajuste.

Luego de definir la fila de cada nodo y conocer el nodo que está a su izquierda se pasa a realizar el ajuste con el fin de encontrar la columna en la que se representará cada nodo. Para ello se realiza un recorrido en profundidad visitando primero a los hijos y luego el nodo padre. El método consiste en definir la columna de los hijos de un nodo en dependencia del nodo que se encuentre a su izquierda, en caso de no tener, su posición es la primera. En dependencia de la columna del primer hijo y la del último determinar la del padre. Luego se comprueba si es una posición válida teniendo en cuenta los nodos que están a su izquierda, si la posición es menor que la del nodo de su izquierda entonces existe un solapamiento por lo que se corrige la posición del nodo y de sus hijos para que queden a la derecha del nodo izquierdo. Si la representación de los hijos es por niveles se deja una columna libre por cada nivel y es utilizada para representar la línea de conexión entre los diferentes niveles. Para el caso anteriormente mencionado la posición del nodo padre coincide con la de la columna libre. En el caso de que para cada nivel sea distinta la columna se utiliza la mayor y se ajustan los demás niveles con respecto a esta. El procedimiento de ajuste para los resultados obtenidos en el pre-ajuste del ejemplo anterior se muestra en la Tabla 7.

Nodo visitado	Nivel	Nodo izquierdo	Columna sin validar	Columna libre	Columna actual	Columna final
B	0 de A	-	1	1	2	3
El nodo B no presenta nodo izquierdo por lo que su columna sería 1, cuando se valida la columna esta coincide con la columna libre del nivel por lo que se ajusta a la derecha de esta.						
D	0 de C	-	1 (no tiene nodo izquierdo)	-	1	2



El nodo D no presenta nodo izquierdo por lo que su columna sería 1. Este hijo no se representa por niveles por lo que no existe columna libre.						
C	1 de A	-	1 (calculado por hijos)	2	1	2
Los hijos del nodo C se encuentran de la columna 1 a la 1 (1 solo hijo) por lo que su columna es $1+1/2=1$ (en el centro de sus hijos). La columna es válida ya que 1 es menor que 2.						
E	1 de A	C	2 (columna nodo izquierdo 1)	2	3	3
El nodo E tiene un nodo a la izquierda por lo que su columna estaría a la derecha del nodo izquierdo (columna 1). Como la columna 2 no es válida se ajusta a la derecha de la columna libre.						
A	-	-	2 (columna libre mayor)	-	2	2
Dado que los niveles de los hijos de A tienen columnas libres distintas se ajustan los nodos de esos niveles.						

Tabla 7. Pasos para realizar el ajuste.

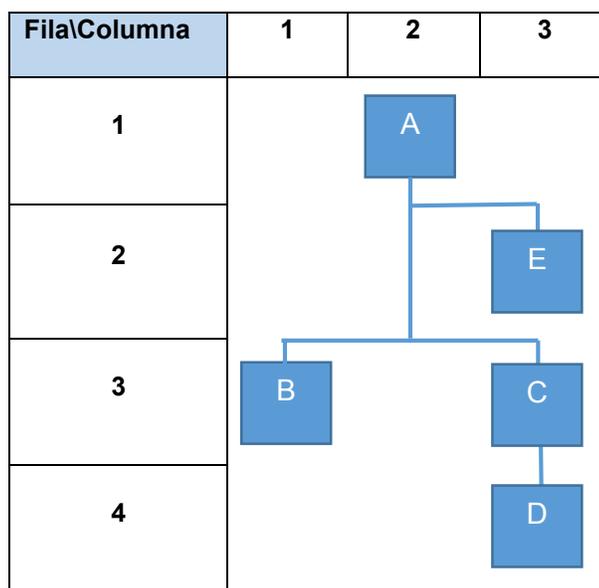


Figura 22. Construcción del organigrama.

El árbol se encuentra ajustado y todos los nodos tienen la fila y la columna en la que se van a representar, por lo que se puede realizar la construcción del organigrama. Para ello se construye la tabla, se recorre el árbol en profundidad y para cada nodo se crea una representación visual en la fila y la columna calculada en los procesos anteriores, el resultado de la construcción del organigrama se muestra en la Figura 22.

3.4 Alternativa para el patrón entidad-atributo-valor

El componente Estructura y composición en su versión 1.0 utiliza el patrón de diseño de base de datos Entidad-Atributo-Valor (EAV) para poder gestionar las estructuras, dado que estas no tienen todos sus atributos definidos. Este patrón es flexible y utiliza entre sus variantes una tabla para gestionar los objetos, una para los atributos y otra para los valores que tiene cada atributo por cada objeto. La propia ventaja de ser flexible se convierte en una desventaja cuando se analiza por rendimiento, crecimiento de la tabla



valores y la complejidad para las consultas. Existen diferentes alternativas para simular el patrón EAV entre las que se encuentra el uso de los tipos de datos json y hstore. Para la versión 2.0 del componente se hace uso del tipo de datos hstore, ya que json solo es viable a partir de la versión 9.3 de PostgreSQL. Hstore es básicamente un campo que permite tener un diccionario con claves y valores, sobre el cual se pueden realizar consultas e índices. Este tipo de dato es parte del paquete adicional de módulos que trae PostgreSQL desde la versión 8.X y proporciona muchas mejoras en tiempo de acceso a la base de datos desde las versiones 9.X. La notación utilizada en este tipo de datos es clave => valor. En la Figura 23 se muestran ejemplos del uso de este tipo de dato para actualizar los datos de una estructura y obtener su información.

```
UPDATE sq_estructura_composicion.tb_destructura
SET atributos = atributos || hstore('IdASSETS', '3')
WHERE id_estructura = '4';

SELECT nombre_estructura, atributos->'IdASSETS'
FROM sq_estructura_composicion.tb_destructura
WHERE id_estructura = '4';
```

Figura 23. Usos de hstore.

A diferencia del patrón EAV el cual necesita realizar varias uniones entre sus tablas para acceder a la información, con hstore se logra centralizar todos los atributos y valores del objeto en un solo campo de la tabla. Además, cuando se desea obtener un listado de todos los objetos con EAV se obtienen varias tuplas representando la información del objeto, mientras que haciendo uso de hstore solo se obtiene una tupla para cada objeto. La alternativa utilizada mejora el acceso a los datos, su flexibilidad y la sencillez en las consultas realizadas.

3.5 Estrategia de pruebas

Las pruebas se enfocan sobre la lógica interna del software y las funciones externas. Las mismas se realizan con la intención de descubrir un error. Además, no pueden asegurar la ausencia de defectos, solo pueden demostrar que existen errores en el software. Las pruebas son aplicadas a diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Existen diferentes tipos de niveles de pruebas entre los que se encuentran:

- **Prueba de Unidad:** las pruebas de unidad tienen como objetivo verificar la unidad más pequeña del diseño del software. Tomando como guía la descripción del diseño y prueba importantes caminos de control para descubrir errores dentro de los límites del componente. Las pruebas de unidad se concentran en la lógica del procesamiento interno y en las estructuras de datos tales como: código



fuentes, archivos binarios, archivos de datos entre otros. Este tipo de prueba se puede aplicar en paralelo a varios componentes (Pressman, 2005).

- **Pruebas del Sistema:** las pruebas del sistema se encargan del comportamiento de un sistema de cómputo. Tienen como propósito principal ejercitar profundamente el sistema de cómputo y evaluar si el producto de software cumple con los requerimientos especificados. La prueba del software es un proceso que puede planearse y especificarse sistemáticamente. El conjunto de pruebas del sistema está conformado por un amplio abanico de pruebas entre las que destacan pruebas funcionales (Pressman, 2005).

Métodos de pruebas

La interfaz del sistema puede ser probada mediante el método de **caja negra**, también denominada prueba de comportamiento, esta se centra en los requisitos funcionales del software. Las estructuras de datos pueden ser probadas mediante el método de **caja blanca** o estructural, que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.

Pruebas de caja blanca o estructural:

La prueba de caja blanca denominada a veces *prueba de caja de cristal* es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca el ingeniero de software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada componente; ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los bucles en sus límites y con sus límites operacionales; y ejerciten las estructuras internas de datos para asegurar su validez (Pressman, 2005).

La técnica de prueba de caja blanca utilizada en la investigación es el camino básico que permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (Pressman, 2005).

Para obtener dicho conjunto de caminos independientes se construye el grafo de flujo asociado y se calcula su complejidad ciclomática.

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. Cuando se usa en el contexto del método de prueba del camino básico, el valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecuta cada sentencia al menos una vez (Pressman, 2005).



Pruebas de caja negra o funcional:

Las pruebas de caja negra también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los detectados por los métodos de caja blanca. La prueba de caja negra intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación (Pressman, 2005).

3.6 Resultados de las pruebas

Pruebas de caja blanca o estructural:

Con el objetivo de verificar el resultado real de la prueba para cada uno de los caminos se empleó un mecanismo que posee CodeIgniter (parte estructural del marco de trabajo, GUUD en este caso) para la automatización de pruebas de unidad y que se describe a continuación. CodeIgniter posee una librería o clase especializada en la ejecución de pruebas estructurales, cuenta con una sola función de evaluación y dos funciones de resultados, permite determinar con certeza si un código específico produce el tipo de dato y resultado esperado. Para ejecutar una prueba utilizando dicha librería es necesario suministrar el código a probar y un resultado esperado como se puede observar en la Figura 24:

```
$this->unit->run($this->registrarEstructura(), true, 'Registrar estructura');
```

Figura 24. Ejecución de una prueba estructural.

El primer parámetro es el segmento de código que se desea probar, luego se especifica el resultado esperado de la evaluación del código (puede ser un tipo de dato o un valor literal) y por último el nombre de prueba el cual es opcional. Los tipos de datos posibles son: “is_string”, “is_bool”, “is_true”, “is_false”, “is_int”, “is_numeric”, “is_float”, “is_double”, “is_array”, “is null”.

En la Tabla 8 se muestra un caso de prueba de caja blanca utilizando la técnica de camino básico:

Prueba estructural de caja blanca	Código caso de prueba: SGU-NUC-010318_DCP_RFEstC30_CrearEstructura
Probador: Julio César Hernández González	
Código al que se aplica:	
<pre>public function registrarEstructura() { \$verificar = array('id_tipo_estructura', 'id_categoria_estructura'); if (\$this->input->is_post_back(\$verificar)) {</pre>	



```

    $all_post = $this->input->all_post();
    if ($this->estructura_lib->registrarEstructura($all_post) != FALSE) {
        return true;
    } else {
        return false;
    }
} else {
    return false;
}
}

```

Complejidad ciclomática:

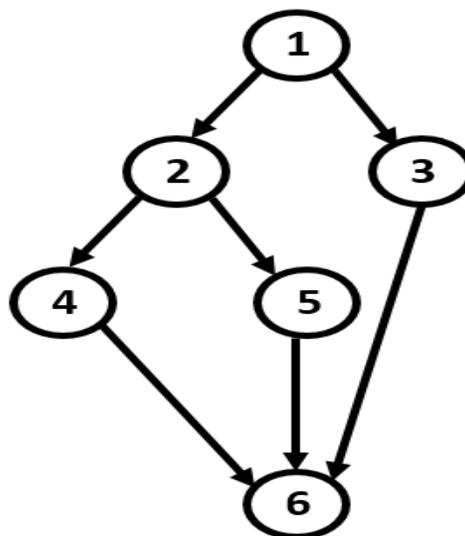
$$V(G)=7 \text{ aristas} - 6 \text{ nodos} + 2 = 3$$

$$V(G)=2 \text{ nodos predicados} + 1 = 3$$

Caminos independientes:

- 1) 1-2-4-6
- 2) 1-2-5-6
- 3) 1-3-6

Representación del grafo:



Función de evaluación:

```

public function pruebaRegistrarEstructura() {
    $_POST['id_estructura_padre_default'] = 3;
    $_POST['id_estructura_entidad_activa'] = 3;
    $_POST['id_tipo_estructura'] = 'unidad_organizativa';
    $_POST['siglas'] = 'EP';
    $_POST['fecha_activacion'] = '09/05/2014';
    $_POST['id_entidad'] = '41';
    $_POST['nombre_estructura'] = 'Estructura prueba';
    $_POST['id_categoria_estructura'] = '11';
    $_POST['activo'] = 't';
    $_POST['id_estructura_padre'] = '151';
    $_POST['nivel'] = '0';
    $_POST['IdASSETS'] = '1';
}

```



```

$_POST['hiddenAtributos'] = '{"IdASSETS": "1"}';
echo $this->unit->run($this->registrarEstructura(),true, 'Registrar estructura');
}
    
```

Resultado:	
Test Name	Registrar estructura
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/estructura_composicion /controllers/estructura.php
Line Number	26
Test Name	Registrar estructura
Estado:	
Satisfactorio	

Tabla 8. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst30_CrearEstructura.

En el Anexo 4 se muestra el resto de los casos de prueba de caja blanca realizados a la solución. Al aplicar la prueba no se encontraron no conformidades, alcanzando resultados satisfactorios en cada una de las iteraciones.

Pruebas de caja negra o funcional:

Para la realización de las pruebas funcionales se generaron los artefactos “Diseño de casos de pruebas basados en requerimientos”. Se generó un documento por cada requerimiento funcional del sistema donde se recogen todos los datos necesarios para probar la interfaz. El objetivo del diseño de casos de prueba es que sean efectivos descubriendo defectos en los programas y muestren que el sistema satisface sus requerimientos. En la Tabla 9 se muestra un caso de prueba de caja negra:

Escenario	Descripción	Respuesta del sistema	Flujo central
"EC 1.1 Insertar Datos correctamente"	Mediante este escenario se inserta en el sistema una nueva estructura.	Una vez creada la estructura se muestra el mensaje “El elemento ha sido creado satisfactoria mente”	Al autenticarse el usuario el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona “Estructura y Composición” en la barra de íconos de los procesos horizontales. El sistema muestra las opciones de menú y el usuario selecciona en la agrupación funcional "Estructura" la funcionalidad "Estructura". El sistema muestra un listado con las estructuras existentes. El usuario selecciona la opción “Crear” en la barra de íconos flotantes. El sistema brinda la posibilidad de llenar todos los campos, además de dar la opción de Aceptar y Cancelar.



			Además, brinda la posibilidad de Vincular a otra estructura. Se llenan los datos correctamente para crear una estructura. Se selecciona el botón Aceptar y se muestra el mensaje “El elemento ha sido creado satisfactoriamente. Se presiona el botón Aceptar y se muestra el árbol de estructuras actualizado.
EC 2.2 Insertar elemento repetido.	Mediante este escenario se introducen datos para insertar una estructura que ya existe.	El sistema muestra un mensaje de error “El elemento ya existe”.	Al autenticarse el usuario el sistema muestra por defecto todos los subsistemas que están contenidos dentro. Una vez autenticado el usuario en el sistema, selecciona “Estructura y Composición” en la barra de íconos de los procesos horizontales. Se selecciona en la derecha la agrupación funcional Estructura la funcionalidad “Estructura”. El sistema muestra un listado de estructuras. El usuario selecciona la opción “Crear” en la barra de íconos flotantes. El sistema brinda la posibilidad de llenar todos los campos, además de dar la opción de Aceptar y Cancelar. Se llenan los datos para crear una estructura con elementos repetidos (Nombre). Se selecciona la opción de Aceptar y se muestra el mensaje de error “El elemento ya existe”.

Tabla 9. Caso de prueba de caja negra.

Para ver los diseños de casos de pruebas en su totalidad consultar el expediente del proyecto SGU (Núcleo). Al aplicar la prueba de caja negra se alcanzaron los siguientes resultados en cada una de las iteraciones, como se muestra en la Tabla 10. Las no conformidades encontradas fueron resueltas en su totalidad.

Iteración	Cantidad de no conformidades	Asociadas a
Primera	11	Errores de interfaz, de validación y ortografía.
Segunda	3	Errores de interfaz y de validación
Tercera	0	-

Tabla 10. Relación de no conformidades por iteración.



3.7 Conclusiones parciales

La utilización de los estándares de codificación permitió que el código estuviera bien estructurado y organizado en la solución desarrollada. El algoritmo desarrollado para la generación del organigrama es factible para su construcción. Las pruebas realizadas permitieron demostrar que la aplicación cumple con todos los requerimientos dados por los usuarios finales y que satisface sus necesidades. Además, demostraron que las funciones implementadas producen un resultado satisfactorio, cuestión que evidencia la fiabilidad y eficiencia del componente Estructura y composición.



Conclusiones generales

Al término de la presente investigación se arribaron a las siguientes conclusiones:

- El estudio realizado a diferentes sistemas para la gestión de estructuras organizativas permitió identificar la incorporación de nuevas funcionalidades que constituyeron la propuesta de solución.
- Con la aplicación del proceso de desarrollo de software que combina las metas y prácticas de las áreas de procesos del nivel 2 de CMMI con las buenas prácticas de la dirección y desarrollo ágil de proyectos de software, se logró obtener y especificar los requisitos, así como elaborar los artefactos propuestos y el diseño de la solución.
- El desarrollo de la versión 2.0 del componente Estructura y composición del SGU permite:
 - Aumentar el nivel funcional del componente Estructura y composición lo que posibilita la creación de diferentes unidades organizativas como políticas y de masa así como vincular a otra estructura.
 - Asignar a una estructura o varias estructuras una cantidad de plazas determinadas o temporales, o miembros de una organización.
 - Gestionar un registro de los diferentes cambios que pueden ocurrir en la estructura organizacional de la entidad.
 - Representar la estructura organizacional de la entidad en un organigrama, así como exportar el organigrama en formato jpeg.
 - Configurar la forma de ordenar la información de las estructuras, las distintas categorías y estructuras que se muestran en el Árbol de Estructura.
- Las pruebas realizadas a la aplicación permitieron garantizar el cumplimiento de los requerimientos, así como demostrar la calidad del componente realizado.



Recomendaciones

Se recomienda perfeccionar e incorporar nuevas funcionalidades para lograr mejoras superiores en la gestión de la estructura organizacional de las entidades, entre las cuales se pueden considerar las siguientes:

- Adicionar a la estructura las relaciones de coordinación y de asesoría para mejorar la comprensión del organigrama.
- El montaje de las ayudas como guías para el usuario.
- Extender el uso de la aplicación para la gestión de la estructura organizacional en otras organizaciones o universidades debido a sus potencialidades y flexibilidad.
- Utilizar las potencialidades de HTML5 haciendo uso de webgl para la construcción del organigrama.
- Hacer uso del tipo de dato json en vez de hstore como alternativa para del patrón EAV.



Bibliografía referenciada

- Alvarez, Miguel Angel. 2009.** DesarrolloWeb.com. *DesarrolloWeb.com*. [En línea] 2009. <http://www.desarrolloweb.com/articulos/modelo-vista-controlador-codeigniter.html>.
- Andrade, Simón. 2005.** *Diccionario de Economía*. Tercera . Buenos Aire : Andrade, 2005. pág. 448.
- Carrasco, Lilibeth, Francia, Briceño y Flireny, Marin. 2009.** *El sistema organizacional*. Barcelona : s.n., 2009.
- Ciberaula. 2010.** Ciberaula. *Una Introducción a Apache*. [En línea] 2010. Una Introducción a Apache.
- Cinergix. 2013.** creately. [En línea] 2013. [www.creately .com](http://www.creately.com).
- Corporation, Oracle. 2013.** NetBeans IDE. *NetBeans IDE*. [En línea] 2013.
- Craig, Larman. 2003.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. [ed.] Prentice Hall. Segunda edición. 2003. 970-17-0261-1.
- Date, Chris J., Ruiz Faudón, Luis Sergio y María. 2001.** *Introducción a los sistemas de bases de datos*. s.l. : Pearson Educación, 2001.
- Díaz, Vilela, Luis. 1998.** *Psicología del Trabajo y las Organizaciones*. La Laguna : Concepto, Historia y Método, 1998.
- Franklin, Enrique B. 2004.** *Organización de Empresas*. Segunda. s.l. : Mc Graw Hill, 2004. pág. 78.
- Gamma, Eric, y otros. 1994.** *Design patterns for object-oriented software development*. Urbana-Champaign : Springer, 1994.
- García, Vidal, Yanio, y otros. 2013.** *SISTEMA INTEGRAL DE GESTIÓN UNIVERSITARIA*. 2013.
- Grau, Ferré, Xavier y Sánchez Segura, María Isabel. 2001.** *Desarrollo orientado a objetos con UML*. [ed.] Tercera. 2001.
- Group, Business Rules. 2014.** Business Rules Buzz. [En línea] 2014. <http://www.businessrulesgroup.org/home-brg.shtml>.
- Guerra, Sotillo, Alexei. 2007.** *Cambios, Organización y Entorno*. Universidad de Málaga : Eumed, 2007. ISBN: 84-689-2425-3.
- Home - Pencil Project. 2012.** Home - Pencil Project. *Home - Pencil Project*. [En línea] 2012. <http://pencil.evolus.vn/>.
- Ingeniería del Software. Patrones del "Gang of Four"*. Facultad de informática - Universidad Politécnica de Madrid : s.n.
- Insperty Business Services. 2014.** Insperty OrgPlus. [En línea] 2014. <http://www.orgplus.com/>.



LibrosWeb. 2014. *Introducción a CSS.* 2014.

Lucid Software. 2014. Lucidchart. [En línea] 2014. <https://www.lucidchart.com/>.

Microsoft, Corporation. 2014. Visio. [En línea] 2014. <http://office.microsoft.com/es-mx/visio/>.

Nieto, Moreno, Juan Manuel. 2010. *Introducción a la Programación Orientada a Aspectos.* Universidad de Sevilla, España : s.n., 2010.

Pavón, Santiago. 2012. *Programación Orientada a Eventos.* 2012.

PostgreSQL administration and management tools. 2013. pgAdmin: PostgreSQL administration and management tools. [En línea] 2013. <http://www.pgadmin.org/>.

Pressman, Roger S. . 2005. *Ingeniería del Software: Un Enfoque Práctico.* 6ta. s.l. : Mc Graw Hill, 2005. pág. 980.

Psicología Online. 2014. Psicología Online. *Clínica Psicológica V. Mars. Sueca.* [En línea] 2014. Valencia, España. <http://www.psicologia-online.com/pir/concepto-de-estructura-organizacional.html>.

Ramos, Leticia O.C., Ferrell, Geoffrey, Hirt y Adriae. 2004. *Introducción a los Negocios en un Mundo Cambiante.* Cuarta. s.l. : Mc Graw-Hill Interamericana, 2004. pág. 243. 9701039424.

Sánchez, Méndez, Alelí. 2013. *Proceso de desarrollo de software. Metodología DAC.* Departamento de Desarrollo de la Dirección de Informatización, Universidad de las Ciencias Informáticas. 2013. pág. 87. DIN-DD-PDS-DAC.

SmartDraw. 2014. SmartDraw. [En línea] 2014. www.smartdraw.com/.

Sommerville, Ian. 2005. *Ingeniería de Software.* Séptima edición. Madrid : Pearson Educación, 2005.

The PHP Group. 2013. The PHP Group. *php manual.* [En línea] 2013. <http://us.php.net/manual/en/preface.php>.

The PostgreSQL Global Development Group. 2014. PostgreSQL. *PostgreSQL.* [En línea] 2014. <http://www.postgresql.org/>.

Visconti, Marcello y Astudillo, Hernán. 2004. *Patrones de Diseño. Fundamentos de Ingeniería de Software.* 2004.

W3C. 2013. W3C. *W3C HTML.* [En línea] 2013. <http://www.w3.org/html/>.

Wang, Paul S. . 2000. *Java con Programación orientada a Objetos y aplicaciones en la www.* s.l. : International Thomson, 2000.

WEB, MAESTROS DEL. 2007. ¿Qué es JavaScript? [En línea] 2007. [Citado el: febrero 7, 2014.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.



Bibliografía Consultada

- Aguilar García, Adrián, y otros. 2009.** *Guía para la elaboración de organigramas.* 2009.
- Alvarez, Miguel Angel. 2009.** DesarrolloWeb.com. *DesarrolloWeb.com.* [En línea] 2009. <http://www.desarrolloweb.com/articulos/modelo-vista-controlador-codeigniter.html>.
- Andrade, Simón. 2005.** *Diccionario de Economía.* Tercera . Buenos Aire : Andrade, 2005. pág. 448.
- Bours Castelo, Eduardo, y otros. 2009.** *Guía para la elaboración de organigramas.* 2009.
- Carrasco, Lilibeth, Francia, Briceño y Flireny, Marin. 2009.** *El sistema organizacional.* Barcelona : s.n., 2009.
- Ciberaula. 2010.** Ciberaula. *Una Introducción a Apache.* [En línea] 2010. Una Introducción a Apache.
- Cinergix. 2013.** creately. [En línea] 2013. [www.creately .com](http://www.creately.com).
- Corporation, Oracle. 2013.** NetBeans IDE. *NetBeans IDE.* [En línea] 2013.
- Craig, Larman. 2003.** *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.* [ed.] Prentice Hall. Segunda edición. 2003. 970-17-0261-1.
- Date, Chris J., Ruiz Faudón, Luis Sergio y María. 2001.** *Introducción a los sistemas de bases de datos.* s.l. : Pearson Educación, 2001.
- Departamento de Soporte Jurídico. 2010.** *Guía Técnica para Elaborar Organigramas.* 2010.
- Díaz, Vilela, Luis. 1998.** *Psicología del Trabajo y las Organizaciones.* La Laguna : Concepto, Historia y Método, 1998.
- Enriquez Caro, Ricardo. 2012.** Administración Moderna. [En línea] Noviembre 9, 2012.
- Franklin, Enrique B. 2004.** *Organización de Empresas.* Segunda. s.l. : Mc Graw Hill, 2004. pág. 78.
- Gamma, Eric, y otros. 1994.** *Design patterns for object-oriented software development.* Urbana-Champaign : Springer, 1994.
- García, Vidal, Yanio, y otros. 2013.** *SISTEMA INTEGRAL DE GESTIÓN UNIVERSITARIA.* 2013.
- Grau, Ferré, Xavier y Sánchez Segura, María Isabel. 2001.** *Desarrollo orientado a objetos con UML.* [ed.] Tercera. 2001.
- Group, Business Rules. 2014.** Business Rules Buzz. [En línea] 2014. <http://www.businessrulesgroup.org/home-brg.shtml>.
- Guerra, Sotillo, Alexei. 2007.** *Cambios, Organización y Entorno.* Universidad de Málaga : Eumed, 2007. ISBN: 84-689-2425-3.



- Hernández León, Rolando Alfredo y Coello González, Sayda. 2011.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. Ciudad de La Habana : Universitaria, 2011. pág. 110. ISBN 978-959-16-1307-3.
- Home - Pencil Project. 2012.** Home - Pencil Project. *Home - Pencil Project*. [En línea] 2012. <http://pencil.evolus.vn/>.
- Ingeniería del Software. Patrones del "Gang of Four"*. Facultad de informática - Universidad Politécnica de Madrid : s.n.
- Insperty Business Services. 2014.** Insperty OrgPlus. [En línea] 2014. <http://www.orgplus.com/>.
- LibrosWeb. 2014.** *Introducción a CSS*. 2014.
- Lucid Software. 2014.** Lucidchart. [En línea] 2014. <https://www.lucidchart.com/>.
- Microsoft, Corporation. 2014.** Visio. [En línea] 2014. <http://office.microsoft.com/es-mx/visio/>.
- Nieto, Moreno, Juan Manuel. 2010.** *Introducción a la Programación Orientada a Aspectos*. Universidad de Sevilla, España : s.n., 2010.
- Pavón, Santiago. 2012.** *Programación Orientada a Eventos*. 2012.
- PCC. 2011.** Partido Comunista de Cuba. [En línea] 2011. http://www.pcc.cu/i_historia.php.
- PostgreSQL administration and management tools. 2013.** pgAdmin: PostgreSQL administration and management tools. [En línea] 2013. <http://www.pgadmin.org/>.
- Pressman, Roger S. . 2005.** *Ingeniería del Software: Un Enfoque Práctico*. 6ta. s.l. : Mc Graw Hill, 2005. pág. 980.
- Psicología Online. 2014.** Psicología Online. *Clínica Psicológica V. Mars. Sueca*. [En línea] 2014. Valencia, España. <http://www.psicologia-online.com/pir/concepto-de-estructura-organizacional.html>.
- Ramos, Leticia O.C., Ferrell, Geoffrey, Hirt y Adriae. 2004.** *Introducción a los Negocios en un Mundo Cambiante*. Cuarta. s.l. : Mc Graw-Hill Interamericana, 2004. pág. 243. 9701039424.
- Room, Instituto Quintana. 2011.** *GUÍA TÉCNICA PARA LA ELABORACIÓN Y AUTORIZACIÓN DE ESTRUCTURAS ORGANICAS Y ORGANIGRAMAS*. 2011.
- Sánchez, Méndez, Alelí. 2013.** *Proceso de desarrollo de software. Metodología DAC*. Departamento de Desarrollo de la Dirección de Informatización, Universidad de las Ciencias Informáticas. 2013. pág. 87. DIN-DD-PDS-DAC.
- SmartDraw. 2014.** SmartDraw. [En línea] 2014. www.smartdraw.com/.
- Sommerville, Ian. 2005.** *Ingeniería de Software*. Séptima edición. Madrid : Pearson Educación, 2005.

BIBLIOGRAFÍA CONSULTADA •



- The PHP Group. 2013.** The PHP Group. *php manual*. [En línea] 2013. <http://us.php.net/manual/en/preface.php>.
- The PostgreSQL Global Development Group. 2014.** PostgreSQL. *PostgreSQL*. [En línea] 2014. <http://www.postgresql.org/>.
- UCI. 2014.** Universidad de las Ciencias Informáticas. *Mision*. [En línea] 2014. <http://www.uci.cu/mision>.
- Visconti, Marcello y Astudillo, Hernán. 2004.** Patrones de Diseño. *Fundamentos de Ingeniería de Software*. 2004.
- W3C. 2013.** W3C. *W3C HTML*. [En línea] 2013. <http://www.w3.org/html/>.
- Wang, Paul S. . 2000.** *Java con Programación orientada a Objetos y aplicaciones en la www*. s.l. : International Thomson, 2000.
- WEB, MAESTROS DEL. 2007.** ¿Qué es JavaScript? [En línea] 2007. [Citado el: febrero 7, 2014.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.



Glosario de términos

Base de datos: Conjuntos de datos interrelacionados entre sí y su fin es servir a uno o varios usuarios, sin redundancias perjudiciales e innecesarias, es independiente de la aplicación que la utilice y tiene operaciones específicas.

Código abierto (en idioma inglés *open source*): es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

Centralización: Tendencia a concentrar autoridad y restringir su delegación.

Desconcentración: Acción mediante la cual el responsable de una organización delega facultades en un órgano que no cuenta con personalidad jurídica y patrimonio propio para que cumpla con una función.

IDE de desarrollo: Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Inversión de Control (IoC): La Inversión de Control es un patrón de diseño pensado para permitir un menor acoplamiento entre componentes de una aplicación y fomentar así el reúso de los mismos.

Página web: Una página web es un documento creado en formato de lenguaje de marcas de hipertexto, por sus siglas en inglés, (*Hypertext Markup Language*, HTML) que es parte de un grupo de documentos hipertexto o recursos disponibles en el WWW. Una serie de páginas web componen lo que se llama un sitio web.

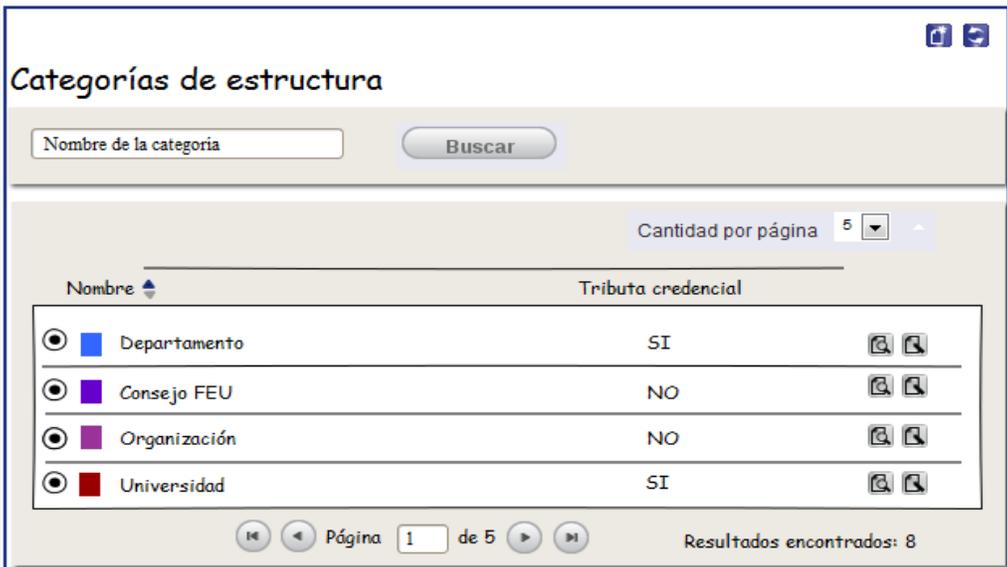
Responsabilidad: Asignación de una tarea a una persona o unidad administrativa, que obligatoriamente deben realizar.

World Wide Web (WWW o W3C): (literalmente telaraña de alcance mundial), cuya traducción podría ser "Red Global Mundial" o "Red de Amplitud Mundial", es un sistema de documentos de hipertextos enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia y navega a través de ellas usando enlaces.

Anexos

Anexo 1: Descripción de requisitos

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC1	Listar categorías de estructura	<ul style="list-style-type: none"> • El requisito permite listar categorías de estructura, donde las categorías engloban diversas estructuras que tengan características similares. El usuario selecciona del componente Estructura y composición del SGU, la opción Categoría del menú de funcionalidades Estructura. • El usuario introduce en el filtro de búsqueda el nombre de la categoría de estructura que desea mostrar y selecciona la opción Buscar. • Si el usuario no introduce el nombre de una categoría de estructura en el filtrado de búsqueda, el sistema mostrará un listado con todas las categorías de 	Baja	Media

		estructura existentes, con los datos: Nombre, Tributa credencial y el Estado.	
Prototipo			
			
	Campos	Tipos de Datos	Reglas o Restricciones
	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura
	<ul style="list-style-type: none"> Tributa credencial 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Las categorías de estructura se ordenan en la lista por orden alfabético. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. 	

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC2	Crear categoría estructura	<p>9. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página.</p> <ul style="list-style-type: none"> • El requisito permite crear categoría de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Categoría del menú de funcionalidades Estructura. • El usuario selecciona la opción crear e introduce el Nombre de la categoría de estructura y la Descripción, selecciona el Estado, el Color y si Tributa credencial. • Si el usuario no selecciona ningún color, se creará la categoría de estructura de color negro. • El usuario puede seleccionar atributos de estructura así como validar los atributos asociados. • El usuario puede crear otra categoría de estructura o volver al listado el cual se actualizará. 	Alta	Alta
Prototipo				

☰

Crear categoría de estructura

Categoría de estructura: *

Descripción:

Color:



Activo
 Tributa credencial

Atributos de estructura:

Atributos asociados

Atributo	
A1	⚙️ 🗑️
B1	⚙️ 🗑️
C1	⚙️ 🗑️

✕

Validar atributo

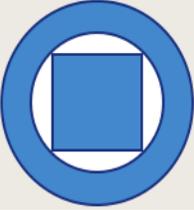
Validación:

Validación	Valor	
Campo requerido	true	🗑️
Longitud mínima	2	🗑️

	Campos	Tipos de Datos	Reglas o Restricciones
	<ul style="list-style-type: none"> Categoría de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. No se admiten: <ul style="list-style-type: none"> 1-% 2-' 3-espacios en los límites de la palabra
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Tributación credencial 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Color 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los 	

		<p>caracteres permitidos aquellos definidos por el dominio válido.</p> <ol style="list-style-type: none"> 4. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente." 5. En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". 6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". 8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo. 9. El Estado siempre se inicia Activo. 10. Cuando el usuario asocia una validación a un atributo se comprobará que dicha validación sea excluyente de las existentes o de las asociadas al atributo mostrando un mensaje de confirmación "Se eliminarán las validaciones excluyentes". 		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC3	Modificar categoría estructura	<ul style="list-style-type: none"> • El requisito permite modificar categoría de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Categoría del menú de funcionalidades Estructura. 	Alta	Alta

		<ul style="list-style-type: none"> • El usuario selecciona la opción modificar e introduce el Nombre de la categoría de estructura y la Descripción, selecciona el Estado (Activo/Desactivo), el Color y si Tributa credencial. • Si el usuario no selecciona ningún color, se mantendrá el color de la categoría de estructura. Además, si no modifica el estado, la categoría de estructura conservará el estado anterior. • El usuario puede seleccionar agregar atributos de estructura o remover los atributos. Ambas opciones se pueden hacer atributo por atributo, o todos los atributos a la vez. • Si el usuario guarda los cambios hechos, se muestra el listado de categorías de estructura actualizado. Si el usuario cancela la modificación de la categoría, se mostrará el listado de categorías de 		
--	--	---	--	--

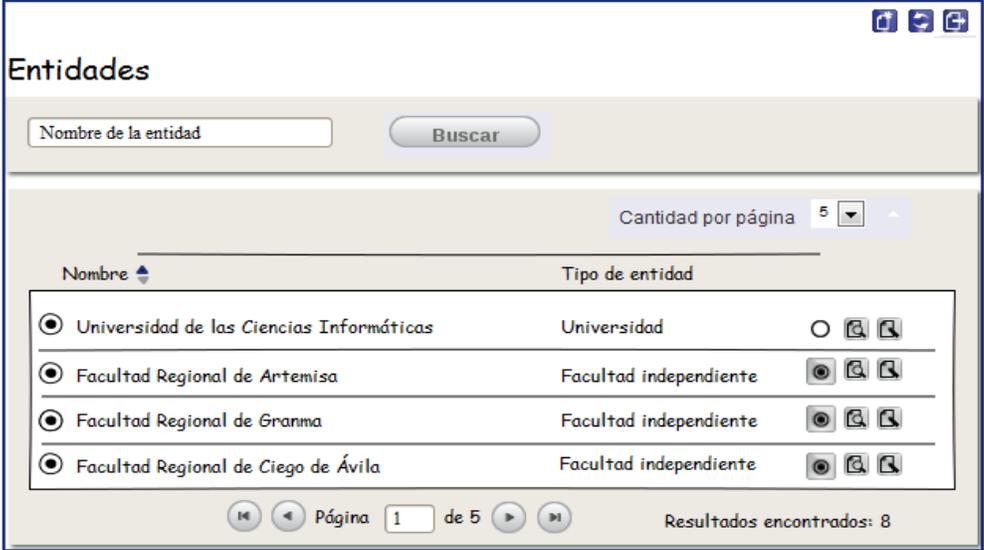
		estructura.								
Prototipo										
<div style="border: 1px solid black; padding: 10px;"> <p>Modificar categoría estructura</p> <p>Categoría de estructura: * <input type="text"/></p> <p>Descripción: <input type="text"/></p> <p>Color:  <input type="text" value="#000000"/></p> <p><input checked="" type="checkbox"/> Activo</p> <p><input type="checkbox"/> Tributa credencial</p> <hr/> <p>Atributos de estructura: *</p> <p><input type="text"/> 2 disponibles <input type="button" value="Agregar todos"/> <input type="text"/> 2 disponibles <input type="button" value="Remover todos"/></p> <table border="1" style="width: 100%;"> <tr> <td>ASSET</td> <td style="text-align: center;">+</td> <td style="text-align: center;">▲</td> </tr> <tr> <td>Finalidad</td> <td style="text-align: center;">+</td> <td style="text-align: center;">■</td> </tr> </table> <p style="text-align: right;"><input type="button" value="Guardar"/> <input type="button" value="Cancelar"/></p> </div>					ASSET	+	▲	Finalidad	+	■
ASSET	+	▲								
Finalidad	+	■								
	Campos	Tipos de Datos	Reglas o Restricciones							
	<ul style="list-style-type: none"> Categoría de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. 							

			<ul style="list-style-type: none"> No se admiten: <ul style="list-style-type: none"> 1-% 2-' 3-espacios en los límites de la palabra
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Tributacional 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Color 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente." En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas" 	

		<p>para una palabra" encima el campo que debe ser llenado correctamente.</p> <p>7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?".</p> <p>8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p> <p>9. Cuando el usuario asocia una validación a un atributo se comprobará que dicha validación sea excluyente de las existentes o de las asociadas al atributo mostrando un mensaje de confirmación "Se eliminarán las validaciones excluyentes".</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC4	Ver detalles de categoría estructura	<ul style="list-style-type: none"> • El requisito permite ver detalles de categoría estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Categoría del menú de funcionalidades Estructura. • El usuario selecciona la opción ver detalles de la categoría de estructura y el sistema muestra una ventana emergente con los datos: Categoría de estructura, Descripción, Tributa credencial, Estado, Color y Atributos de estructura. Para volver al 	Baja	Baja

		listado se selecciona la opción cerrar.		
Prototipo				
				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Categoría de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Tributa credencial 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Color 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Atributos de 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	

	estructura			
	Observaciones	<p>10. El usuario debe estar autenticado en el sistema.</p> <p>11. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción.</p>		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFestC5	Listar entidades	<ul style="list-style-type: none"> • El requisito permite listar entidades. El usuario selecciona del componente Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración. • El usuario introduce en el filtro de búsqueda el nombre de la entidad que desea mostrar y selecciona la opción Buscar. • Si el usuario no introduce el nombre de una entidad en el filtrado de búsqueda, el sistema mostrará un listado con todas las entidades existentes, con los datos: Nombre, Tipo de entidad y el Estado. 	Baja	Media
Prototipo				

				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección 	
	Observaciones	<p>12. El usuario debe estar autenticado en el sistema.</p> <p>13. Las entidades se ordenan en la lista por orden alfabético.</p> <p>14. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario.</p> <p>15. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página.</p>		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC6	Crear entidades	<ul style="list-style-type: none"> El requisito permite crear entidades. El usuario selecciona del componente 	Media	Alta

		<p>Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración.</p> <ul style="list-style-type: none"> • El usuario selecciona la opción crear e introduce el Tipo de entidad, Código, Nombre, Siglas, Subordinado a, Teléfono, Provincia, Municipio, Dirección, Dirección electrónica y el Estado. • Si el usuario desea que la entidad no esté activa debe Desactivarla, pues siempre se inicia Activo. • El usuario puede crear otra entidad o volver al listado el cual se actualizará. 		
Prototipo				

Entidades		
<div style="border: 1px solid black; padding: 5px;"> <p>Tipo de entidad: * <input type="text" value="-Seleccione-"/></p> <p>Código: * <input type="text"/></p> <p>Nombre: * <input type="text"/></p> <p>Siglas: * <input type="text"/></p> <p>Subordinado a: <input type="text" value="-Seleccione-"/></p> <p>Teléfono: * <input type="text"/></p> <p>Provincia: * <input type="text" value="-Seleccione-"/></p> <p>Municipio: * <input type="text" value="-Seleccione-"/></p> <p>Dirección: * <input type="text"/></p> <p>Dirección electrónica: * <input type="text"/></p> <p><input checked="" type="checkbox"/> Activo</p> <p style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Código 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. No se admiten: <ol style="list-style-type: none"> % ' espacios en los límites de la palabra Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio

			<ul style="list-style-type: none"> • Único • Alfanumérico, admite guiones bajos y espacios entre palabras.
	<ul style="list-style-type: none"> • Siglas 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras. • Admite cualquier tipo de carácter. • Admite de 1 a 6 caracteres.
	<ul style="list-style-type: none"> • Subordinado a 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Teléfono 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Admite solo números. • No admiten: <ol style="list-style-type: none"> 1. % 2. ' 3. Espacios 4. guiones bajos 5. letras
	<ul style="list-style-type: none"> • Provincia 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Municipio 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Dirección 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite

			<p>guiones bajos y espacios entre palabras.</p> <ul style="list-style-type: none"> • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> • Dirección electrónica 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Admite solo direcciones válidas. • No admite cualquier tipo de carácter.
	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
Observaciones		<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo “Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 4. Cuando el usuario introduce un carácter extraño en un campo que solo acepta números el sistema muestra un mensaje en rojo “Entre solo números”. 5. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. 6. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 7. Si se introduce más de la cantidad de caracteres por palabra 	

		<p>definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente.</p> <p>8. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?".</p> <p>9. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p> <p>10. Cuando el usuario introduce un carácter extraño en un campo que solo acepta direcciones electrónicas el sistema muestra un mensaje en rojo "Entre un email válido."</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFestC7	Modificar entidades	<ul style="list-style-type: none"> El requisito permite modificar entidades. El usuario selecciona del componente Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración. El usuario selecciona la opción modificar e introduce el Tipo de entidad, Código, Nombre, Siglas, Subordinado a, Teléfono, Provincia, Municipio, Dirección, Dirección electrónica y el Estado. 	Media	Alta

		<ul style="list-style-type: none"> • Si el usuario no modifica algún campo, la entidad conservará los datos anteriores. • Si el usuario guarda los cambios hechos se muestra el listado de entidades actualizado. Si el usuario cancela la modificación de la entidad, se mostrará el listado de entidades. 		
--	--	---	--	--

Prototipo

Entidades

Tipo de entidad: * <input type="text" value="-Seleccione-"/>	Código: * <input type="text"/>	Nombre: * <input type="text"/>
Siglas: * <input type="text"/>	Subordinado a: <input type="text" value="-Seleccione-"/>	Teléfono: * <input type="text"/>
Provincia: * <input type="text" value="-Seleccione-"/>	Municipio: * <input type="text" value="-Seleccione-"/>	Dirección: * <input type="text"/>
Dirección electrónica: * <input type="text"/>	<input checked="" type="checkbox"/> Activo	

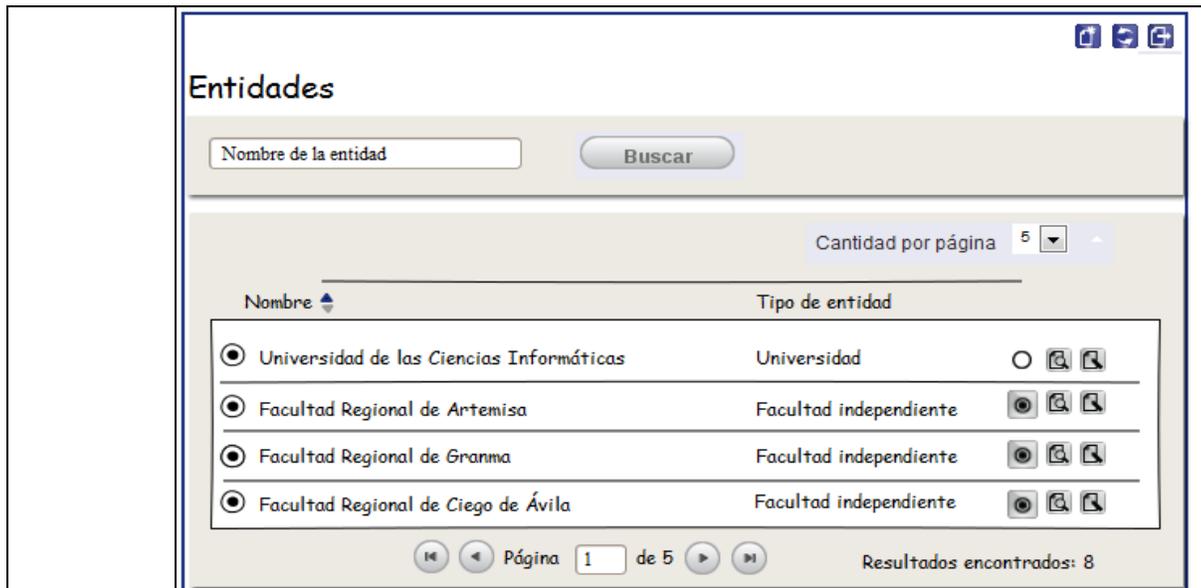
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> • Tipo de entidad 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Selección

	<ul style="list-style-type: none"> • Código 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras. • No se admiten: <ol style="list-style-type: none"> 4. % 5. ' 6. espacios en los límites de la palabra • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> • Nombre 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras.
	<ul style="list-style-type: none"> • Siglas 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras. • Admite cualquier tipo de carácter. • Admite de 1 a 6 caracteres.
	<ul style="list-style-type: none"> • Subordin 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección

	ado a		
	<ul style="list-style-type: none"> • Teléfono 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Admite solo números. • No admiten: <ol style="list-style-type: none"> 6. % 7. ' 8. Espacios 9. guiones bajos 10. letras
	<ul style="list-style-type: none"> • Provincia 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Municipio 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Dirección 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras. • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> • Dirección electrónica 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Admite solo direcciones válidas. • No admite cualquier tipo de carácter.
	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección

	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo “Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 4. Cuando el usuario introduce un carácter extraño en un campo que solo acepta números el sistema muestra un mensaje en rojo Entre solo números”. 5. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido modificado satisfactoriamente.”. 6. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 7. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 8. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”. 9. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo. 10. Cuando el usuario introduce un carácter extraño en un campo que solo acepta direcciones electrónicas el sistema muestra un mensaje en rojo “Entre un email válido.”.
--	---------------	---

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC8	Establecer entidad activa	<ul style="list-style-type: none"> • El requisito permite establecer entidad activa. El usuario selecciona del componente Estructura y Composición del SGU, la opción Entidad del menú de funcionalidades Configuración. • El usuario introduce en el filtro de búsqueda el nombre de la entidad que desea activar y selecciona la opción Buscar. • Si el usuario no introduce el nombre de una entidad en el filtrado de búsqueda, el sistema mostrará un listado con todas las entidades existentes, con los datos: Nombre, Tipo de entidad y el Estado. El usuario selecciona la entidad que desea establecer como entidad activa del sistema. 	Baja	Alta
Prototipo				



Campos	Tipos de Datos		Reglas o Restricciones	
• Nombre	• Varchar	• Solo lectura		
• Tipo de entidad	• Varchar	• Solo lectura		
• Estado	• Bool	• Selección		
Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. Las entidades se ordenan en la lista por orden alfabético. 3. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. 4. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página. 			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC9	Ver detalles de entidad	<ul style="list-style-type: none"> • El requisito permite ver detalles de entidad. El usuario selecciona del 	Baja	Media

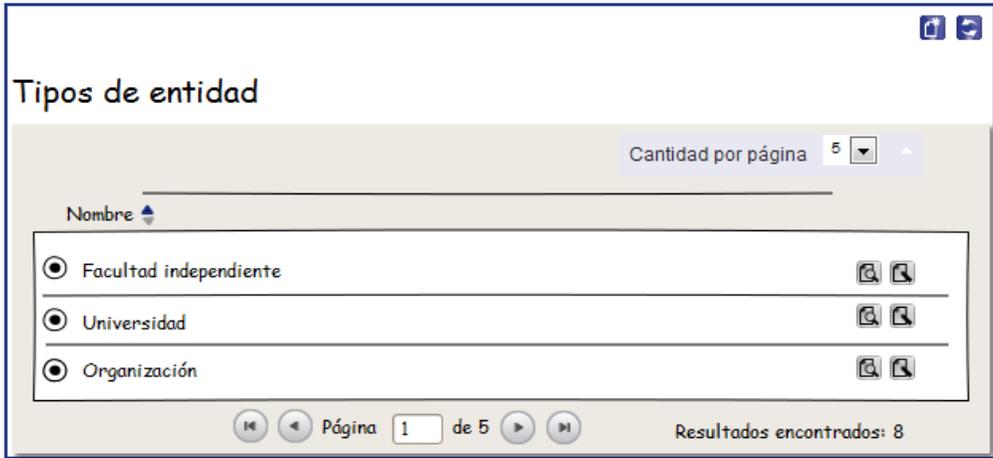
		<p>componente Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración.</p> <ul style="list-style-type: none"> El usuario selecciona la opción ver detalles de la entidad y el sistema muestra una ventana emergente con los datos: Tipo de entidad, Código, Nombre, Siglas, Subordinado a, Teléfono, Provincia, Municipio, Dirección, Dirección electrónica y el Estado. Para volver al listado se selecciona la opción cerrar. 		
--	--	--	--	--

Prototipo

Detalles de entidad X

Tipo de entidad: Universidad
Código: 104
Nombre: Facultad Regional de Ciego de Ávila
Siglas: FRCA
Subordinado a:
Provincia: Ciego de Ávila
Municipio: Ciego de Ávila
Dirección: Carretera San Juan
Dirección electrónica: uci@uci.cu
Teléfono: 033225689
Estado: Activo

	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Código 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Siglas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Subordinado a 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Teléfono 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Provincia 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Municipio 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Dirección 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Dirección electrónica 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción. 		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC10	Listar tipo de entidad	<ul style="list-style-type: none"> El requisito permite listar tipo de entidad. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de entidad del menú de 	Baja	Media

		<p>funcionalidades</p> <p>Configuración.</p> <ul style="list-style-type: none"> El sistema muestra un listado con los tipos de entidad existentes con el Nombre, Estado y acciones a realizar sobre el elemento. 		
Prototipo				
				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Los tipos de entidad se ordenan en la lista por orden alfabético. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página. 		

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC11	Crear tipo de entidad	<ul style="list-style-type: none"> • El requisito permite crear tipo de entidad. El usuario selecciona del componente Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración. • El usuario selecciona la opción crear e introduce el Tipo de entidad, Descripción Código y el Estado. • Si el usuario desea que el tipo de entidad no esté activa debe Desactivarla, pues siempre se inicia Activo. • El usuario puede seleccionar las entidades a la cual se subordinará el Tipo de entidad. Además, puede remover dichas entidades. Ambas opciones se pueden hacer entidad por entidad, o todas las entidades a la vez. 	Media	Alta

		<ul style="list-style-type: none"> El usuario crea el tipo de entidad y se muestra el listado de los tipos de entidad actualizado. 											
Prototipo													
<div style="border: 1px solid #ccc; padding: 10px;"> <h3>Crear tipo de entidad</h3> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Tipo de entidad: *</p> <input type="text"/> </div> <div style="width: 45%;"> <p>Descripción:</p> <div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div> </div> </div> <div style="margin-top: 10px;"> <input checked="" type="checkbox"/> Activo </div> <div style="margin-top: 10px; border: 1px solid #ccc; padding: 5px;"> <p>Puede subordinarse a:</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <input type="text"/> 2 disponibles Agregar todos </div> <div style="width: 45%;"> <input type="text"/> 0 disponibles Remover todos </div> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Facultad independiente</td> <td style="text-align: center; padding: 2px;">+</td> <td style="width: 10px;"></td> </tr> <tr> <td style="padding: 2px;">Universidad</td> <td style="text-align: center; padding: 2px;">+</td> <td style="width: 10px;"></td> </tr> <tr> <td style="height: 30px;"></td> <td></td> <td></td> </tr> </table> </div> </div>					Facultad independiente	+		Universidad	+				
Facultad independiente	+												
Universidad	+												
	Campos	Tipos de Datos	Reglas o Restricciones										
	<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre 										

			<p>palabras.</p> <ul style="list-style-type: none"> No se admiten: <ul style="list-style-type: none"> 1-% 2-' 3-espacios en los límites de la palabra Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> Activo 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<p>Observaciones</p> <ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente." En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado 		

		<p>correctamente.</p> <p>7. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”.</p> <p>8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC12	Modificar tipo de entidad	<ul style="list-style-type: none"> • El requisito permite modificar tipo de entidad. El usuario selecciona del componente Estructura y composición del SGU, la opción Entidad del menú de funcionalidades Configuración. • El usuario selecciona la opción modificar e introduce el Tipo de entidad, Descripción Código y el Estado. • Si el usuario desea que el tipo de entidad no esté activa debe Desactivarla, pues siempre se inicia Activo. • El usuario puede seleccionar las entidades a la cual se subordinará el Tipo de entidad. Además, 	Media	Alta

		<p>puede remover dichas entidades. Ambas opciones se pueden hacer entidad por entidad, o todas las entidades a la vez.</p> <ul style="list-style-type: none"> • Si el usuario guarda los cambios hechos, se muestra el listado de los tipos de entidad actualizado. Si el usuario cancela la modificación del tipo de entidad, se mostrará el listado de los tipos de entidad. 		
<p>Prototipo</p>				

Modificar tipo de entidad

Tipo de entidad: * Descripción:

Activo

Puede subordinarse a:

2 disponibles 0 disponibles

Facultad independiente <input type="button" value="+"/>	▲		▲
Universidad <input type="button" value="+"/>	■		■

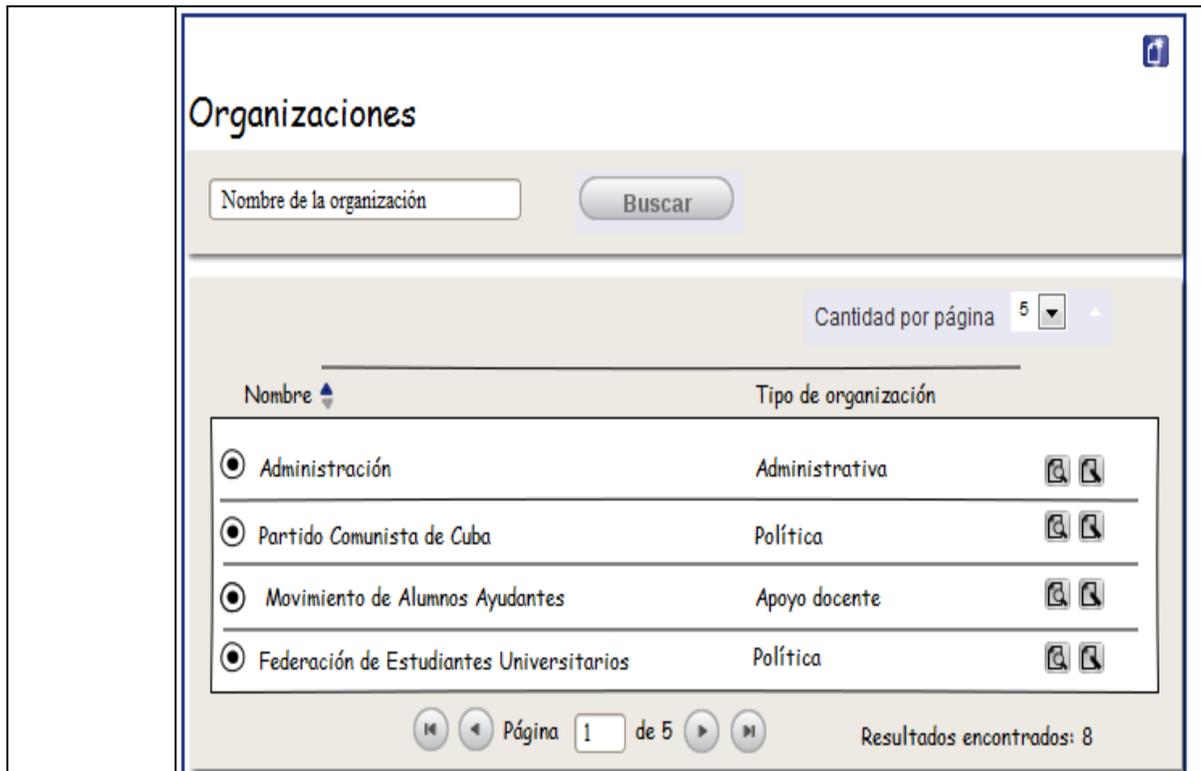
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. No se admiten: <ul style="list-style-type: none"> 1-% 2-'

			<p>3-espacios en los límites de la palabra</p> <ul style="list-style-type: none"> • Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> • Descripción 	<ul style="list-style-type: none"> • Text 	<ul style="list-style-type: none"> • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	<ul style="list-style-type: none"> • Activo 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
Observaciones		<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 4. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido modificado satisfactoriamente." 5. En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". 6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". 	

		8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC13	Ver detalles de tipo de entidad	<ul style="list-style-type: none"> El requisito permite ver detalles de tipo de entidad. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de entidad del menú de funcionalidades Configuración. El usuario selecciona la opción ver detalles de tipo de entidad y el sistema muestra una ventana emergente con los datos: Tipo de entidad, Descripción, Subordinado a y el Estado. Para volver al listado se selecciona la opción cerrar. 	Baja	Media
Prototipo				

<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #cccccc; padding: 2px; display: flex; justify-content: space-between; align-items: center;"> Detalles de tipo de entidad ✕ </div> <div style="padding: 5px;"> <p>Tipo de entidad: Facultad independiente</p> <p>Descripción:</p> <p>Estado: Activo</p> <p>Subordinado a:</p> </div> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Cerrar"/> </div> </div>				
Campos		Tipos de Datos		Reglas o Restricciones
• Tipo de entidad		• Varchar		• Solo lectura
• Descripción		• Text		• Solo lectura
• Subordinado a		• Varchar		• Solo lectura
• Estado		• Bool		• Solo lectura
Observaciones		1. El usuario debe estar autenticado en el sistema. 2. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción.		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC14	Listar organización	<ul style="list-style-type: none"> El requisito permite listar organizaciones. El usuario selecciona del componente Estructura y composición del SGU, la opción Organización del menú de funcionalidades Configuración. 	Baja	Media

		<ul style="list-style-type: none"> • El usuario introduce en el filtro de búsqueda el nombre de la organización que desea mostrar y selecciona la opción Buscar. • Si el usuario no introduce el nombre de una organización en el filtrado de búsqueda, el sistema mostrará un listado con todas las organizaciones existentes, con los datos: Nombre, Estado y Tipo de organización. 		
<p>Prototipo</p>				



Campos	Tipos de Datos		Reglas o Restricciones	
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 		<ul style="list-style-type: none"> Solo lectura 	
<ul style="list-style-type: none"> Tipo de organización 	<ul style="list-style-type: none"> Varchar 		<ul style="list-style-type: none"> Solo lectura 	
Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Las organizaciones se ordenan en la lista por orden alfabético. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página. 			
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente

RFEstC15	Crear organización	<ul style="list-style-type: none"> • El requisito permite crear organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Organización del menú de funcionalidades Configuración. • El usuario selecciona la opción crear e introduce el Nombre, Tipo de organización, Descripción, Siglas y el Estado. • Si el usuario desea que la organización no esté activa debe Desactivarla, pues siempre se inicia Activo. • El usuario puede crear otra organización o volver al listado el cual se actualizará. 	Media	Alta
Prototipo				

Organizaciones		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras.
<ul style="list-style-type: none"> Tipo de organización 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Siglas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. Admite cualquier tipo de carácter. Admite de 1 a 6 caracteres.

	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Descripción 	<ul style="list-style-type: none"> • Text 	<ul style="list-style-type: none"> • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	<p>Observaciones</p>	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo “Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 4. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. 5. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 7. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”. 8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo. 	

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC16	Modificar organización	<ul style="list-style-type: none"> • El requisito permite modificar una organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Organización del menú de funcionalidades Configuración. • El usuario selecciona la opción modificar e introduce el Nombre, Tipo de organización, Descripción, Siglas y el Estado. • Si el usuario no modifica algún campo, la organización conservará los datos anteriores. • Si el usuario guarda los cambios hechos se muestra el listado de organizaciones actualizado. Si el usuario cancela la modificación de la organización, se mostrará el listado de organizaciones. 	Media	Alta

Prototipo		
<div style="border: 1px solid black; padding: 10px;"> <p>Organizaciones ☰</p> <p>Nombre: * <input type="text" value="Administración"/> Tipo de organización: * <input type="text" value="Administrativa"/> Descripción <input type="text"/></p> <p>Siglas: <input type="text" value="Admin"/> <input checked="" type="checkbox"/> Activo</p> <p style="text-align: right;"><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></p> </div>		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras.
<ul style="list-style-type: none"> Tipo de organización 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Siglas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. Admite cualquier tipo de carácter. Admite de 1 a 6

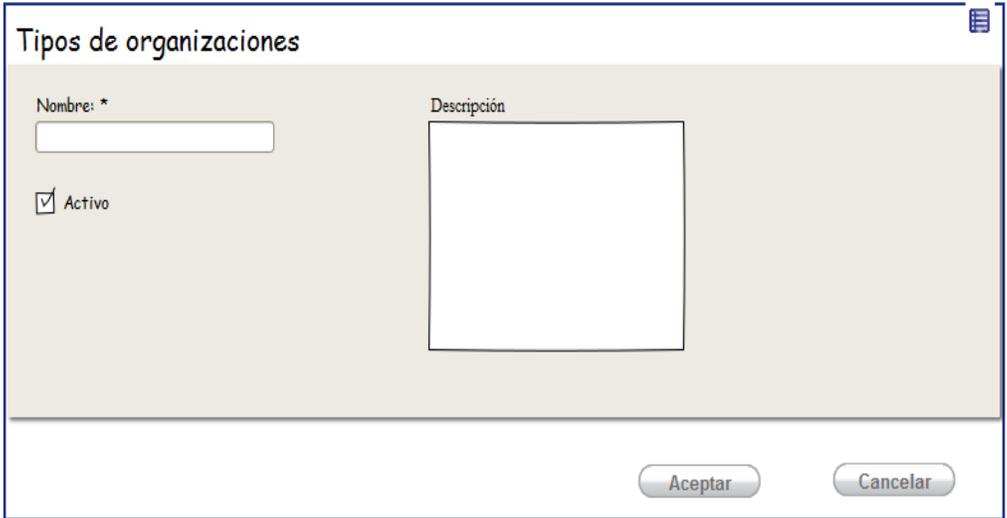
			caracteres.
	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Descripción 	<ul style="list-style-type: none"> • Text 	<ul style="list-style-type: none"> • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 4. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido modificado satisfactoriamente." 5. En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". 6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". 8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no 	

permitirá continuar escribiendo.				
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC17	Ver detalles de organización	<ul style="list-style-type: none"> El requisito permite ver detalles de organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Organización del menú de funcionalidades Configuración. El usuario selecciona la opción ver detalles de organización y el sistema muestra una ventana emergente con los datos: Tipo de organización, Nombre, Siglas, Descripción y el Estado. Para volver al listado se selecciona la opción cerrar. 	Baja	Media
Prototipo				

<div style="border: 1px solid black; padding: 5px;"> <div style="background-color: #e0e0e0; padding: 2px;">Detalles de organización X</div> <p>Tipo de organización: Administrativa Nombre: Administración Siglas: Admin Descripción: Estado: Activo</p> <div style="text-align: right; margin-top: 10px;"> <input type="button" value="Cerrar"/> </div> </div>				
Campos		Tipos de Datos		Reglas o Restricciones
• Nombre		• Varchar		• Solo lectura
• Tipo de organización		• Varchar		• Solo lectura
• Siglas		• Varchar		• Solo lectura
• Estado		• Bool		• Solo lectura
• Descripción		• Text		• Solo lectura
Observaciones		<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC18	Listar tipo de organización	<ul style="list-style-type: none"> • El requisito permite listar tipo de organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de organización del menú de 	Baja	Media

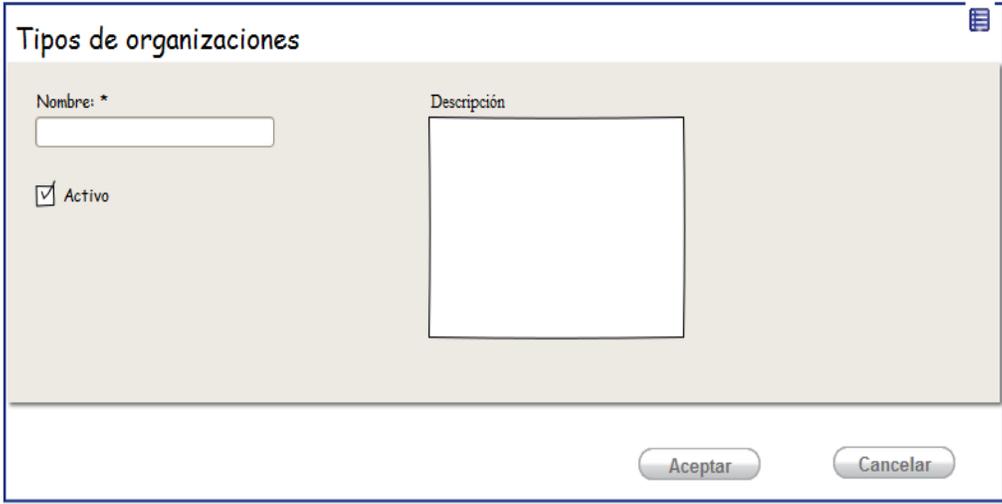
		<p>funcionalidades Configuración.</p> <ul style="list-style-type: none"> El sistema muestra un listado con todos los tipos de organizaciones existentes con el Nombre, Estado y acciones a realizar sobre el elemento. 		
Prototipo				
				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Los tipos de organizaciones se ordenan en la lista por orden alfabético. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página. 		

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC19	Crear tipo de organización	<ul style="list-style-type: none"> • El requisito permite crear tipo de organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de organización del menú de funcionalidades Configuración. • El usuario selecciona la opción crear e introduce el Nombre, Descripción y el Estado. • Si el usuario desea que la organización no esté activa debe Desactivarla, pues siempre se inicia Activo. • El usuario puede crear otro tipo de organización o volver al listado el cual se actualizará. 	Media	Alta
Prototipo				

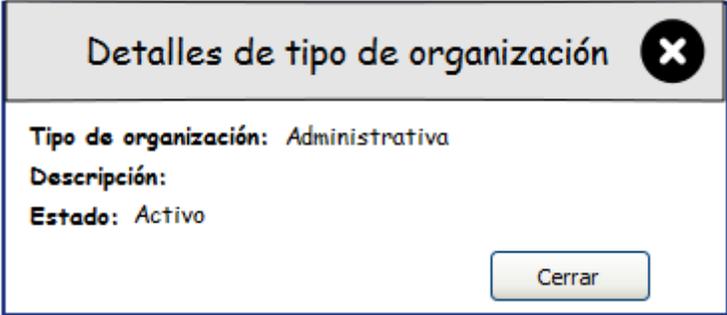
		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras.
<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo 	

		<p>“Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido.</p> <ol style="list-style-type: none"> 4. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. 5. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente. 7. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”. 8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo. 		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC20	Modificar tipo de organización	<ul style="list-style-type: none"> • El requisito permite modificar un tipo de organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de organización del menú de funcionalidades Configuración. 	Media	Alta

		<ul style="list-style-type: none"> • El usuario selecciona la opción modificar e introduce el Nombre, Descripción y el Estado. • Si el usuario no modifica algún campo, el tipo de organización conservará los datos anteriores. • Si el usuario guarda los cambios hechos se muestra el listado de los tipos de organizaciones actualizado. Si el usuario cancela la modificación del tipo de organización, se mostrará el listado de los tipos de organizaciones. 		
Prototipo				

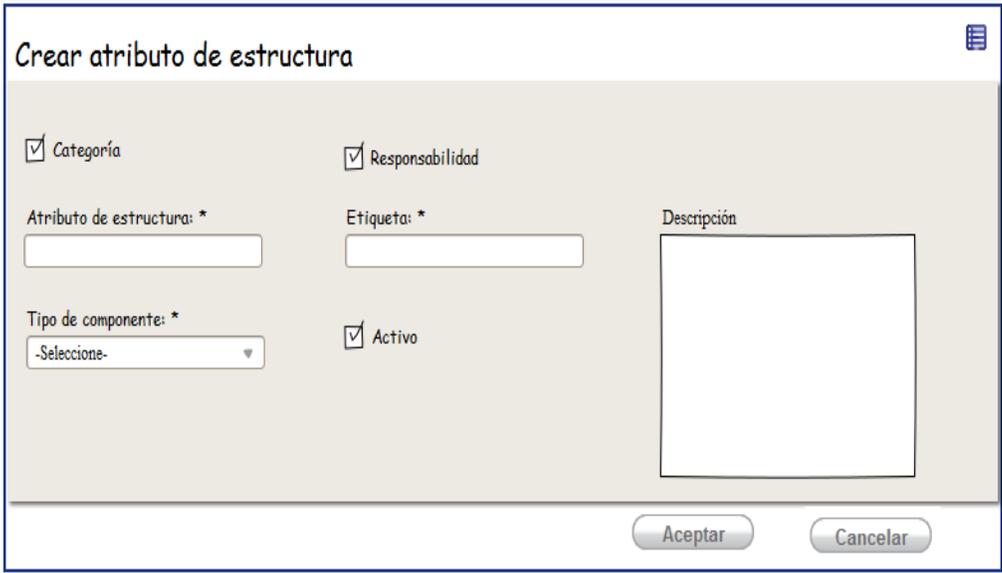
		
Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras.
<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo 	

		<p>“Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido.</p> <p>4. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido modificado satisfactoriamente.”.</p> <p>5. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”.</p> <p>6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente.</p> <p>7. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”.</p> <p>8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC21	Ver detalles de tipo de organización	<ul style="list-style-type: none"> El requisito permite ver detalles de tipo de organización. El usuario selecciona del componente Estructura y composición del SGU, la opción Tipo de organización del menú de funcionalidades Configuración. 	Baja	Baja

		<ul style="list-style-type: none"> El usuario selecciona la opción ver detalles de tipo de organización y el sistema muestra una ventana emergente con los datos: Tipo de organización, Descripción y el Estado. Para volver al listado se selecciona la opción cerrar. 		
Prototipo				
				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> Tipo de organización 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Solo lectura 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción. 		

N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC22	Listar atributos de estructura	<ul style="list-style-type: none"> • El requisito permite listar atributos de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Atributo del menú de funcionalidades Configuración. • El sistema muestra un listado con todos los atributos de estructura existentes con el Nombre, Tipo, Estado y acciones a realizar sobre el elemento. 	Baja	Media
Prototipo				
				
Campos	Tipos de Datos	Reglas o Restricciones		

	<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Tipo de entidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Las entidades se ordenan en la lista por orden alfabético. La cantidad de elementos a mostrar en la lista son 5, 10, 15 y 20 según la preferencia del usuario. Si el usuario introduce un número de página mayor que la cantidad de páginas el sistema muestra la última página. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC23	Crear atributos de estructura	<ul style="list-style-type: none"> El requisito permite crear atributos de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Atributos del menú de funcionalidades Configuración. El usuario selecciona la opción crear e introduce la Categoría, Responsabilidad, Nombre del atributo de estructura, Etiqueta, Descripción, Tipo de componente y el Estado. Si el usuario desea que de los atributos de 	Media	Media

		<p>estructura no esté activado la Categoría, Responsabilidad y el Estado debe Desactivarlo, pues siempre se inicia Activo.</p> <ul style="list-style-type: none"> El usuario puede crear otros atributos de estructura o volver al listado el cual se actualizará. 		
Prototipo				
				
Campos		Tipos de Datos	Reglas o Restricciones	
<ul style="list-style-type: none"> Categoría 		<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección 	
<ul style="list-style-type: none"> Responsabilidad 		<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección 	

	<ul style="list-style-type: none"> Nombre de atributo de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Admite solo letras.
	<ul style="list-style-type: none"> Etiqueta 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras.
	<ul style="list-style-type: none"> Tipo de componente 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Obligatorio Selección
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Admite cualquier tipo de carácter. Solo hasta 30 caracteres por palabras
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo “Entre solo letras, números y espacio o guión bajo entre palabras” encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 	

		<p>6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente.</p> <p>7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?".</p> <p>8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p> <p>9. En caso que el usuario introduce un carácter extraño en un campo que solo acepta letras el sistema muestra un mensaje en rojo "Entre solo letras" encima del campo que debe ser llenado correctamente.</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFestC24	Modificar atributos de estructura	<ul style="list-style-type: none"> El requisito permite modificar atributos de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Atributos del menú de funcionalidades Configuración. El usuario selecciona la opción modificar e introduce la Categoría, Responsabilidad, Nombre del atributo de estructura, Etiqueta, 	Media	Media

		<p>Descripción, Tipo de componente y el Estado.</p> <ul style="list-style-type: none"> • Si el usuario no modifica algún campo, los atributos de estructura conservarán los datos anteriores. • Si el usuario guarda los cambios hechos se muestra el listado de atributos de estructura actualizado. Si el usuario cancela la modificación de los atributos de estructura, se mostrará el listado de atributos de estructura. 		
--	--	--	--	--

Prototipo

Modificar atributo de estructura 

Categoría Responsabilidad

Atributo de estructura: * Etiqueta: * Descripción

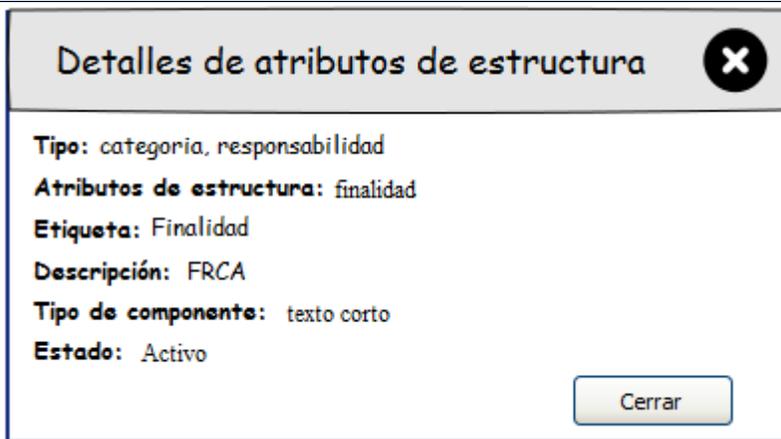
Tipo de componente: * Activo

	Campos	Tipos de Datos	Reglas o Restricciones
	<ul style="list-style-type: none"> • Categoría 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Responsabilidad 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Nombre de atributo de estructura 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Admite solo letras.
	<ul style="list-style-type: none"> • Etiqueta 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras.
	<ul style="list-style-type: none"> • Tipo de componente 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Obligatorio • Selección
	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Descripción 	<ul style="list-style-type: none"> • Text 	<ul style="list-style-type: none"> • Admite cualquier tipo de carácter. • Solo hasta 30 caracteres por palabras
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. 3. Cuando el usuario introduce un carácter extraño en un campo que solo acepta letras, números, espacio o guión bajo entre palabras el sistema muestra un mensaje en rojo "Entre solo letras, números y espacio o guión bajo entre palabras" encima del campo que debe ser llenado correctamente. Siendo los caracteres permitidos aquellos definidos por el dominio válido. 	

		<p>4. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido modificado satisfactoriamente."</p> <p>5. En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe".</p> <p>6. Si se introduce más de la cantidad de caracteres por palabra definidos para los campos de texto (30) el sistema muestra un mensaje en rojo: "Ha excedido el número de letras permitidas para una palabra" encima el campo que debe ser llenado correctamente.</p> <p>7. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?".</p> <p>8. Para todos los campos de texto cuando llegue a la cantidad máxima de caracteres definidos el sistema no permitirá continuar escribiendo.</p> <p>9. En caso que el usuario introduce un carácter extraño en un campo que solo acepta letras el sistema muestra un mensaje en rojo "Entre solo letras" encima del campo que debe ser llenado correctamente.</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC25	Ver detalles de atributos de estructura	<ul style="list-style-type: none"> El requisito permite ver detalles de atributos de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Atributo del menú de funcionalidades Configuración. 	Baja	Baja

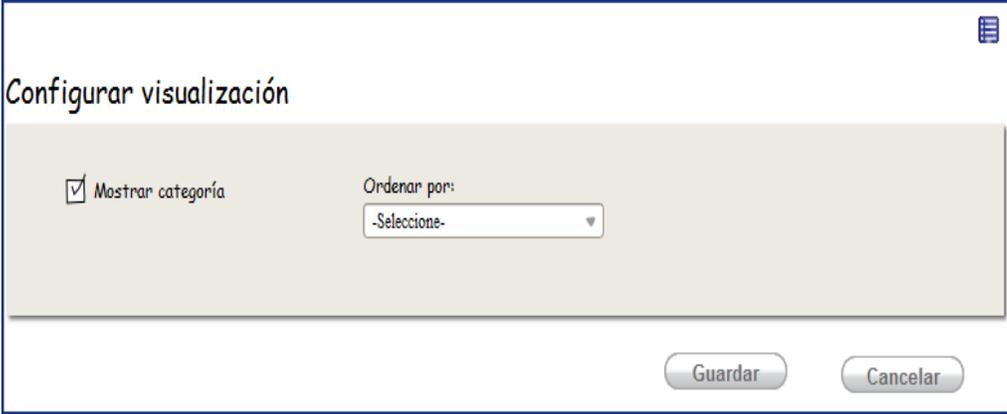
		<ul style="list-style-type: none"> El usuario selecciona la opción ver detalles de atributos de estructura y el sistema muestra una ventana emergente con los datos: Tipo, Atributo de estructura, Etiqueta, Descripción, Tipo de Componente y el Estado. Para volver al listado de atributos de estructura se selecciona la opción cerrar. 		
--	--	--	--	--

Prototipo



	Campos	Tipos de Datos	Reglas o Restricciones
	<ul style="list-style-type: none"> Tipo 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Solo lectura
	<ul style="list-style-type: none"> Atributo de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura

	<ul style="list-style-type: none"> Etiqueta 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Descripción 	<ul style="list-style-type: none"> Text 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Tipo de componente 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Solo lectura 	
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Solo lectura 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. Al pasar el puntero del mouse sobre el ícono de ver detalles este debe indicar la acción. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC26	Configurar visualización	<ul style="list-style-type: none"> El requisito permite Configurar visualización. El usuario selecciona del componente Estructura y composición del SGU, la opción Visualización de estructura del menú de funcionalidades Configuración. El sistema muestra un árbol de estructuras con el Nombre y acciones a realizar sobre el elemento. El usuario selecciona la opción configurar visualización y el sistema muestra el dato Mostrar categoría, el cual se inicializa activado. 	Baja	Media

		<ul style="list-style-type: none"> • El usuario puede seleccionar el criterio de ordenación del árbol de estructura. • Si el usuario guarda los cambios hechos se muestra el árbol de estructuras actualizado. Si el usuario cancela la acción, se mostrará el árbol de estructuras. 		
Prototipo				
				
	Campos	Tipos de Datos	Reglas o Restricciones	
	<ul style="list-style-type: none"> • Mostrar categoría 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección 	
	<ul style="list-style-type: none"> • Ordenar por 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Selección 	
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. Si no ocurren errores debe mostrarse el mensaje de 		

		<p>información: “La configuración ha sido guardada satisfactoriamente.”.</p> <p>3. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”.</p>		
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFestC27	Establecer nivel de prioridad de categoría de estructura	<ul style="list-style-type: none"> • El requisito permite establecer nivel de prioridad de categoría de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Visualización de estructura o Estructura del menú de funcionalidades Configuración o Estructura. • El usuario podrá aumentar o disminuir la prioridad con la que se muestra la categoría de las estructuras en el árbol de estructuras. • El sistema muestra un árbol de estructuras con la opción de modificar estructuras. 	Alta	Media
Prototipo				

<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: right;"></p> <h3>Estructuras</h3> <p>Entidad: 100-001 Universidad de las Ciencias Informáticas</p> <div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> [-] Rectorado + - [-] 100-002-Rectorado  [-] Organizaciones + - [-] Secretaría + - [-] Departamento central + - [-] Dirección + - [-] Facultad + - </div> </div>				
Campos		Tipos de Datos	Reglas o Restricciones	
Observaciones		<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. La cantidad de elementos a mostrar en el árbol de estructuras los seleccionará el usuario desplegando las estructuras. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC28	Asociar responsabilidad es con cantidad de plazas variables	<ul style="list-style-type: none"> • El requisito permite asociar responsabilidades con cantidad de plazas variables. El usuario selecciona del componente Estructura y composición del SGU, la opción Asociar responsabilidad del menú 	Alta	Alta

		<p>de funcionalidades Composición.</p> <ul style="list-style-type: none"> • El usuario selecciona la estructura o las estructuras a la cual le va a asociar responsabilidades con cantidad de plazas variables. El usuario puede filtrar la búsqueda mediante el nombre o la categoría a la que pertenece dicha estructura. • El usuario selecciona la responsabilidad a la cual le va a asignar una determinada cantidad de plazas variables. El sistema muestra los datos a llenar según la responsabilidad seleccionada. • El usuario introduce la cantidad de plazas y asocia dichas plazas a la responsabilidad. • El usuario puede asociar otras responsabilidades a una cantidad de plazas variables. Además, puede volver a la selección de 		
--	--	---	--	--

		estructuras o al listado el cual se actualizará.								
Prototipo										
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="text-align: right; margin-bottom: 10px;"></div> <h3>Asociar composición a esa estructura</h3> <div style="display: flex; justify-content: space-around; margin-bottom: 10px;"> <div style="background-color: #444; color: white; padding: 5px 10px; border-radius: 3px;">Selección de estructura</div> <div style="background-color: #eee; padding: 5px 10px; border-radius: 3px;">Selección de responsabilidad</div> </div> <div style="background-color: #f9f9f9; padding: 10px; border: 1px solid #ccc; margin-bottom: 10px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <input style="width: 150px;" type="text" value="Nombre de la estructura"/> <div style="background-color: #eee; padding: 5px 15px; border-radius: 3px; cursor: pointer;">Buscar</div> <div style="text-align: right;">Filtros</div> </div> <div style="text-align: right; margin-top: 5px;"> <input style="width: 100px;" type="text" value="-Selecione-"/> </div> </div> <div style="background-color: #f9f9f9; padding: 10px; border: 1px solid #ccc;"> <div style="text-align: right; margin-bottom: 5px;">Cantidad por página 5 </div> <div style="margin-bottom: 5px;">Estructura </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><input checked="" type="checkbox"/> Centro de Diseño</td> <td style="text-align: right; padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;"><input checked="" type="checkbox"/> Dirección Complejo Alimentos 1</td> <td style="text-align: right; padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;"><input type="checkbox"/> Dirección Complejo Alimentos 2</td> <td style="text-align: right; padding: 5px;"></td> </tr> </table> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <div style="text-align: center;">   Página <input style="width: 20px; text-align: center;" type="text" value="1"/> de 5   </div> <div style="text-align: right;">Resultados encontrados: 8</div> </div> </div> <div style="text-align: right; margin-top: 10px;"> <div style="display: flex; justify-content: flex-end; gap: 20px;"> <div style="background-color: #eee; padding: 5px 15px; border-radius: 3px; cursor: pointer;">Siguiente</div> <div style="background-color: #eee; padding: 5px 15px; border-radius: 3px; cursor: pointer;">Cancelar</div> </div> </div> </div>					<input checked="" type="checkbox"/> Centro de Diseño		<input checked="" type="checkbox"/> Dirección Complejo Alimentos 1		<input type="checkbox"/> Dirección Complejo Alimentos 2	
<input checked="" type="checkbox"/> Centro de Diseño										
<input checked="" type="checkbox"/> Dirección Complejo Alimentos 1										
<input type="checkbox"/> Dirección Complejo Alimentos 2										



Asociar composición a esa estructura

Selección de estructura
Selección de responsabilidad

Responsabilidad: *

Cantidad de plazas

-Selecione- ▼

Asociar

Responsabilidades asociadas

Responsabilidad	Cantidad de plazas	
Administrador	2	
Administrador General	6	
Auditor	9	

Anterior

Aceptar

Cancelar

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Único Alfanumérico, admite guiones bajos y espacios entre palabras. Admite cualquier tipo de carácter.
<ul style="list-style-type: none"> Filtros 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección

	<ul style="list-style-type: none"> Responsabilidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección 	
	<ul style="list-style-type: none"> Cantidad de plazas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Admite de 1 a 6 caracteres. 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente". En caso que el valor de un campo único exista se muestra un mensaje de error: "El elemento ya existe". En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". El estado se muestra activo por defecto. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC29	Listar árbol de estructura	<ul style="list-style-type: none"> El requisito permite listar el árbol de estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Estructura del menú de funcionalidades Estructura. El sistema muestra un árbol de estructuras con las acciones a realizar sobre las estructuras. 	Alta	Media
Prototipo				

	<div data-bbox="298 369 1268 856" style="border: 1px solid black; padding: 5px;"> <p>Estructuras</p> <p>Entidad: 100-001 Universidad de las Ciencias Informáticas</p> <div style="border: 1px solid gray; padding: 5px;"> <p>[-] Rectorado</p> <ul style="list-style-type: none"> [-] 100-002-Rectorado [icon] [-] Organizaciones [icon] [-] Secretaría [icon] [-] Departamento central [icon] [-] Dirección [icon] </div> </div>			
	Campos	Tipos de Datos	Reglas o Restricciones	
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. La cantidad de elementos a mostrar en el árbol de estructuras los seleccionará el usuario desplegando las estructuras. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC30	Crear estructura	<ul style="list-style-type: none"> • El requisito permite crear estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Estructura del menú de funcionalidades Estructura. • El usuario selecciona la opción crear e introduce los datos: Tipo de 	Alta	Alta

		<p>estructura, Nombre, Hijo de, Siglas, Categoría de estructura, Nivel, Fecha de activación y Estado.</p> <ul style="list-style-type: none"> • Cuando el usuario selecciona la categoría de la estructura datos a llenar según la selección que realice. • El usuario puede seleccionar vincular la estructura a otra estructura, donde deberá seleccionar en el árbol de estructura que estructura desea vincular a la estructura creada. • Si el usuario desactiva la estructura, no se mostrará el dato fecha de activación. • El usuario puede crear otra estructura o volver al listado el cual se actualizará. • El usuario puede realizar diferentes acciones sobre la estructura de mando creada como Ver detalles, Generar organigrama, Mostrar estructuras 		
--	--	---	--	--

		<p>asociadas, Definir estructuras de mando, Crear una estructura hija directamente en el árbol de estructuras, Modificar, Asociar responsabilidades, Clonar y Combinar.</p>		
<p>Prototipo</p>				
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="text-align: right; margin-bottom: 10px;">  </div> <h3 style="margin: 0;">Crear estructura</h3> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p>Tipo de estructura: *</p> <input type="text" value="-Seleccione-"/> </div> <div style="width: 30%;"> <p>Nombre: *</p> <input type="text" value="-Seleccione-"/> </div> <div style="width: 30%;"> <p>Hijo de: *</p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p>Siglas: *</p> <input type="text"/> </div> <div style="width: 30%;"> <p>Categoría de estructura: *</p> <input type="text" value="-Seleccione-"/> </div> <div style="width: 30%;"> <p>Nivel: *</p> <input type="text" value="-Seleccione-"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p>Fecha de activación: *</p> <input type="text"/>  </div> <div style="width: 30%;"> <input checked="" type="checkbox"/> Activo </div> </div> <div style="margin-top: 20px;"> <input type="checkbox"/> Vincular a estructura </div> <div style="text-align: right; margin-top: 20px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>				

Crear estructura

Tipo de estructura: *
-Seleccione-

Nombre: *

Hijo de: *
-Seleccione-

Siglas: *

Categoría de estructura: *
-Seleccione-

Nivel: *
-Seleccione-

Fecha de activación: * 

Activo

Vincular a estructura

Rectorado

- 100-002-Rectorado     
- Organizaciones     
- Secretaría     
- Departamento central     
- Dirección     

Aceptar
Cancelar

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Nombre 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Único Alfanumérico, admite guiones bajos y espacios entre palabras.

			<ul style="list-style-type: none"> Admite cualquier tipo de carácter.
	<ul style="list-style-type: none"> Hijo de 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
	<ul style="list-style-type: none"> Siglas 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Único Alfanumérico, admite guiones bajos y espacios entre palabras. Admite cualquier tipo de carácter. Admite de 1 a 6 caracteres.
	<ul style="list-style-type: none"> Categoría de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Alfanumérico, admite guiones bajos y espacios entre palabras. No se admiten: <ul style="list-style-type: none"> 1-% 2-' 3-espacios en los límites de la palabra
	<ul style="list-style-type: none"> Nivel 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Obligatorio Selección
	<ul style="list-style-type: none"> Fecha de activación 	<ul style="list-style-type: none"> Timestamp 	<ul style="list-style-type: none"> Obligatorio Selección
	<ul style="list-style-type: none"> Estado 	<ul style="list-style-type: none"> Bool 	<ul style="list-style-type: none"> Selección
	<ul style="list-style-type: none"> Vincular a 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Selección

	estructura			
	<ul style="list-style-type: none"> Atributos 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Selección Admite solo letras. 	
	Observaciones	<ol style="list-style-type: none"> El usuario debe estar autenticado en el sistema. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”. El estado se muestra activo por defecto. La entidad varía según el valor del tipo de estructura seleccionado. 		
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC31	Clonar estructura	<ul style="list-style-type: none"> El requisito permite Clonar estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Estructura del menú de funcionalidades Estructura. El usuario debe seleccionar la estructura que desea clonar en el árbol de estructuras. Luego selecciona la cantidad de 	Alta	Alta

		<p>clonaciones que desea de dicha estructura.</p> <ul style="list-style-type: none"> • El usuario selecciona en qué estructura desea clonar o la categoría de una estructura, donde se clonará la estructura dentro de todas las estructuras pertenecientes a dicha categoría, tantas veces como fue seleccionado. • El sistema volverá al árbol de estructuras el cual se actualizará. 		
<p>Prototipo</p>				

Clonar estructura

Cantidad: *

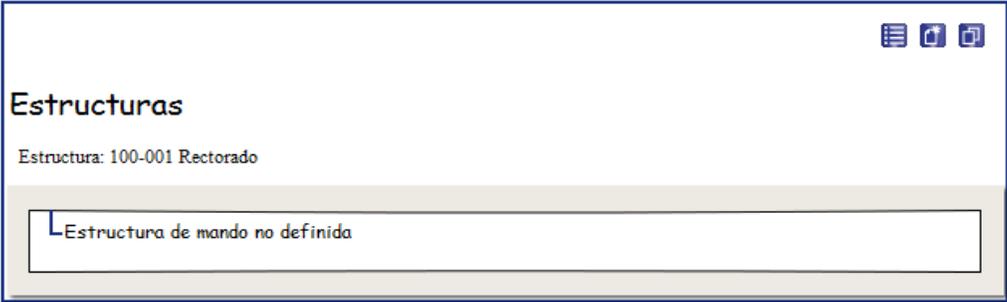
[-] 100-000-Administración	<input type="radio"/>
[-] 100-001-Rectorado	<input type="radio"/>
[-] 100-002-Dirección de Informatización	<input type="radio"/>
[-] 100-003-Dirección de Recursos Humanos	<input type="radio"/>
[-] 100-004-Vicerrectoría de Formación	<input type="radio"/>
[-] 100-005-Vicerrectoría de Extensión	<input type="radio"/>
[-] 100-006-Facultad 1	<input type="radio"/>
[-] 100-014-Facultad 3	<input type="radio"/>
[-] 100-014-Facultad 4	<input type="radio"/>
[-] 100-014-Facultad 2	<input type="radio"/>
[-] 100-014-Facultad 6	<input type="radio"/>
[-] 100-014-Facultad 7	<input type="radio"/>

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Cantidad 	<ul style="list-style-type: none"> Integer 	<ul style="list-style-type: none"> Obligatorio Solo admite de 1 a 10 números
Observaciones	<p>16. El usuario debe estar autenticado en el sistema.</p> <p>17. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Campo requerido" encima del campo que debe ser llenado obligatoriamente.</p> <p>18. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente."</p> <p>19. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?".</p> <p>20. En caso de no seleccionar una estructura o categoría se muestra un mensaje en rojo "Debe seleccionar un elemento."</p>	

encima del campo que debe ser llenado obligatoriamente.				
N°	Nombre	Descripción	Complejidad	Prioridad para cliente
RFEstC32	Combinar estructura	<ul style="list-style-type: none"> • El requisito permite combinar estructura. El usuario selecciona del componente Estructura y composición del SGU, la opción Estructura del menú de funcionalidades Estructura. • El usuario debe seleccionar la estructura que desea Combinar en el árbol de estructuras. Luego selecciona con que estructura se desea combinar, quedando agrupado en esta toda la composición de la estructura que se desea combinar. • El sistema volverá al árbol de estructuras el cual se actualizará. 	Alta	Alta
Prototipo				

<div style="border: 1px solid black; padding: 10px;"> <p>Combinar estructura</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>-100-014-Facultad 1</td><td style="text-align: right;"><input type="radio"/></td></tr> <tr><td>-100-014-Facultad 3</td><td style="text-align: right;"><input type="radio"/></td></tr> <tr><td>-100-014-Facultad 4</td><td style="text-align: right;"><input type="radio"/></td></tr> <tr><td>-100-014-Facultad 2</td><td style="text-align: right;"><input type="radio"/></td></tr> <tr><td>-100-014-Facultad 6</td><td style="text-align: right;"><input type="radio"/></td></tr> <tr><td>-100-014-Facultad 7</td><td style="text-align: right;"><input type="radio"/></td></tr> </table> <p style="text-align: right;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>					-100-014-Facultad 1	<input type="radio"/>	-100-014-Facultad 3	<input type="radio"/>	-100-014-Facultad 4	<input type="radio"/>	-100-014-Facultad 2	<input type="radio"/>	-100-014-Facultad 6	<input type="radio"/>	-100-014-Facultad 7	<input type="radio"/>
-100-014-Facultad 1	<input type="radio"/>															
-100-014-Facultad 3	<input type="radio"/>															
-100-014-Facultad 4	<input type="radio"/>															
-100-014-Facultad 2	<input type="radio"/>															
-100-014-Facultad 6	<input type="radio"/>															
-100-014-Facultad 7	<input type="radio"/>															
Campos		Tipos de Datos	Reglas o Restricciones													
Observaciones		<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo "Debe seleccionar un elemento." encima del campo que debe ser llenado obligatoriamente. 3. Si no ocurren errores debe mostrarse el mensaje de información: "El elemento ha sido creado satisfactoriamente." 4. En caso de cancelar la acción se muestra un mensaje de advertencia "¿Está seguro de realizar la acción?". 														
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente												
RFestC33	Crear estructura de mando	<ul style="list-style-type: none"> • El requisito permite Crear estructura de mando. El usuario selecciona del componente Estructura y 	Alta	Alta												

		<p>composición del SGU, la opción Estructura del menú de funcionalidades Estructura.</p> <ul style="list-style-type: none"> • El usuario selecciona la opción definir estructura de mando. Luego selecciona la opción crear e introduce los datos: Tipo de estructura, Responsabilidad, Hijo de, Siglas, Categoría de estructura, Nivel, Fecha de activación y Cantidad de plazas. • Cuando el usuario selecciona la categoría de la estructura se muestran datos a llenar según la selección que realice. • El usuario puede seleccionar asociar a responsabilidad. • Si el usuario desactiva la estructura, no se mostrará el dato fecha de activación. • El usuario puede realizar diferentes acciones sobre la estructura creada como Ver detalles, Generar 		
--	--	--	--	--

		<p>organigrama, Modificar, Clonar y Crear una estructura hija directamente en el árbol de estructuras.</p> <ul style="list-style-type: none"> • El usuario puede crear otra estructura o volver al listado el cual se actualizará. 		
<p>Prototipo</p>				
 <p>The screenshot shows a web interface titled "Estructuras". Below the title, it displays "Estructura: 100-001 Rectorado". A list of structures is shown, with the first entry being "Estructura de mando no definida". In the top right corner of the interface, there are three icons: a list icon, a plus icon, and a refresh icon.</p>				

Crear estructura

Tipo de estructura: *
Cadena de mando

Siglas: *

Fecha de activación: *

Responsabilidad: *
-Seleccione-

Categoría de estructura: *
-Seleccione-

Activo

Hijo de: *
-Seleccione-

Nivel: *
-Seleccione-

Cantidad de plazas: *

Asociar a responsabilidad

Aceptar
Cancelar

Estructuras

Estructura: 100-001 Rectorado

Campos	Tipos de Datos	Reglas o Restricciones
<ul style="list-style-type: none"> Tipo de estructura 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección
<ul style="list-style-type: none"> Responsabilidad 	<ul style="list-style-type: none"> Varchar 	<ul style="list-style-type: none"> Obligatorio Selección Único Alfanumérico, admite guiones bajos y

			<p>espacios entre palabras.</p> <ul style="list-style-type: none"> • Admite cualquier tipo de carácter.
	<ul style="list-style-type: none"> • Hijo de 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Selección
	<ul style="list-style-type: none"> • Siglas 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Único • Alfanumérico, admite guiones bajos y espacios entre palabras. • Admite cualquier tipo de carácter. • Admite de 1 a 6 caracteres.
	<ul style="list-style-type: none"> • Categoría de estructura 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Obligatorio • Selección • Alfanumérico, admite guiones bajos y espacios entre palabras. • No se admiten: <ul style="list-style-type: none"> • 1-% • 2-' • 3-espacios en los límites de la palabra
	<ul style="list-style-type: none"> • Nivel 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Obligatorio • Selección
	<ul style="list-style-type: none"> • Fecha de activación 	<ul style="list-style-type: none"> • Timestamp 	<ul style="list-style-type: none"> • Obligatorio • Selección

	<ul style="list-style-type: none"> • Estado 	<ul style="list-style-type: none"> • Bool 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Asociar a responsabilidad 	<ul style="list-style-type: none"> • Integer 	<ul style="list-style-type: none"> • Selección
	<ul style="list-style-type: none"> • Cantidad de plazas 	<ul style="list-style-type: none"> • Varchar 	<ul style="list-style-type: none"> • Admite de 1 a 10 caracteres.
	Observaciones	<ol style="list-style-type: none"> 1. El usuario debe estar autenticado en el sistema. 2. En caso que se deje un campo de los obligatorios vacío se muestra un mensaje en rojo “Campo requerido” encima del campo que debe ser llenado obligatoriamente. 3. Si no ocurren errores debe mostrarse el mensaje de información: “El elemento ha sido creado satisfactoriamente.”. 4. En caso que el valor de un campo único exista se muestra un mensaje de error: “El elemento ya existe”. 5. En caso de cancelar la acción se muestra un mensaje de advertencia “¿Está seguro de realizar la acción?”. 6. El estado se muestra activo por defecto. 	

Anexo 3 Mapa conceptual

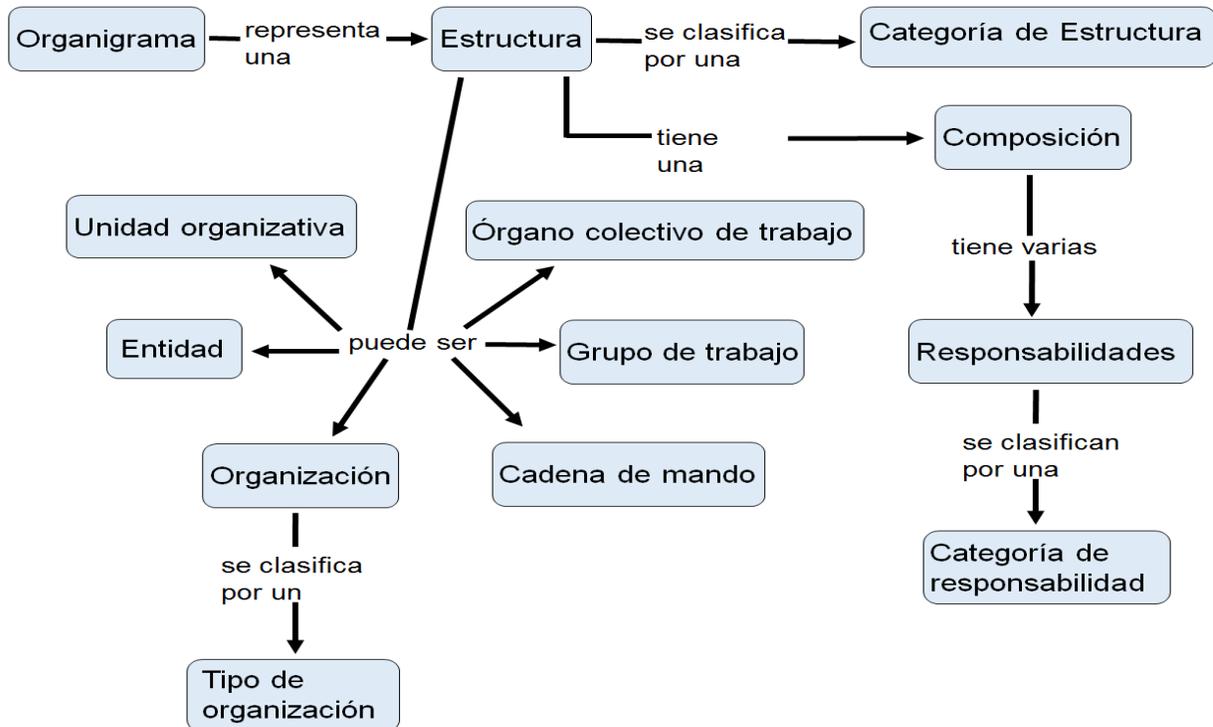


Figura 26. Mapa Conceptual.

Anexo 4: Casos de prueba de caja blanca

Prueba estructural de caja blanca	Código caso de prueba: SGU-NUC-EstC-010318_DCP_RFEst29_Listar árbol de estructura
Probador: Julio César Hernández González	
Código al que se aplica: <pre> public function arbol() { \$all_post = \$this->input->all_post(); \$idEntidad = \$this->xml->configuraciones->configuracion->entidad; \$estructuraActiva = \$this->estructura_lib->obtenerEstructuraEntidadActiva(\$idEntidad); if (empty(\$estructuraActiva))//Verificamos si existen datos return; \$idNodo = 0; </pre>	

```

$IdNodoEstructura = $estructuraActiva->id_estructura;
if ($all_post['root'] != 'source') {
    $IdNodo = $all_post['root'];
    $IdNodoEstructura = $all_post['id_nodo'];
} else
    $all_post['tipo'] = 'estructura';
$mostrarCategoria = $this->xml->configuraciones->estructura_composicion->opciones_visualizacion-
>mostrar_categoria;
if ($mostrarCategoria == 1) {
    if ($all_post['tipo'] === 'estructura')
        $hijos = $this->estructura_lib->hijosArbolCategoria($IdNodo, $IdNodoEstructura);
    else {
        $hijos = $this->estructura_lib->hijosArbolDadoldCategoria($IdNodo, $all_post['id_nodo_padre'],
$IdNodoEstructura);
    }
} else
    $hijos = $this->estructura_lib->hijosArbol($IdNodo, $IdNodoEstructura);
if (empty($hijos)) {
    $hijos[0] = new stdClass ();
    $hijos[0]->text = "Estructura no definida.";
}
return $hijos;
}

```

Complejidad ciclomática:

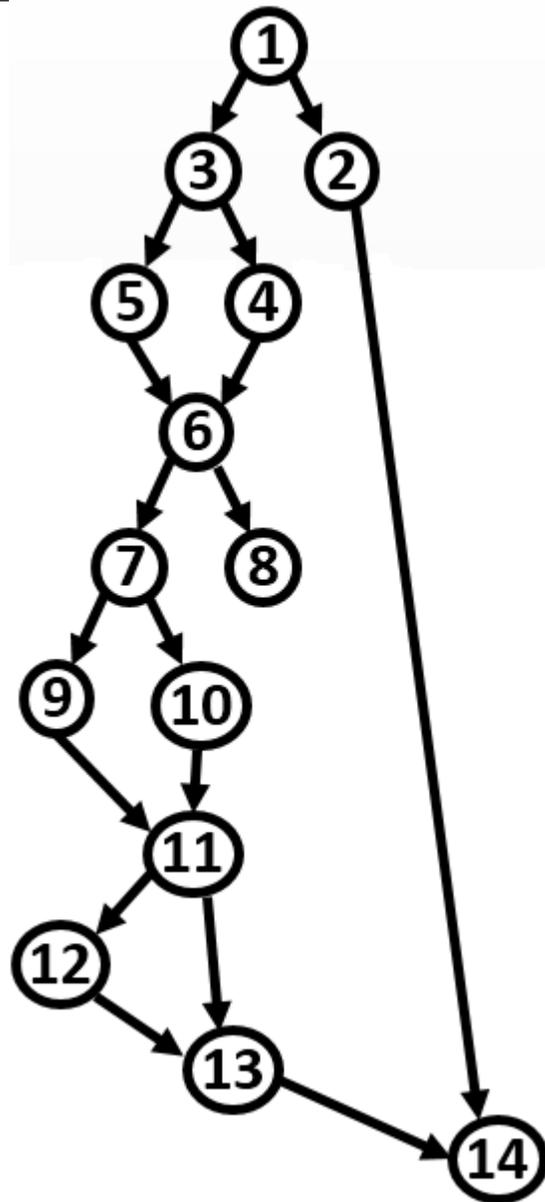
$V(G)=18 \text{ aristas} - 14 \text{ nodos} + 2 = 6$

$V(G)=5 \text{ nodos predicados} + 1 = 6$

Caminos independientes:

- 1) 1-2-14
- 2) 1-3-4-6-10-11-13-14
- 3) 1-3-5-6-10-11-13-14
- 4) 1-3-5-6-7-8-11-13-14
- 5) 1-3-5-6-7-9-11-13-14
- 6) 1-3-5-6-7-9-11-12-13-14

Representación del grafo:



Función de evaluación:

```

public function pruebaArbol() {
    $_POST['root'] = 'source';
    echo $this->unit->run($this->arbol(), is_array, 'Arbol de estructura');
}
  
```

Resultado:	
Test Name	Árbol de estructura
Test Datatype	Array
Expected Datatype	Array
Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/estructura_composicion/controllers/estructura.php
Line Number	31
Estado:	
Satisfactorio	

Tabla 11. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst29_Listar árbol de estructura.

Prueba estructural de caja blanca	Código caso de prueba: SGU-NUC-EstC-010318_DCP_RFEst35_ Generar organigrama
Probador: Julio César Hernández González	
Código al que se aplica:	
<pre> public function loadOrg() { \$all_post = \$this->input->all_post(); if (\$all_post['id_estructura'] != "") { \$id_estructura = \$all_post['id_estructura']; \$tipo = \$all_post['tipo']; } else { \$idEntidad = \$this->xml->configuraciones->configuracion->entidad; \$estructura = \$this->estructura_lib->obtenerEstructuraEntidadActiva(\$idEntidad); \$id_estructura = \$estructura->id_estructura; \$tipo = 1; } \$order = 0; \$fecha = ""; if (isset(\$all_post['fecha'])) \$fecha = \$all_post['fecha']; </pre>	

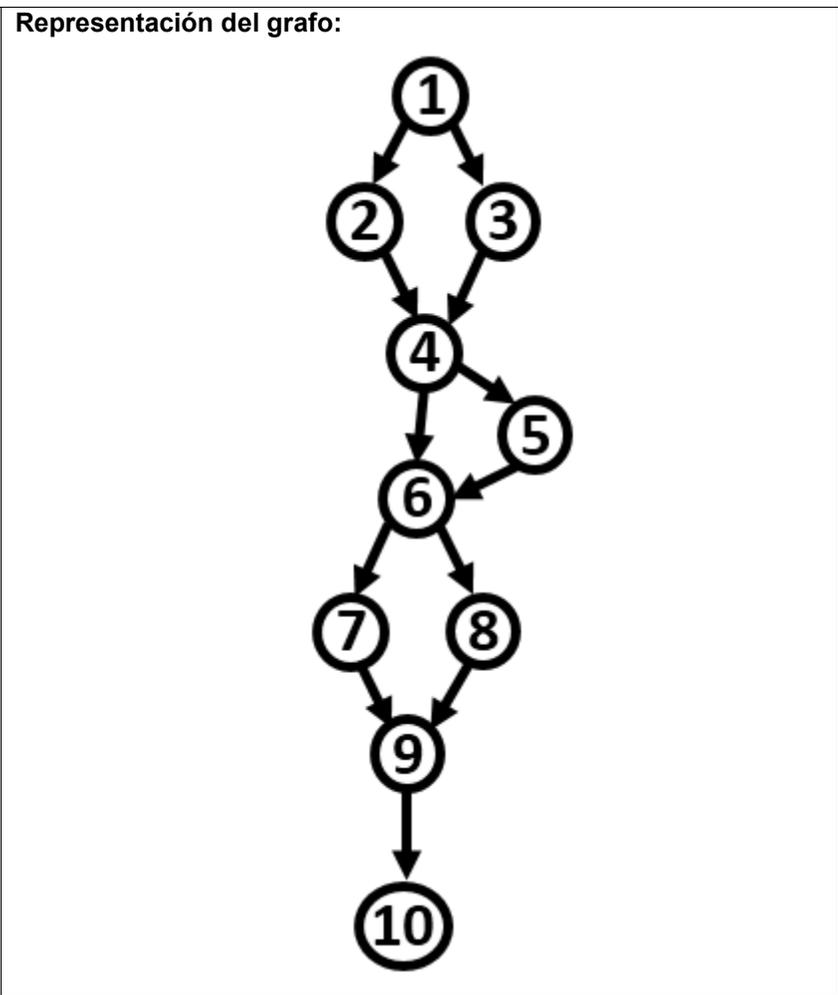
```

if ($tipo == 2)
    $arbol = $this->estructura_lib->obtenerArbolEstructuraOrgHTMLFormat($id_estructura, 0, $tipo, true, false,
"-1", "inner", $fecha);
else
    $arbol = $this->estructura_lib->obtenerArbolEstructuraOrgHTMLFormat($id_estructura, 0, 1, true, false, "-
1", "simple", $fecha);

return $arbol;
}
    
```

Complejidad ciclomática:
 $V(G)=12 \text{ aristas} - 10 \text{ nodos} + 2 = 4$
 $V(G)=3 \text{ nodos predicados} + 1 = 4$

Caminos independientes:
 1-2-4-6-8-9-10
 1-3-4-5-6-8-9-10
 1-3-4-6-7-9-10
 1-3-4-6-8-9-10



Función de evaluación:

<pre>public function pruebaloadOrg() { \$_POST['id_estructura'] = ""; echo \$this->unit->run(\$this->loadOrg(),is_string, 'Cargar organigrama'); }</pre>	
Resultado:	
Test Name	Cargar organigrama
Test Datatype	String
Expected Datatype	String
Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/estructura_composicion/controllers/organigrama.php
Line Number	14
Estado:	
Satisfactorio	

Tabla 12. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst35_Generar organigrama.

Prueba estructural de caja blanca	Código caso de prueba: SGU-NUC-EstC-010318_DCP_RFEst30_CrearEstructura
Probador: Julio César Hernández González	
Código al que se aplica:	
<pre>public function obtenerValidaciones() { \$all_post = \$this->input->all_post('true'); \$validaciones = \$this->xml->configuraciones_nomencladores->nucleo->estructura_composicion->validaciones->\$all_post['tipo_componente']; \$validaciones = json_decode(json_encode(\$validaciones)); \$infoValidaciones = array(); foreach (\$validaciones as \$key => \$value) { \$info = array(); \$data = \$this->xml->configuraciones_nomencladores->nucleo->estructura_composicion->validaciones->\$all_post['tipo_componente']->\$key; \$info['validador'] = \$key; \$info['texto'] = \$value; } }</pre>	

```

$info['valor'] = (string) $data['valor'];
$info['tipo_dato'] = (string) $data['tipo_dato'];
$info['agrupacion'] = (string) $data['agrupacion'];
$infoValidaciones[$key] = $info;
}
return $infoValidaciones;
}

```

Complejidad ciclomática:

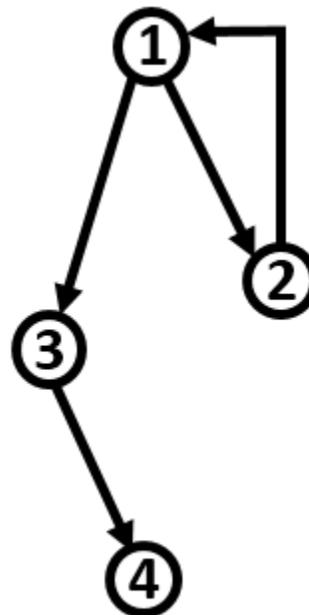
$V(G) = 4 \text{ aristas} - 4 \text{ nodos} + 2 = 2$

$V(G) = 1 \text{ nodo predicado} + 1 = 2$

Caminos independientes:

- 1) 1-3-4
- 2) 1-2-1-3-4

Representación del grafo:



Función de evaluación:

```

public function pruebaObtenerValidaciones() {
    $_POST['tipo_componente'] = 'fecha';
    echo $this->unit->run($this->obtenerValidaciones(), is_array, 'Obtener validaciones');
}

```

Resultado:

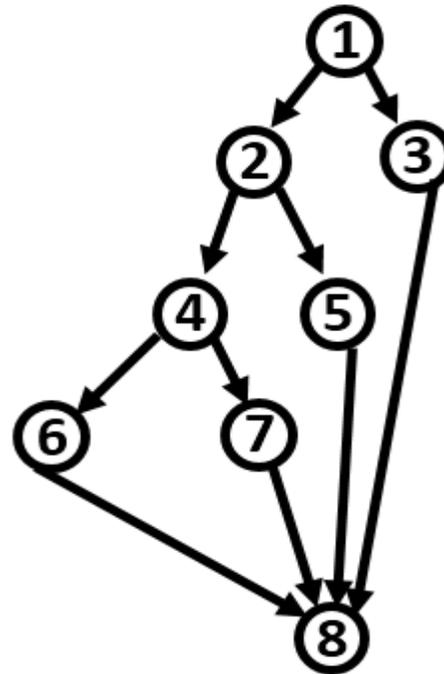
Test Name	Obtener validaciones
Test Datatype	Array
Expected Datatype	Array

Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/estructura_composicion/controllers/categoria_estructura.php
Line Number	13
Estado:	
Satisfactorio	

Tabla 13. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst30_CrearEstructura.

Prueba estructural de caja blanca	Código caso de prueba: SGU-NUC-EstC-010318_DCP_RFEst2_Crear estructura	categoría estructura
Probador: Julio César Hernández González		
Código al que se aplica:		
<pre> public function registrarCategoriaEstructura() { \$verificar = array('nombre_categoria_estructura'); if (\$this->input->is_post_back(\$verificar)) { \$all_post = \$this->input->all_post(true); \$existe = \$this->categoria_estructura_lib- >comprobarSiExisteDadoNombreCategoriaEstructura(\$all_post['nombre_categoria_estructura']); if (empty(\$existe)) { if (\$this->categoria_estructura_lib->registrarCategoriaEstructura(\$all_post) != false) { return true; } else { return false; } } else { return false; } } else { return false; } } </pre>		
Complejidad ciclomática:	Representación del grafo:	

$V(G) = 10 \text{ aristas} - 8 \text{ nodos} + 2 = 4$
 $V(G) = 3 \text{ nodos predicados} + 1 = 5$
Caminos independientes:
 1) 1-3-8
 2) 1-2-5-8
 3) 1-2-4-7-8
 4) 1-2-4-6-8



Función de evaluación:

```

public function pruebaRegistrarCategoria() {
    $_POST['nombre_categoria_estructura']='Categoria de prueba';
    echo $this->unit->run($this->registrarCategoriaEstructura(),true,'Registrar categoría');
}
    
```

Resultado:

Test Name	Registrar categoría
Test Datatype	Boolean
Expected Datatype	Boolean
Result	Passed
File Name	/var/www/Nucleo/base/application/sistema/estructura_composicion/controllers/categoria_estructura.php
Line Number	13

Estado:

Satisfactorio

Tabla 14. Caso de prueba de caja blanca SGU-NUC-EstC-010318_DCP_RFEst2_Crear categoría estructura.