



**DESARROLLO DE MÓDULO DE ESTRUCTURA Y
COMPOSICIÓN PARA EL SISTEMA DE GESTIÓN
ACADÉMICA DEL MINISTERIO DE EDUCACIÓN
(MINED)**



[Subtítulo del documento] Autor (a):

Yudenis Regla Lara Leandro

Tutores:

Ing. Mairelis Boeras Velázquez

Ing. Alexander Rodríguez Mompíe

septiembre de 2020

FORTES

DECLARACIÓN DE AUTORÍA:

Declaro por este medio que yo **Yudenis Regla Lara Leandro**, con carné de identidad **97122012176** soy el autor principal del trabajo titulado “**Desarrollo de Módulo de Estructura y Composición para el Sistema de Gestión Académica del Ministerio de Educación (MINED)**” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firmamos la presente a los 14 días del mes de septiembre de 2020



Autor: Yudenis Regla Lara Leandro

Tutor: Ing. Mairelis Boeras Velázquez

Ing. Alexander Rodríguez Mompíe

RESUMEN

Con esta investigación se pretende desarrollar el análisis y diseño del módulo Estructura y Composición para el Sistema de Gestión Académica del Ministerio de Educación de la República de Cuba, que permita gestionar la información de las estructuras organizativas asociadas al proceso educativo haciendo uso de herramientas libres para el desarrollo. Se estará utilizando como entorno de trabajo Django 3.0 para el backend y Angular 8.2 para el frontend. Para la solución de sistema estará apoyada en soluciones similares, utilizando una metodología propuesta por el centro de Tecnologías para la Formación (AUP en su variante UCI), para documentar de una forma más efectiva el sistema. Se propone además una estrategia de pruebas para el logro del correcto funcionamiento del sistema implementado. Como resultado se obtiene el análisis y diseño de una herramienta de trabajo más efectiva para la gestión de la información que constantemente se necesita relacionada con la estructura educacional.

Palabras Claves: Composición, Centro educacional, Estructura, Ministerio de educación

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA	5
1.1 <i>Conceptos Asociados</i>	5
1.2 <i>Análisis de soluciones similares</i>	6
1.3 <i>Tecnologías</i>	9
1.4 <i>Conclusiones del capítulo</i>	17
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN	18
2.1 <i>Descripción de la propuesta de solución</i>	18
2.2 <i>Conceptualización del dominio</i>	19
2.3 <i>Requisitos de Software</i>	19
2.4 <i>Descripción de requisitos por procesos</i>	24
2.5 <i>Descripción de la arquitectura</i>	25
2.6 <i>Modelo de datos</i>	27
2.7 <i>Modelo de despliegue</i>	30
2.8 <i>Conclusiones del capítulo</i>	30
CAPÍTULO 3: Implementación y pruebas de la propuesta de solución	30
3.1 <i>Diagrama de Componente</i>	31
3.2 <i>Estándares de Codificación</i>	31
3.3 <i>Estrategias de Pruebas</i>	32
3.4 <i>Estrategia de Validación (Técnica de IADOV)</i>	36
3.5 <i>Conclusiones del capítulo</i>	37
Conclusiones Generales	37
Recomendaciones.....	39
Bibliografía	40
Anexos	43

INTRODUCCIÓN

La sociedad actual se encuentra en un proceso de constante cambio, y el mundo enfrenta en reto de automatizarla, utilizando las nuevas Tecnologías de la Información y la Comunicaciones (TIC).

Las TIC posibilitan poner en práctica estrategias comunicativas y educativas para establecer nuevas formas de enseñar y aprender, mediante el empleo de concepciones avanzadas de gestión.

La evolución del concepto de formación se ha visto deslumbrado por la aparición y consolidación de las TIC, Internet y su realización en la *World Wide Web* (WWW) ha facilitado el acceso a todo tipo de información necesaria, provocando un aumento considerable de la interactividad entre personas de distintos continentes y países del mundo, brindando la posibilidad de desarrollar sus capacidades y habilidades para el tele trabajo, la interacción multicultural, el acceso a la información, al conocimiento y la educación con el objetivo de disminuir la brecha digital. Desde hace un par de décadas las nuevas TIC han sido incorporadas en los sistemas educativos del alrededor del mundo con la promesa de brindar mejoras en el sistema escolar. Los países de América Latina han realizado a lo largo de este tiempo importantes esfuerzos para no permanecer al margen de esta tendencia global [1] .

Cuba tiene la voluntad de fortalecer el sector de la informática y las comunicaciones, por eso se busca mediante las TIC informatizar los procesos de todas aquellas instituciones que se dedican al proceso de la enseñanza; para esta tarea se creó la Universidad de las Ciencias Informáticas (UCI) con sus centros de desarrollo de software.

En Cuba, en la actualidad se evidencian una diversidad de cambios en el ámbito de las Tecnologías, relacionados con el gran avance en el campo científico-técnico a nivel mundial. Estos avances llevan consigo la necesidad de crear nuevos sistemas que posibiliten facilidades a la hora de interactuar con estas tecnologías. Uno de los sectores donde el país está aportando es en el sector de la educación. Entre los ministerios que organizan y controlan la actividad educacional del país se encuentran el Ministerio de Educación (MINED) y el Ministerio de Educación Superior (MES).

El MINED tiene como misión esencial, dirigir, ejecutar y controlar la aplicación de la política del Estado y el Gobierno en la actividad educacional, excepto en la Educación Superior; o sea, para los niveles de

enseñanza: Círculos Infantiles, Educación Pre-escolar, Primaria, Secundaria básica, Preuniversitaria, Educación Técnica Profesional, Educación de adultos y la Educación especial. Estas enseñanzas son impartidas en centros educacionales dispersos por todo el país.

El departamento de servicio de información estadística perteneciente al MINED publicó en su anuario estadístico del curso 2018-2019 que Cuba cuenta con un total de 10 795 centros educacionales, y para lograr una organización de estos centros se han establecido niveles estructurales, tales como: el nivel nacional, como órgano superior, el provincial y el municipal. Los niveles provinciales coinciden con las provincias y los niveles municipales con los municipios, es por ello que cada centro educacional responde al municipio y provincia donde esté territorialmente ubicado.

Esta organización ha permitido que la información y control de los procesos educacionales fluya desde el nivel nacional hasta los centros y viceversa, de una forma aceptable. Actualmente los centros y los niveles de la estructura utilizan las hojas de cálculo como herramienta fundamental para guardar y consolidar la información, método que requiere de mucho tiempo y se torna engorroso cuando el volumen de datos es grande. Constantemente el MINED requiere de información relacionada con los centros, que debe ser consolidada para sus análisis estadísticos, y la agilidad de entrega de la misma se ve afectada por los problemas antes explicados; es por ello, que se plantea el siguiente problema de investigación.

Problema científico: ¿Cómo contribuir con la gestión de la información de las estructuras organizativas asociadas al proceso educativo guiado por el Ministerio de Educación en la República de Cuba?

El **objeto de estudio** es el Proceso de gestión de información de estructuras organizativas el cuál enmarca al **campo de acción** el Proceso de gestión de información de las estructuras organizativas del MINED encargadas de la consolidación de información del proceso educativo en Cuba.

El **Objetivo general definido** es Desarrollar un módulo para el Sistema de Gestión Académica del Ministerio de Educación de la República de Cuba, que permita gestionar la información de las estructuras organizativas asociadas al proceso educativo haciendo uso de herramientas libres para el desarrollo.

Preguntas Científicas

1. ¿Cómo realizar la selección de la metodología a utilizar para el desarrollo del módulo Estructura para el Sistema de Gestión Académica para el MINED?
2. ¿Cómo realizar la selección de las herramientas y tecnologías a utilizar para el desarrollo del módulo

Estructura para el Sistema de Gestión Académica para el MINED?

3. ¿Cómo realizar el análisis y diseño del módulo Estructura para el Sistema de Gestión Académica del MINED?
4. ¿Cómo definir una estrategia de pruebas y de validación acorde a la investigación?

Tareas de la investigación

1. Fundamentación de la selección de la metodología, herramientas y tecnologías a utilizar para el desarrollo del módulo Estructura para el Sistema de Gestión Académica del MINED.
2. Realización del análisis y diseño del módulo Estructura y Composición para el Sistema de Gestión Académica del MINED.
3. Definición de una estrategia de pruebas y de validación para el módulo Estructura y Composición para el Sistema de Gestión Académica del MINED.

Para el desarrollo de la investigación se utilizaron algunos métodos, los cuales se clasifican en teóricos y empíricos.

En la investigación se utilizaron los siguientes **métodos teóricos**:

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

Modelado: Se ha utilizado para modelar todos los diagramas correspondientes a la etapa de análisis, diseño e implementación del módulo a desarrollar en la presente investigación.

También se utilizan para la investigación los **métodos empíricos**:

Documental: Se utilizó para indagar en bibliografías de corte documental tales como, libros, artículos, etc.

Entrevista: Se utiliza para lograr un mejor entendimiento sobre el funcionamiento del proceso de estructura llevado a cabo en el Ministerio de Educación de la República de Cuba.

Resultados:

1. Contar con una herramienta de trabajo más efectiva para la gestión de la información que constantemente se necesita relacionada con la estructura educacional.
2. Aumentar la disponibilidad de la información relacionada con la estructura educacional de todo el País.
3. Dotar al MINED de una información más actualizada relacionada con la estructura educacional de todo el País, requerida para sus análisis estadísticos.
4. Obtener reportes de consolidación de información relacionados con la estructura educacional de una forma más rápida.

La investigación está estructurada en tres capítulos:

Capítulo 1: Fundamentación teórica del desarrollo del módulo Estructura para el Sistema de Gestión Académica del MINED.

En este capítulo se definen algunos conceptos importantes para la comprensión del objeto de estudio de la investigación. Se abordan los temas relacionados con el marco teórico de la investigación, es decir, se hace referencia al estado del arte del tema tratado en el ámbito nacional e internacional. Además, se realiza una descripción de las herramientas y tecnologías empleadas en el desarrollo de la solución.

Capítulo 2: Concepción de la propuesta de solución del desarrollo del módulo Estructura para el Sistema de Gestión Académica del MINED.

En este capítulo se describen las actividades que forman parte del proceso de Estructura y Composición, se muestra el objeto de informatización y la descripción general de la propuesta. Se definen los requisitos funcionales y no funcionales del módulo a desarrollar. Por último, se describe la arquitectura, los patrones de diseño y se muestra el modelo de la base de datos.

Capítulo 3: Implementación y pruebas del desarrollo del módulo Estructura para el Sistema de Gestión Académica del MINED.

En este capítulo se presenta el diagrama de componentes, los estándares de codificación a seguir por el equipo de desarrollo. Así como los métodos y técnicas para la realización de pruebas en el módulo.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En este capítulo se realiza un estudio del arte referente a problema de investigación, se describen los principales conceptos asociados, se hace un análisis de sistemas similares que no permitirá un mejor entendimiento del sistema, y se detallan las características de las tecnologías, herramientas y metodologías a utilizar para el desarrollo de la solución.

1.1 Conceptos Asociados

1.1.1 Estructura Organizacional

Se denomina estructura organizacional a la manera elegida por una entidad para gestionar su actividad y sus recursos. Esta estructura está dada por una serie de relaciones formales e informales que la corporación desarrolla para alcanzar sus objetivos y cumplir sus metas.

La estructura organizacional de una empresa implica disponer los roles de los empleados y ejecutivos de un cierto modo para alcanzar el mejor rendimiento posible. Con una buena estructura, las interrelaciones fluyen de manera óptima y cada uno de los actores de la entidad puede desempeñar su tarea de forma eficiente [2].

1.1.2 Organigrama

El organigrama es una representación gráfica del esqueleto de una organización, mostrando los cargos jerárquicos. Permite obtener una idea rápida de cómo está organizada una entidad.

Es una herramienta para el análisis organizacional, ya que como instrumento de análisis servirá para detectar fallas en la estructura, en él se ve claramente cada unidad o departamento y la relación con otras unidades. Siendo así, si algo está funcionando mal podemos estudiar el gráfico para entender y de este modo solucionar una situación [3].

1.1.3 Centro Educativo

Entre los múltiples significados del término centro, este se refiere al lugar donde las personas se reúnen con un determinado fin. Un centro, en este sentido, es un espacio físico (edificio) que permite la reunión y que ofrece determinados servicios o prestaciones.

Educativo, por otra parte, es lo perteneciente o relativo a la educación (el proceso de socialización de los individuos). Cuando una persona accede a la educación, recibe, asimila y aprende conocimientos, además de adquirir una concienciación cultural y conductual por parte de las generaciones anteriores.

Un centro educativo, por lo tanto, es un establecimiento destinado a la enseñanza. Es posible encontrar centros educativos de distinto tipo y con diferentes características, desde una escuela hasta una institución que se dedica a enseñar oficios pasando por un complejo cultural[4].

1.1.4 Proceso Educativo

El proceso educativo, se desarrolla en instituciones educativas como escuelas o universidades, contando con docentes profesionales, programas de estudio aprobados por el Estado y sistemas de evaluación que exigen al alumno el cumplimiento de ciertos objetivos.

El proceso educativo incluso puede desarrollarse a distancia, sin que las personas involucradas en el mismo estén cara a cara o tengan un contacto personal.

Estos procesos, en definitiva, permiten que los individuos que aprenden asimilen información necesaria para desenvolverse con éxito a nivel social, adquiriendo valores y pautas de conducta[5].

1.2 Análisis de soluciones similares

1.2.1 Akademos

El sistema fue creado por la Dirección de Informatización de la Universidad de las Ciencias Informáticas. Integra a su vez los marcos de trabajo CodeIgniter en su versión 1.7.3 y JQuery en su versión 1.7.1 en una sola infraestructura, razón por la que posee las características de ambos. En la integración se incluyen mejoras y modificaciones hechas específicamente al CodeIgniter, además de creación de plugins y componentes de interfaz de usuario en JQuery.

Este sistema permite representar de forma jerárquica la estructura administrativa de la entidad y asignar responsabilidades a personas pertenecientes a dicha estructura[6].

1.2.2 Cedrux

Cedrux, es un sistema integrado de gestión que se desarrolla en la Universidad de las Ciencias Informáticas. Estos sistemas poseen tres características que los identifican: integrales, modulares y adaptables. Durante su desarrollo se ha determinado la necesidad de diagnosticar dichas características a partir de análisis automatizados sobre su arquitectura en diferentes fases de su construcción. Para automatizar los análisis arquitectónicos es necesario antes: definir el estilo arquitectónico, formalizarlo de manera que pueda ser interpretado por un sistema informático, definir las métricas correspondientes e implementar una herramienta que permita aplicarlas. Este trabajo muestra la definición del estilo arquitectónico de Cedrux como primer paso en el desarrollo de la herramienta para el análisis arquitectónico del mismo. Se basa en la definición del vocabulario arquitectónico y las restricciones estructurales, sintácticas y semánticas de los elementos que conforman la arquitectura. Finalmente se exponen las ideas del trabajo futuro a partir de los resultados mostrados [7].

1.2.3 OrgPlus

OrgPlus está diseñado para administrar cada paso en el proceso de creación de un organigrama, permitiendo importar datos desde sistemas externos, así como automatizar la distribución de la información a formatos estandarizados (Microsoft PowerPoint o Word), a la web o a copias impresas. Puede utilizarse incluso para realimentar información actualizada en su sistema informático de gestión empresarial [8].

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.

1.2.4 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. Así mismo, 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. 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. Cuenta con un gran conjunto de plantillas de diseño profesional adaptables a las necesidades de la organización. 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 [9].

Características:

- Variedad de formas inteligentes
- Colaboración en tiempo real o asíncrona
- Exportar y compartir documentos en múltiples formatos simples
- Variedad de bibliotecas
- Gran velocidad.

1.2.5 Conclusiones del estudio de homólogos

De los sistemas homólogos estudiados se puede decir que todos presentan ventajas que pueden ser utilizadas en el desarrollo de la propuesta de solución. El sistema Akademos específicamente presenta funcionalidades que con algunas mejoras pueden ser utilizadas para la creación de este módulo. De los otros sistemas analizados serán utilizadas algunas funcionalidades para realizar organigramas ya que poseen gran facilidad de creación.

Sistemas homólogos	Privativo	Procedencia	Establecer estructura	Gestionar centro	Web
Akademos	No	Nacional	Si	No	Si
Cedrux	No	Nacional	Si	No	Si
OrgPlus	Si	Internacional	Si	No	No
Creately	Si	Internacional	Si	No	Si

Tabla 1: Comparación de los sistemas homólogos

1.3 Tecnologías

Para el desarrollo de la solución informática el Centro de Tecnologías para la Formación (FORTES) propuso utilizar como entorno de trabajo Django 3.0 para el backend y Angular 8.2 para el frontend, ya que es de código abierto:

Django es un framework web diseñado para realizar aplicaciones de cualquier complejidad en unos tiempos muy razonables. Está escrito en Python y tiene una comunidad muy amplia, que está en continuo crecimiento. Los motivos principales para usar Django son:

- Es muy rápido
- Viene bien cargado
- Es seguro
- Es escalable
- Es increíblemente versátil

Está dotado de una gran librería con funcionalidades dirigidas para ayudar al programador a ahorrar tiempo en ejecutar ciertas tareas. Entre la gran variedad de librerías que existen se destacan funciones como el envío de correos, leer datos de páginas web o también trabajar con archivos comprimidos [10].

Angular es un framework JavaScript, destinado a facilitar la creación de aplicaciones web modernas. Su primera versión, AngularJS (Angular JavaScript), se convirtió en muy poco tiempo en el estándar para el desarrollo de aplicaciones web avanzadas.

Angular utiliza como lenguaje de programación principal TypeScript, un súper conjunto de JavaScript/ECMAScript que facilita mucho el desarrollo. A continuación, se exponen las principales ventajas de utilizar Angular, ideal para el desarrollo empresarial [11].

Ventajas de usar Angular

- Se usa como lenguaje a TypeScript, esto aumenta la consistencia del código fuente.
- Es fácil mantener las aplicaciones. Como se usa TypeScript, cualquier cambio que se necesite hacer a una aplicación podrá realizarse rápidamente y sin cometer errores.
- Se usan componentes web. Un componente web es una porción de código que se puede reutilizar en otros proyectos hechos con Angular. Simplemente se puede agregar al proyecto el componente web que se necesita y listo, ese componente web pudo haber sido escrito por terceros o por nosotros mismos.
- El futuro es estable. Con JavaScript uno de los grandes problemas es que existen muchas bibliotecas y muchas versiones de algunas de esas bibliotecas, incompatibles entre sí. En Angular todo está integrado y no necesitas de bibliotecas de terceros para realizar tareas.

1.3.1 Arquitectura del sistema

A continuación, se hablará de las herramientas y lenguajes que se estarán utilizando en la realización de la aplicación:

Nombre de la herramienta: **PyCharm**

Es un IDE, es decir, no solo es un editor de código, sino que también tiene un depurador, un intérprete y otras herramientas que ayudarán a crear y exportar los programas. Tiene un intérprete en el editor de código que nos ayudará a saber o conocer los posibles errores del código en tiempo real, algo que ha hecho que Python y PyCharm sean elegidos por muchos usuarios que comienzan a programar.

PyCharm no solo funciona correctamente con archivos de Python, sino que también admite otros lenguajes de programación como JavaScript, kotlin o CoffeeScript y otras herramientas como HTML o CSS [12].

Nombre de la herramienta: **Visual Paradigm for UML 8.0**

Es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación [13].

Nombre de la herramienta: **PostgreSQL 10.0**

Es un sistema de gestión de bases de datos relacional, gratuito y de código abierto, que últimamente está ganando un importante protagonismo al ser muy sencillo de utilizar y tener mejor rendimiento que otros gestores como, por ejemplo, MySQL o MariaDB [14].

Algunas de las ventajas al utilizar PostgreSQL 10.0 son:

- Mejora en el proceso de autenticación, que ahora es mucho más seguro que antes, gracias al nuevo sistema de autenticación SCRAM-SHA-256.
- La nueva replicación lógica permite a PostgreSQL enviar cambios a las bases de datos de forma rápida y segura implementando un mecanismo contra fallos y errores que permitirá a las bases de datos recuperarse en caso de fallo o error a la hora de cambiar sus parámetros. Esta nueva función también ayudará a muchos administradores a la hora de migrar sus bases de datos.
- La replicación síncrona también ha mejorado, y ahora los cambios en las bases de datos primarias pueden realizarse de forma mucho más segura.
- Todo el sistema de particionado declarativo de tabla ha sido mejorado, permitiendo a los usuarios crear y mantener rasgos y listas de tablas de forma mucho más sencilla de como se hacía hasta ahora.
- PostgreSQL 10 ha mejorado uno de sus principales puntos débiles, las consultas en paralelo, que ahora es capaz de devolver resultados de forma mucho más rápida.

Nombre de la herramienta: **PgAdmin 3.0**

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados. Incluye:

- Interfaz administrativa gráfica
- Herramienta de consulta SQL (con un EXPLAIN gráfico)

- Editor de código procedural
- Agente de planificación SQL/shell/batch
- Administración de Slony-I

Se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

La interface gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft, Windows, Linux, FreeBSD, Mac OSX y Solaris [15].

Nombre de la herramienta: **Nginx 1.14**

Es un famoso software de servidor web de código abierto. En su versión inicial, funcionaba en servidores web HTTP. Sin embargo, hoy en día también sirve como proxy inverso, balanceador de carga HTTP y proxy de correo electrónico para IMAP, POP3 y SMTP. NGINX ofrece una arquitectura asíncrona y controlada por eventos, característica que hace de NGINX uno de los servidores más confiables para la velocidad y la escalabilidad. Debido a su excelente capacidad para manejar muchas conexiones y a su velocidad, muchos sitios web de alto tráfico usan el servicio de NGINX. Algunos de estos gigantes del internet son Google, Netflix, Adobe, Cloudflare, WordPress.com, entre otros [16].

Nombre del lenguaje: **Lenguaje Unificado de Modelado (UML) 2.0**

Fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software orientada a objetos, representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios.

No es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. Guarda una relación directa con el análisis y el diseño orientados a objetos.

Nombre del lenguaje: **HTML5**

HTML (*Hyper Text Markup Language*), es un lenguaje de marcado usado para estructurar y presentar el contenido para la web.

Sirve para estructurar documentos (títulos, párrafos, listas, etc.), pero no describe la apariencia o el diseño de un documento, sino que ofrece las herramientas necesarias para dar formato, según la capacidad del servidor web en el que se almacenan las páginas web y la capacidad del navegador (tamaño de la pantalla, fuentes que tiene instaladas, etc.). Tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación web: su compatibilidad y la facilidad que plantea su aprendizaje debido al reducido número de etiquetas en las que se apoya.

Nombre del lenguaje: **CSS3**

CSS ("*Cascade StyleSheets*", traducido al español como "Hojas de Estilo en Cascada") es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. Se utiliza para definir el aspecto de todos los contenidos, el formato de tablas, la separación, el color, tamaño y tipo de letra de titulares y/o textos, la tabulación con la que se muestran los elementos de una lista o menú.

Nombre del lenguaje: **Python 3.0**

Es un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por su código legible y limpio. Una de las razones de su éxito es que cuenta con una licencia de código abierto que permite su utilización en cualquier escenario. Entre otras de sus características la automatización de procesos y ejecución de tareas, tanto en entorno cliente como servidor. Python es ideal para trabajar con grandes volúmenes de datos ya que, el ser multiplataforma, favorece su extracción y procesamiento [17].

Nombre del lenguaje: **TypeScript 2.0**

Es un lenguaje de programación de código abierto desarrollado por Microsoft, el cual cuenta con herramientas de programación orientada a objetos, muy favorable si se tienen proyectos grandes. TypeScript convierte su código en JavaScript común. Es llamado también súper conjunto de JavaScript, lo

que significa que, si el navegador está basado en JavaScript, este nunca llegará a saber que el código original fue realizado con TypeScript y ejecutará el JavaScript como lenguaje original.

1.3.2 Metodología de desarrollo de software

Centro de Tecnologías para la Formación propone la utilización de la metodología AUP en su variante UCI para guiar el proceso de desarrollo del módulo de estructura y composición para el sistema de gestión académica del Ministerio de Educación de la República de Cuba.

Existe la creencia extendida de que los programas de software tienen que ver solamente con los ordenadores y las grandes computadoras. Sin embargo, la ingeniería del software va mucho más allá. Se trata del proceso cuya finalidad es desarrollar productos o soluciones para un cliente en particular, teniendo en cuenta factores como los costos, la planificación, la calidad y las dificultades asociadas. A todo esto, es a lo que se denomina metodologías de desarrollo de software. Es decir, se trata del proceso que se suele seguir a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas. O para ser más precisos, las metodologías de desarrollo de software son enfoques de carácter estructurado y estratégico que permiten el desarrollo de programas con base a modelos de sistemas, reglas, sugerencias de diseño y guías. Estas metodologías se dividen en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y tradicionales (o robustas) [18].

Metodología ágil

Es una de varias metodologías de desarrollo de software basadas en el desarrollo iterativo e incremental, en contraposición a las metodologías tradicionales de desarrollo de software lineal o cascada. Una metodología ágil proporciona un marco para apoyar el diseño, desarrollo y pruebas de soluciones de software a lo largo de su ciclo de vida.

Los métodos ágiles ofrecen respuestas rápidas y flexibles al cambio mediante la promoción de la planificación adaptativa, la identificación de requisitos colaborativos y la racionalización entre el equipo interfuncional autoorganizado, así como el desarrollo gradual de soluciones. Muchos esfuerzos de desarrollo de software comerciales modernos siguen este tipo de enfoque.

Metodología Tradicional

Estas metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

Metodología AUP-UCI

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP [19].

El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (test driven development - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad.

La Metodología AUP en su variación UCI está formada por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad, las cuales contienen las características de las cuatro fases (Inicio, Elaboración, Construcción y Transición) propuestas en AUP. Entre sus disciplinas están:

Modelado de negocio:

Para modelar el negocio se proponen las siguientes variantes:

1. Casos de Uso del Negocio (CUN)
2. Descripción de Proceso de Negocio (DPN)
3. Modelo Conceptual (MC)

Requisitos:

Existen tres formas de encapsular los requisitos

1. Casos de Uso del Sistema (CUS)
2. Historias de usuario (HU)
3. Descripción de requisitos por proceso (DRP)

Análisis y diseño:

En esta disciplina, si se considera necesario, los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura).

Implementación:

En la implementación, a partir de todos los análisis realizados y diseño se comienza a construir el sistema.

Pruebas:

Estas se dividen en tres:

1. Pruebas internas
2. Pruebas de liberación
3. Pruebas de aceptación

Escenario No 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad. Se debe tener presente que este escenario

es muy conveniente si se desea representar una gran cantidad de niveles de detalles y la relaciones entre los procesos identificados.



Figura 1. Escenario 3 de la metodología AUP-UCI

El Centro de Tecnologías para la Formación propuso la utilización de este escenario, pero ya que el módulo de estructura y composición no tiene procesos concretos, no se utilizará la Descripción de Proceso del Negocio (DPN) para el modelado de negocio, se utilizará el Modelado Conceptual (MC) para describir las relaciones existentes entre los principales conceptos asociados a la estructura y composición, y se especificarán y describirán los requisitos a través de Descripción de Requisitos por Proceso (DRP).

1.4 Conclusiones del capítulo

Los conceptos asociados al objeto de estudio permitieron obtener un mejor entendimiento del negocio y los conocimientos necesarios para el desarrollo de la propuesta de solución. Después de realizar un análisis a sistemas similares se concluyó que estos aportaron ideas y características para realizar nuevas funcionalidades en la propuesta de solución, así como la obtención de organigramas. Por otro lado, la aplicación del escenario número 3 de la metodología AUP-UCI permitió guiar el proceso de desarrollo. Se definieron las tecnologías, herramientas y lenguajes a utilizar, siendo estas las más adecuadas para el desarrollo del proyecto.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se abordan las características generales del sistema a desarrollar, es vital el análisis del modelo de dominio para una mejor comprensión del entorno de la investigación. Luego se describe la propuesta de solución, se representa un mapa conceptual con los principales conceptos y relaciones entre ellos. Se explican los requisitos funcionales y no funcionales, así como las técnicas para su obtención. Se describe la arquitectura, el diseño del sistema y los patrones utilizados. Se muestra el diagrama lógico de datos, las principales vistas del sistema, el diagrama de despliegue y el mapa de navegación.

2.1 Descripción de la propuesta de solución

Para poder cumplir con una solución informática adecuada, en la cual el MINED pueda gestionar la información de cada una de sus estructuras, se propone la siguiente solución informática que permita:

- Gestionar las diferentes estructuras educacionales del país.
- Gestionar los diferentes inmuebles de cada centro educación, así como sus características.
- Gestionar la información acerca de la geolocalización de cada una de sus estructuras, mediante un mapa.
- Gestionar los niveles de cada estructura mediante el uso de organigramas.
- Gestionar diferentes responsabilidades asociadas a cada estructura, dependiendo del tipo de estructura.
- Gestionar los diferentes organismos formadores.
- Gestionar las diferentes enseñanzas.
- Gestionar las diferentes características asociadas a los inmuebles y los centros educacionales.
- Gestionar

2.2 Conceptualización del dominio

Para la modelación del negocio no se definieron procesos específicos del entorno, solo se identificaron conceptos y objetos relacionados con el mismo, es por ello que se propone la realización de un modelo de dominio. Con este modelo se pretende contribuir a la comprensión del negocio actual. Dirigirse a [Anexo 1](#).

2.3 Requisitos de Software

2.3.1 Requisitos Funcionales

Los requerimientos funcionales hacen referencia a la descripción de las actividades y servicios que un sistema debe proveer. Normalmente este tipo de requerimientos están vinculados con las entradas, las salidas de los procesos y los datos a almacenar en el sistema.

Tabla 2: Requisitos funcionales

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Incluir estructura	El requisito permite seleccionar e introducir los datos necesarios para crear una estructura.	Media	Media
RF2	Listar estructuras	El requisito muestra una lista con las estructuras existentes.	Baja	Baja
RF3	Modificar estructura	El requisito permite modificar los datos de una estructura.	Media	Baja
RF4	Ver detalles estructura	El requisito muestra la información de una estructura.	Baja	Baja
RF5	Filtrar estructura	El requisito permite obtener un conjunto de estructuras a partir de la introducción de un conjunto de parámetros.	Baja	Baja

RF6	Buscar estructura	El requisito permite encontrar una estructura a partir del nombre.	Baja	Baja
RF7	Asociar responsabilidad a estructura	El requisito permite asociar a cada estructura las responsabilidades correspondientes.	Baja	Baja
RF8	Generar organigrama de estructuras	El requisito permite generar un organigrama con todas las estructuras creadas según sus responsabilidades y categorías	Baja	Media
RF9	Incluir responsabilidad	El requisito permite seleccionar e introducir los datos necesarios para crear una responsabilidad.	Media	Media
RF10	Listar responsabilidad	El requisito muestra una lista con las responsabilidades existentes.	Baja	Baja
RF11	Modificar responsabilidad	El requisito permite modificar los datos de una responsabilidad.	Baja	Baja
RF12	Ver detalles responsabilidad	El requisito muestra la información de una responsabilidad.	Baja	Baja
RF13	Filtrar responsabilidad	El requisito permite obtener un conjunto de responsabilidades a partir de la introducción de un conjunto de parámetros.	Baja	Baja
RF14	Buscar responsabilidad	El requisito permite encontrar una responsabilidad a partir del nombre.	Baja	Baja
RF15	Incluir inmueble	El requisito permite seleccionar e introducir los datos necesarios para crear un inmueble.	Alta	Media
RF16	Listar inmueble	El requisito muestra una lista con los inmuebles existentes.	Baja	Baja

RF17	Modificar inmueble	El requisito permite modificar los datos de un inmueble.	Baja	Baja
RF18	Ver detalles inmueble	El requisito muestra la información de un inmueble.	Baja	Baja
RF19	Filtrar inmueble	El requisito permite obtener un conjunto de inmuebles a partir de la introducción de un conjunto de parámetros.	Baja	Baja
RF20	Buscar inmueble	El requisito permite encontrar un inmueble a partir del nombre.	Baja	Baja
RF21	Incluir organismos formadores	El requisito permite seleccionar e introducir los datos necesarios para crear un organismo formador.	Media	Media
RF22	Listar organismos formadores	El requisito muestra una lista con los organismos formadores existentes.	Baja	Baja
RF23	Modificar organismos formadores	El requisito permite modificar los datos de un organismos formadores.	Baja	Baja
RF24	Buscar organismos formadores	El requisito permite encontrar una organismo formador a partir del nombre.	Baja	Baja
RF25	Incluir enseñanza	El requisito permite seleccionar e introducir los datos necesarios para crear una enseñanza.	Media	Media
RF26	Listar enseñanza	El requisito muestra una lista con las enseñanzas existentes.	Baja	Baja
RF27	Modificar enseñanza	El requisito permite modificar los datos de una enseñanza.	Baja	Baja
RF28	Buscar enseñanza	El requisito permite encontrar una enseñanza a partir del nombre.	Baja	Baja

RF29	Incluir características	El requisito permite seleccionar e introducir los datos necesarios para crear una característica.	Media	Media
RF30	Listar características	El requisito muestra una lista con las características existentes.	Baja	Baja
RF31	Modificar características	El requisito permite modificar los datos de una característica.	Baja	Baja
RF32	Ver detalles características	El requisito muestra la información de una característica.	Baja	Baja
RF33	Buscar características	El requisito permite encontrar una característica a partir del nombre.	Baja	Baja
RF34	Filtrar características	El requisito permite obtener un conjunto de características a partir de la introducción de un conjunto de parámetros.	Baja	Baja
RF35	Incluir categorías	El requisito permite seleccionar e introducir los datos necesarios para crear una categoría.	Media	Media
RF36	Listar categorías	El requisito muestra una lista con las categorías existentes.	Baja	Baja
RF37	Modificar categorías	El requisito permite modificar los datos de una categoría.	Baja	Baja
RF38	Ver detalles categorías	El requisito muestra la información de una categoría.	Baja	Baja
RF39	Buscar categorías	El requisito permite encontrar una categoría a partir del nombre.	Baja	Baja
RF40	Asociar categoría a una responsabilidad	El requisito permite asociar a cada responsabilidad la categoría correspondiente.	Baja	Baja

RF41	Incluir país	El requisito permite seleccionar e introducir los datos necesarios para crear un país.	Baja	Baja
RF42	Listar país	El requisito muestra una lista con los país existentes.	Baja	Baja
RF43	Modificar país	El requisito permite modificar los datos de un país.	Baja	Baja
RF44	Buscar país	El requisito permite encontrar un país a partir del nombre.	Baja	Baja

2.3.2 Requisitos no Funcionales

Los requerimientos no funcionales engloban características como rendimiento, facilidad de uso, presupuestos, tiempo de entrega, documentación, seguridad y auditorías internas. Son restricciones de los servicios o funciones ofrecidos por el sistema.

A continuación, se enuncian los requisitos no funcionales identificados:

Usabilidad

RNF1 El sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo para usuarios sin experiencia y el acceso de manera rápida y efectiva a la información buscada.

RNF2 El sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a mayor comprensión por parte del cliente de la herramienta de trabajo.

RNF3 El sistema debe presentar un menú lateral que permita el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.

Confiabilidad

RNF4 Solo el personal autorizado para interactuar con la aplicación podrá utilizarla. Para lograr esto se necesitará de un nombre de usuario único y una contraseña definida anteriormente.

Eficiencia

RNF5 El sistema debe tener un tiempo de respuesta corto.

RNF6 El sistema debe soportar la conexión simultánea de todos los posibles usuarios.

Soporte

RNF7 El sistema contará con un grupo de soporte y asesoría al cliente que proporcionará soporte al sistema siempre que sea necesario.

RNF8 El sistema debe brindar como apoyo un manual de usuario, en el cual se refleja detalladamente la explicación de cada una de sus funcionalidades, brindándole al usuario una mayor experiencia del trabajo con el mismo.

RNF9 El componente contará con toda la documentación definida en el expediente de proyecto asociada a su proceso de desarrollo para las actividades de soporte.

RNF10 Se precisa que la documentación del sistema este actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.

Restricciones de diseño e implementación

RNF11 El sistema debe cumplir con la arquitectura de información definida para el Sistema de Gestión Académica.

RNF12 El sistema debe ser desarrollado con las herramientas y tecnologías definidas por el Centro de Formación para las Tecnologías.

RNF13 El sistema debe ser desarrollado con herramientas y tecnologías de código abierto.

2.4 Descripción de requisitos por procesos

El levantamiento y descripción de los procesos es una forma de representar la realidad de la manera más exacta posible, a partir de la identificación de las diferentes actividades y tareas que se realizan en un proceso, para lograr un determinado resultado o producto. Diríjase a [Anexo 22](#).

2.5 Descripción de la arquitectura

Un patrón arquitectónico es una solución general y reutilizable a un problema común en la arquitectura de software dentro de un contexto dado. Los patrones arquitectónicos son similares al patrón de diseño de software, pero tienen un alcance más amplio.

Cliente-Servidor

Este patrón consiste en dos partes; un servidor y múltiples clientes. El componente del servidor proporcionará servicios a múltiples componentes del cliente. Los clientes solicitan servicios del servidor y el servidor proporciona servicios relevantes a esos clientes. Además, el servidor sigue escuchando las solicitudes de los clientes [20].

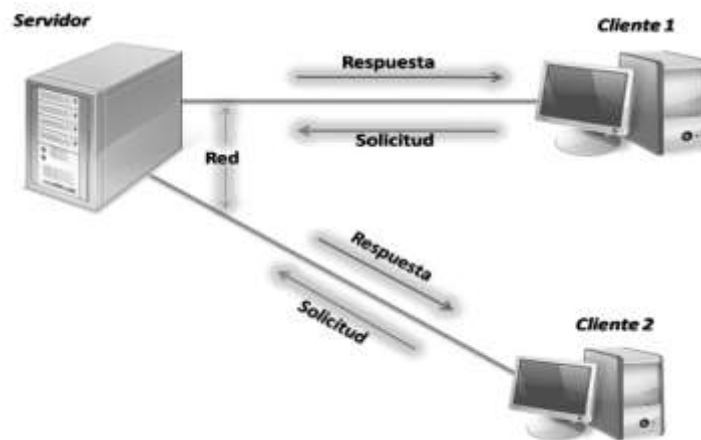


Figura 2: Arquitectura Cliente-Servidor

2.5.1 Patrones arquitectónicos

Patrón de diseño arquitectónico que utiliza Angular.

Angular no tiene un modelo-vista-controlador (MVC) clásico, sino que, el modelo tiene mucha relación con la vista.

Esto es así por el concepto base de Angular de enlace de datos bidireccional, ya que la forma de sincronizar los datos entre la vista y el modelo-vista es totalmente dependiente, es decir, en la vista se puede modificar el modelo y en el modelo se puede modificar la vista.

Esto hace que la independencia que se produce en un modelo-vista-controlador clásico aquí no se produzca, y por lo tanto tiende a llamarse modelo-vista vista-modelo (MVVM) o modelo-vista-whatever (MVW), porque no se sabe muy bien dónde identificarlo [21].

Patrón de diseño arquitectónico que utiliza Django.

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. Someramente, la M, V y C se separan en Django de la siguiente manera:

- *M*, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.
- *V*, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.
- *C*, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URLconf y llamando a la función apropiada de Python para la URL obtenida.

Debido a que la "C" es manejada por el mismo framework y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un *Framework MTV*. En el patrón de diseño MTV,

- *M* significa "Model" (Modelo), la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos.
- *T* significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento.

- V significa "View" (Vista), la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada: puedes pensar en esto como un puente entre el modelo y las plantillas.

Ninguna de las interpretaciones es más "correcta" que otras. Lo importante es entender los conceptos subyacentes [22].

2.5.2 Patrones de diseño

Los patrones de diseño son técnicas para resolver problemas comunes en el desarrollo de software. Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas de software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modelo en que se realiza e diseño.
- Facilitar el aprendizaje a las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Entre los patrones de diseño más conocidos están los Patrones de Principios Generales para Asignar Responsabilidades (GRASP) y los patrones de la Gang of Four (GoF).

GRASP: Principios Generales para Asignar Responsabilidades

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprehender (grasping en inglés) estos principios para diseñar con éxito el software orientado a objetos. Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones [23].

A continuación, se presentan los patrones GRASP utilizados.

- **Experto:** El GRASP de experto en información es el principio básico de asignación de responsabilidades. Determina cuál es la clase que debe asumir una responsabilidad a partir de la información que posee. O sea, la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Este patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases StructureCreateComponent.ts que se encarga de la creación de una estructura.

Creador: Ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Se identifica al asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes:

1. B agrega objetos de A.
2. B contiene objetos de A.
3. B registra instancias de objetos de A.
4. B tiene los datos de inicialización que se pasarán a un objeto de A cuando sea creado.
5. B es un creador de los objetos A.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Patrones GOF

Los patrones que se presentan proceden de Design Patterns, un libro básico y muy popular 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". Describen las formas comunes en que objetos de tipos diferentes pueden organizarse para trabajar entre sí [24].

Los patrones GoF utilizados en el desarrollo de la aplicación son:

Mediador (Mediator): Este patrón se evidencia en las librerías, las cuales son mediadoras entre las clases controladoras y los modelos o acceso a datos. Se evidencia en la clase `models.py` donde se hace uso de la librería `models` para declarar el tipo de dato de los diferentes atributos.

Instancia única (Singleton): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto.

La clase `school.component.ts` la cual tiene en su constructor una instancia `NgxPermissions` `“NgxPermissionsService”`. De esta manera no hay necesidad de instanciar a `NgxPermissions` en cualquier otra parte de la clase. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

2.6 Modelo de datos

El modelado de datos es el proceso de documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando texto y símbolos para representar la forma en que los datos necesitan fluir. El diagrama se puede utilizar como un mapa para la construcción de un nuevo software o para la reingeniería de una aplicación antigua [25].

Una opción bastante usada a la hora de clasificar los modelos de datos es hacerlo de acuerdo al nivel de abstracción que presentan:

Modelos de Datos Conceptuales

Son los orientados a la descripción de estructuras de datos y restricciones de integridad. Se usan fundamentalmente durante la etapa de Análisis de un problema dado y están orientados a representar los elementos que intervienen en ese problema y sus relaciones. El ejemplo más típico es el Modelo Entidad-Relación.

Modelos de Datos Lógicos

Son orientados a las operaciones más que a la descripción de una realidad. Usualmente están implementados en algún Manejador de Base de Datos. El ejemplo más típico es el Modelo Relacional, que cuenta con la particularidad de contar también con buenas características conceptuales (Normalización de bases de datos).

Modelos de Datos Físicos

Son estructuras de datos a bajo nivel implementadas dentro del propio manejador. Ejemplos típicos de estas estructuras son los Árboles B+, las estructuras de Hash, etc.

Diríjase a [Anexo 3](#).

2.7 Modelo de despliegue

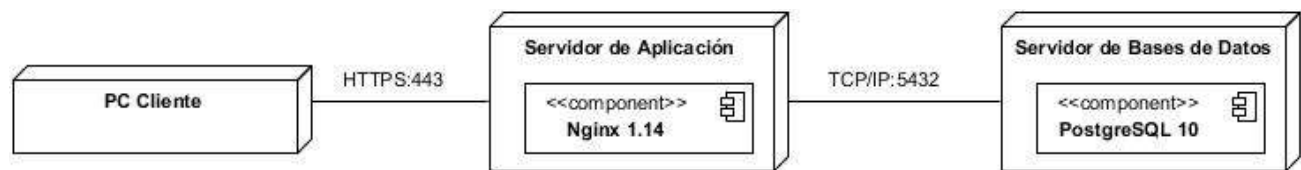


Figura 2: Diagrama de despliegue

2.8 Conclusiones del capítulo

En este capítulo se abordaron las características generales del sistema a desarrollar, así como la descripción de la propuesta de solución. La conceptualización del dominio se hizo imprescindible para un mejor entendimiento del sistema, al igual que sus requisitos. Abordando también los patrones arquitectónicos y de diseño, se realizó el modelo de datos, así como el diagrama de despliegue.

CAPÍTULO 3: Implementación y pruebas de la propuesta de solución

En el presente capítulo se define el diagrama de componentes, los estándares de codificación con los que debe cumplir el equipo de desarrollo durante la implementación, así como los métodos y técnicas para la realización de pruebas. El proceso de pruebas está dirigido a componentes del software, con el objetivo de medir el grado en que se cumplen los requisitos exigidos por el cliente y si posee la calidad requerida.

3.1 Diagrama de Componente

El diagrama de componentes representa la arquitectura del sistema basado en la dependencia entre sus componentes y permitirá tomar decisiones con respecto a las tareas de implementación. Diríjase a [Anexo 5](#).

3.2 Estándares de Codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de códigos. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, se debe establecer un estándar de codificación, para asegurarse que todos los programadores del proyecto trabajen de forma coordinada. En el desarrollo del proyecto se definieron los siguientes estándares por el Centro de Tecnologías para la Formación (FORTES).

- Se debe utilizar como idioma el inglés.
- Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas.
- Se deben usar siempre dos líneas en blanco en las siguientes circunstancias:
 - Entre las secciones de un fichero fuente.
 - Entre las definiciones de clases e interfaces.
- Se debe usar siempre una línea en blanco en las siguientes circunstancias:
 - Entre métodos
 - Entre las variables locales de un método y su primera sentencia.
 - Antes de un comentario de bloque o de un comentario de una línea.
 - Entre las distintas secciones lógicas de un método para facilitar la lectura.
- Se debe dar un espacio en blanco en la siguiente situación:
 - Entre palabras claves del lenguaje y un paréntesis.
- Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos.

- Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales.
- Los nombres de las variables declaradas deben aparecer totalmente en minúsculas separando las palabras con un guion bajo (“_”).
- Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas.
- Los nombres de las clases deben ser simples y descriptivos. Usar palabras completas, evitar acrónimos y abreviaturas.
- Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúsculas y la primera letra de la siguiente palabra que lo forma en mayúsculas.
- Evitar líneas de más de ochenta caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

3.3 Estrategias de Pruebas

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre localización del producto a la parte interesada. Es una actividad más en el proceso de control de calidad. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo [26].

El objetivo principal de las pruebas es aportar calidad al producto que se está desarrollando:

¿Qué es calidad?

Este término es definido por muchas organizaciones en el mundo de la calidad de formas diferentes ejemplos de estas son las siguientes:

- ISO 9000 Es un conjunto de normas sobre calidad y gestión de calidad, la define como “La calidad es el grado en el que un conjunto de características inherentes cumple con los requisitos “.

- ISO 25000 es un conjunto de norma que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto de software, dice: “la calidad es el grado en el que el producto de software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas [27].

3.3.1 Pruebas de rendimiento

Verifican cómo responde el sistema cuando éste se encuentra bajo una alta carga.

Estas pruebas son no-funcionales, y pueden tener diversas formas para entender

- La fiabilidad
- Estabilidad
- Disponibilidad de la plataforma

Por ejemplo, pueden observar los tiempos de respuesta cuando se ejecuta un alto número de peticiones (consultas al servidor), o ver cómo se comporta el sistema ante una cantidad significativa de datos.

Las pruebas de rendimiento son, por su naturaleza, bastante costosas de implementar y ejecutar, pero pueden ayudarnos a comprender si nuevos cambios van a degradar el sistema (como hacerlo más lento o aumentar su consumo de recursos).

Las pruebas de rendimiento no fallan del mismo modo en que lo hacen las demás pruebas. En vez de ello su objetivo es recolectar métricas y definir objetivos por alcanzar.

Generalmente es buena idea realizar pruebas de este tipo ante nuevos lanzamientos y/o refactorizaciones importantes en el código [28].

3.3.2 Pruebas de aceptación

Son pruebas formales, ejecutadas para verificar si un sistema satisface sus requerimientos de negocio.

Estas pruebas requieren que el software se encuentre en funcionamiento, y se centran en replicar el comportamiento de los usuarios, a fin de rechazar cambios si no se cumplen los objetivos. Estos objetivos pueden ir más allá de obtener una respuesta específica, y medir el rendimiento del sistema.

- Son usualmente un conjunto de pruebas manuales que se realizan luego de que una fase de desarrollo ha finalizado (de modo que se pueda volver rápidamente e iterar si algo no está correcto).
- Verifican que las características del software estén alineadas con todas las especificaciones iniciales y criterios de aceptación.
- Suelen realizarse luego de las pruebas unitarias o de integración, para evitar que se avance mucho con el proceso de prueba, y determinar a tiempo si se necesitan cambios significativos.

Para que este tipo de pruebas se lleve a cabo correctamente resulta importante que los responsables del proyecto definan los criterios de aceptación justo antes de empezar a trabajar en el mismo. Así mismo, cualquier requerimiento adicional que surja durante el proceso deberá verse reflejado en tales criterios de aceptación [29].

3.3.3 Pruebas unitarias

Son a bajo nivel (cercanas al código fuente de nuestra aplicación).

Este tipo de pruebas consiste en probar de forma individual las funciones y/o métodos (de las clases, componentes y/o módulos que son usados por el software). Debido a lo específicas que son, generalmente son las pruebas automatizadas de menor coste, y pueden ejecutarse rápidamente por un servidor de integración continua.

Más detalles acerca de las pruebas unitarias:

- Idealmente, cuando se planea y se describen las pruebas unitarias, se debe aislar la funcionalidad hasta un punto en que no se pueda desglosar más, y entonces escribir pruebas a partir de ello. Justamente, el nombre de este tipo de pruebas hace referencia a una "unidad de código", que es independiente del resto.
- Estas pruebas verifican que el nombre de la función o método sea adecuado, que los nombres y tipos de los parámetros sean correctos, y así mismo el tipo y valor de lo que se devuelve como resultado.

- Dado que las pruebas unitarias no deben tener ningún tipo de dependencia, se suele reemplazar los llamados a APIs y servicios externos por funcionalidad que los imite (para que no exista interacción que vaya más allá de la unidad que está siendo probada).

En muchos casos inclusive se suele reemplazar las consultas a bases de datos, de modo que la prueba se enfoque en operar a partir de los valores de entrada, sin depender de ninguna fuente externa.

- Si no es factible aislar el uso de bases de datos de nuestras pruebas unitarias, será importante tener en cuenta el rendimiento y buscar optimizar nuestras consultas.

Esto es importante, porque si nuestras pruebas unitarias son de larga duración, resultará incómodo ejecutarlas y ralentizará significativamente los tiempos de desarrollo [30].

3.3.4 Pruebas de integración

Verifican que los diferentes módulos y/o servicios usados por nuestra aplicación funcionen en armonía cuando trabajan en conjunto. Por ejemplo:

- Pueden probar la interacción con una o múltiples bases de datos,
- Asegurar que los micro servicios operen como se espera.

Las pruebas de integración son típicamente el paso siguiente a las pruebas unitarias.

Y son generalmente más costosas de ejecutar, ya que requieren que más partes de nuestra aplicación se configuren y se encuentren en funcionamiento.

3.3.5 Pruebas de seguridad y control de acceso

El objetivo de esta prueba es verificar que un actor solo pueda acceder a las funciones y datos que su usuario tenga permitido, y que solo los actores con acceso al sistema y a la aplicación estén habilitados para accederla [31].

3.4 Estrategia de Validación (Técnica de IADOV)

Esta técnica se compone de cinco preguntas: tres cerradas y dos abiertas, las cuales se reformulan en la investigación para valorar el grado de satisfacción de los clientes sobre un tema en específico. Una vez establecidas las preguntas se conforma el cuadro lógico de IADOV y el número resultante de la interrelación de las tres preguntas, indica la posición de los sujetos en la escala de satisfacción [32].

La escala de satisfacción está dada por los criterios:

1. Máxima satisfacción.
2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Máxima insatisfacción.
6. Contradictoria.

Diríjase a [Anexo 4 Tabla 23](#).

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y -1. Diríjase a [Anexo 4 Tabla 24](#).

La satisfacción grupal (ISG) se calcula por la siguiente fórmula:

$$ISG = \frac{A(+1) + B(+0,5) + C(0) + D(-0,5) + E(-1)}{N}$$

Donde:

- A representa el número de sujetos con índice individual 1
- B representa el número de sujetos con índice individual 2
- C representa el número de sujetos con índice individual 3 ó 6
- D representa el número de sujetos con índice individual 4

- E representa el número de sujetos con índice individual 5
- N representa el número total de sujetos del grupo

3.5 Conclusiones del capítulo

El uso de los estándares de codificación como guía de desarrollo del sistema permitirá como resultado final una aplicación que cumpla con las necesidades de parte del cliente.

Las pruebas definidas para realizar al sistema permitirán obtener una aplicación web con seguridad y calidad.

Conclusiones Generales

Con la presente investigación se llega a la conclusión de que:

- Los conceptos asociados al objeto de estudio permitieron obtener un mejor entendimiento del negocio y los conocimientos necesarios para el desarrollo de la propuesta de solución.
- Después de realizar un análisis a sistemas similares se concluyó que estos aportaron ideas y características para realizar nuevas funcionalidades en la propuesta de solución, así como la obtención de organigramas.
- La aplicación del escenario número 3 de la metodología AUP-UCI permitió guiar el proceso de desarrollo.
- Se definieron las tecnologías, herramientas y lenguajes a utilizar, siendo estas las más adecuadas para el desarrollo del proyecto.
- La correcta realización del análisis y diseño del módulo Estructura y Composición del sistema de gestión académica del MINED permitió lograr un mayor entendimiento de la propuesta de solución.

Recomendaciones

- Desarrollar el módulo Estructura y Composición del sistema de gestión académica del MINED.
- Realizar un mapa de geolocalización que permita, poder acceder con mayor facilidad cada uno de los centros educativos.

Bibliografía

- [1] J. Díaz Lazo, A. Pérez Gutiérrez, and R. Florido Bacallao, "IMPACTO DE LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES (TIC) PARA DISMINUIR LA BRECHA DIGITAL EN LA SOCIEDAD ACTUAL," *Cultivos Tropicales*, vol. 32, no. 1, pp. 81–90, Mar. 2011.
- [2] "¿Cuál es el significado de estructura organizacional?" [Online]. Available: <https://pyme.lavoztx.com/cul-es-el-significado-de-estructura-organizacional-4749.html>. [Accessed: 04-Dec-2019].
- [3] "Organigrama: Concepto, Tipos y Usos." [Online]. Available: <https://concepto.de/organigrama/>. [Accessed: 08-Dec-2019].
- [4] "url centro educacional.htm." .
- [5] "proceso de enseñanza aprendizaje - Buscar con Google." [Online]. Available: https://www.google.com/search?source=hp&ei=wc4lXsc9goLnApKdmvAK&q=proceso+de+ense%C3%B1anza+aprendizaje&oq=Procese+de+en&gs_l=psy-ab.3.0.0i13110.3221.6594..7723...0.0..0.163.1484.1j12.....0....1..gws-wiz.....0i131j0j0i10.9PK9FGBEQYA. [Accessed: 20-Jan-2020].
- [6] "Akademos.pdf." .
- [7] Y. Matos-Arias and N. Silega-Martínez, "Estilo Arquitectónico Para El Sistema Integrado De Gestión Cedrux (Architectonical Style for the Management Integrated System Cedrux)," *Social Science Research Network*, Rochester, NY, SSRN Scholarly Paper ID 2396840, Jul. 2013.
- [8] "Manual del usuario de OrgPlus 9." [Online]. Available: http://www.altula.com/support/orgplus9/es/help/OP9Help.htm#01Introduction_files/La_solucion_OrgPlus.htm. [Accessed: 08-Dec-2019].
- [9] "Características," 12-Jul-2019. [Online]. Available: </es/tour-de-productos/>. [Accessed: 09-Dec-2019].
- [10] "Qué es Django y por qué usarlo | OpenWebinars." [Online]. Available: <https://openwebinars.net/blog/que-es-django-y-por-que-usarlo/>. [Accessed: 19-Feb-2020].
- [11] "Ventajas de utilizar Angular, un framework JavaScript," *El Blog de Aitana – Partner Microsoft y Sage en España*, 10-Apr-2018. [Online]. Available: <https://blog.aitana.es/2018/04/10/ventajas-de-utilizar-angular/>. [Accessed: 04-Dec-2019].

- [12] "PyCharm, un potente IDE para crear programas con Python." [Online]. Available: <https://www.linuxadictos.com/pycharm-un-potente-ide-para-crear-programas-con-python.html>. [Accessed: 19-Feb-2020].
- [13] "Visual Paradigm - EcuRed." [Online]. Available: https://www.ecured.cu/Visual_Paradigm. [Accessed: 04-Dec-2019].
- [14] "PostgreSQL: ¿Qué es? Características, Ventajas y Desventajas." [Online]. Available: <https://hostingpedia.net/postgresql.html>. [Accessed: 04-Dec-2019].
- [15] "pgadmin." [Online]. Available: <http://es.tldp.org/Postgresql-es/web/navegable/user/app-pgadmin.html>. [Accessed: 04-Dec-2019].
- [16] G. B, "¿Qué Es NGINX Y Cómo Funciona?," *Tutoriales Hostinger*, 11-Apr-2019. .
- [17] "Qué es Python: Características, evolución y futuro | OpenWebinars." [Online]. Available: <https://openwebinars.net/blog/que-es-python/>. [Accessed: 19-Feb-2020].
- [18] "¿Qué son las metodologías de desarrollo de software? | OBS Business School." [Online]. Available: <https://obsbusiness.school/int/blog-project-management/metodologia-agile/que-son-las-metodologias-de-desarrollo-de-software>. [Accessed: 04-Dec-2019].
- [19] "urlmetodologia.htm." .
- [20] W. C. huaman, "Los 10 patrones comunes de arquitectura de software," *Medium*, 07-Sep-2018. [Online]. Available: <https://medium.com/@maniakhitoccori/los-10-patrones-comunes-de-arquitectura-de-software-d8b9047edf0b>. [Accessed: 27-Feb-2020].
- [21] "¿Qué patron usa Angular? MVC o MVVM," *OpenWebinars.net*, 16-Oct-2018. [Online]. Available: <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/>. [Accessed: 27-Feb-2020].
- [22] B. R. M. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. New York: McGraw-Hill Education, 2015.
- [23] RODRIGUEZ SANCHEZ, TAMARA., "Metodología de Desarrollo Para La Actividad Productiva En La UCI." 2015.
- [24] "5.2. El patrón de diseño MTV (El libro de Django 1.0)." [Online]. Available: <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>. [Accessed: 27-Feb-2020].
- [25] "¿Qué es Modelado de datos? - Definición en WhatIs.com," *SearchDataCenter en Español*. [Online]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Modelado-de-datos>. [Accessed: 12-Sep-2020].
- [26] "Pruebas-de-software.pdf."

- [27] "PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf." .
- [28] "Los diferentes tipos de Pruebas de software." [Online]. Available: <https://programacionymas.com/blog/tipos-de-testing-en-desarrollo-de-software>. [Accessed: 13-Sep-2020].
- [29] capacitación y guía para el desarrollo de software "PruebasSoftware.pdf."
- [30] "104-Texto del artículo-167-1-10-20190308.pdf."
- [31] I. Sommerville, *Ingeniería del software*. Pearson educación, 2005.
- [32] Dr. Alejandro López Rodríguez, "Niveles de satisfacción por las clases de educación física." [Online]. Available: <http://www.efdeportes.com/>.

Anexos

Anexo 1 Conceptualización del dominio

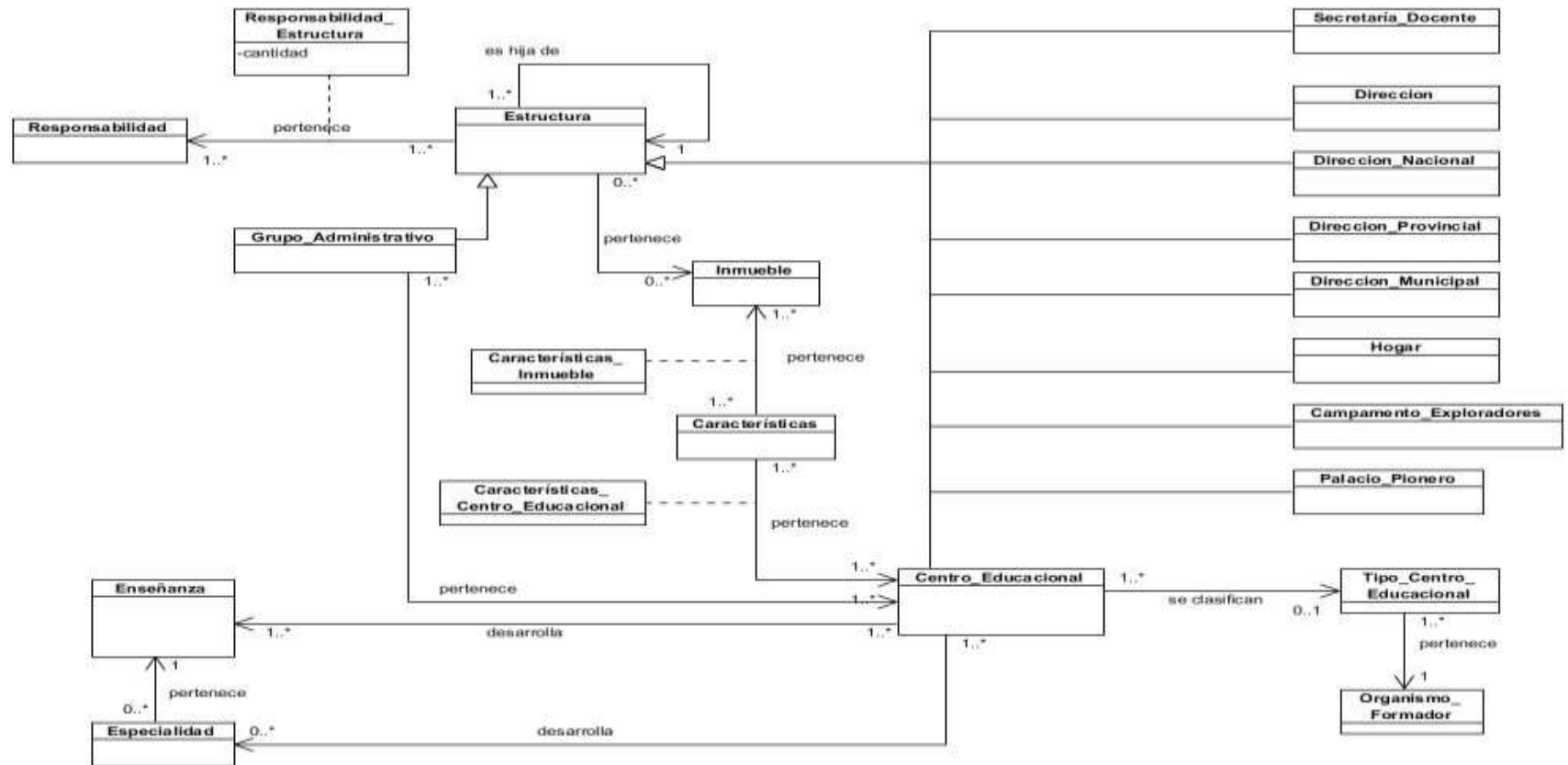


Figura 1. Modelado del Dominio

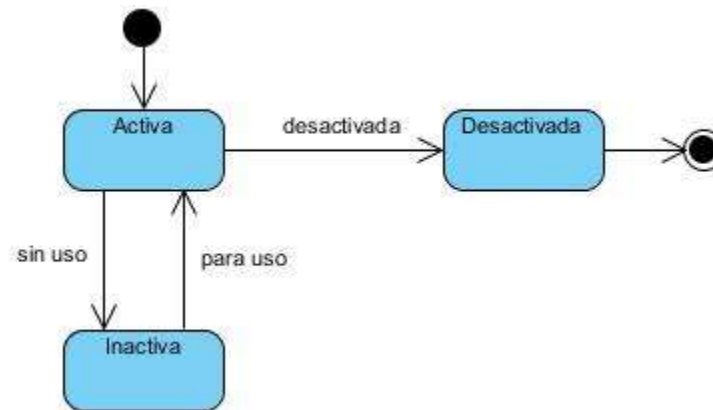
Anexo 2 Descripción del modelado de dominio

Tabla 1. Descripción del modelado de dominio. Estructura

Descripción		<i>Representa cada una de las entidades dentro de cada organismo formador.</i>			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
Clases válidas					
Clases no válidas		texto	no	U	- Dirección Nacional - Dirección provincia - Dirección municipal - Campamento - Centro - Hogar - Palacio - Dirección - Secretaria docente
-	Representa quien es el padre de la estructura	texto	no	U	

-	Representa el nombre de la estructura	texto	no	U	dirección municipal de matanzas
-	Representa las siglas con que serán representadas las estructuras	texto	no	u	
-	Representa la fecha cuando fue activada la estructura		no	u	Está compuesto por día, mes, año
-	Representa la fecha cuando fue desactivada la estructura		no	u	Está compuesto por día, mes, año
-	Representa el estado de la estructura: <ul style="list-style-type: none"> - activa - inactiva - desactivada 	texto	no	u	<ul style="list-style-type: none"> - activa - inactiva - desactivada

Estados de validez



Activa

Estado que toman las Estructuras que serán usadas por el negocio, para las cuales se podrán definir estructuras hijas, responsabilidades, etc.

Inactiva

Estado que toman las Estructuras que no podrán ser usadas por el negocio y que podrán ser recuperadas para su uso en cualquier momento.

Desactivada

Estado que toman las Estructuras que no podrán ser usadas por el negocio, que tuvieron su función en algún momento y ya terminó su vida útil. No pueden ser recuperadas.

Tabla 2. Descripción del modelado de dominio. E_Responsabilidad_Estructura

Descripción	Representa el número de plazas que existen por cada una de las responsabilidades asignadas a las estructuras.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	
cantidad_plazas	Cantidad de plazas que para la responsabilidad tiene la estructura	entero	si	u	5	-

Tabla 3. Descripción del modelado de dominio.E_Responsabilidad

Descripción	Representa las responsabilidades que se le pueden otorgar a cada estructura.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	

					Clases válidas
Responsabilidad	Campo que representa los tipos de responsabilidades:	Texto	no	U	- Director - Secretaria
clasificador_cargo	Campo que representa los tipos de cargos:	Texto	no	u	- Administrativos - dirigentes - operarios - servicios - técnicos

Tabla 4. Descripción del modelado de dominio.E_Inmueble

Descripción		Representa el local físico donde estarán ubicadas cada una de las estructuras.			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
					Clases válidas
REEUP	Representa el código del inmueble	Texto	no	U	
Estado	Campo que representa el estado del inmueble:	Texto	no	U	- en uso - cerrado - entregado
estado_constructivo	Representa si el centro está en malas o buenas condiciones	Texto	no	U	- buena - mala

Municipio	Municipio al que pertenece el centro	Texto	no	U	Matanzas
Provincia	Provincia al que pertenece el centro	Texto	no	U	colon
construcción	Campo que representa el tipo de construcción: - convencional - casa adaptada - madera - sandino - viga Inclinada - otro tipo	Texto	no	m	- convencional - casa adaptada - madera - sandino -viga Inclinada - otro tipo
Patrimonio	Campo que representa el tipo de patrimonio que puede ser el centro: - local - nacional	Texto	si	U	- local - nacional
Electricidad	Representa si el inmueble tiene o no electricidad	Texto	no	U	- si - no
fuente_agua	Representa si el inmueble tiene o no una fuente de agua	Texto	no	U	- si - no
agua_disponible	Representa si el inmueble tiene o no agua disponible	Texto	no	U	- si - no
instalaciones_lava_manos	Representa si el inmueble tiene o no instalaciones de lava mano	Texto	no	U	- si - no
adaptación_est_discapacitado	Representa si el inmueble tiene o no adaptaciones para estudiantes discapacitados	Texto	no	U	- si - no
servicios_higienico	Representa si el inmueble tiene o no servicios higiénicos	Texto	no	U	- si - no

servicios_higienico_c on_agua	Representa si el inmueble tiene o no servicios higiénicos con agua	Texto	no	U	- si - no
servicios_higienico_u n_sexo	Representa si el inmueble tiene o no servicios higiénicos de un solo sexo	Texto	no	U	- si - no
servicios_higienico_u n_sexo	Representa si el inmueble tiene o no servicios higiénicos que son utilizados	Texto	no	U	- si - no
fecha_entrega	Representa la fecha en que es entregado el inmueble	Numérico	no	U	Está compuesta por día, mes, año
entregado_a	Representa a quien es entregado el inmueble	Texto	no	U	Nombre de la institución

Estados de validez

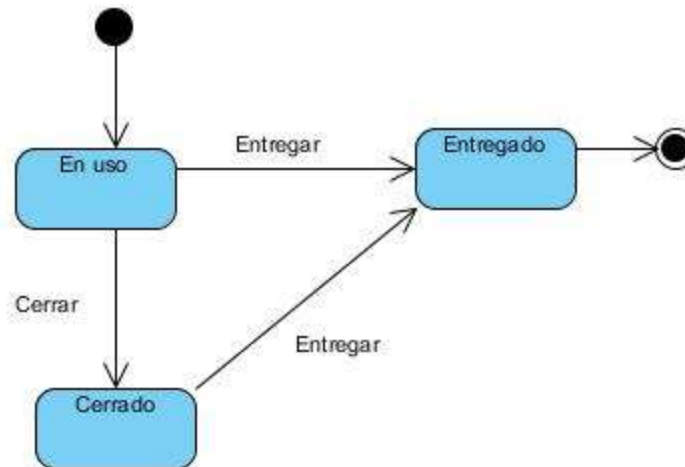


Tabla 5. Descripción del modelado de dominio. Centro Educativo

Descripción	Representa donde estarán ubicadas cada una de las enseñanzas.
Atributos	

Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
					Clases válidas
código_centro	Código único que identifica a los centros en el MINED a nivel nacional. Compuesto por: - 2 dígitos representan la provincial - 2 dígitos el municipio - 4 dígitos para el número del centro	numérico	No	U	29044062
composición	Campo que representa el ministerio al que pertenecen los centros: - mixto - puro	Texto	No	U	- mixto - puro
municipio	Representa el municipio al que pertenece la estructura	texto	No	U	Colón
provincia	Representa la provincia al que pertenece la estructura	texto	No	u	Matanzas
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	u	
Consejo_popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	u	Nombre del consejo popular
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	U	- númeroº numero' numero", luego una N o una S.

				<ul style="list-style-type: none"> - el primer numero debe estar entre 0 y 90. - el segundo y el tercer numero no son obligatorios
		Texto	No	u
Longitud	Representa la longitud a la cual pertenece la estructura			
				<ul style="list-style-type: none"> - númeroº numero' numero", luego una E o una O. - el primer numero debe estar entre 0 y 180. - el segundo y el tercer numero no son obligatorios

plan_turquino	Representa si el centro educacional pertenece o no al plan turquino	Texto	No	u	- si - no
cant_sesión	Campo que representa las sesiones de clases que tiene el centro: - sesión mañana - sesión tarde - doble sesión	Texto	no	u	- sesión mañana - sesión tarde - doble sesión
sector	Campo que representa el tipo de sector: - rural - urbano	Texto	No	U	- rural - urbano
adscripta_oficina	Campo de texto que representa a que oficina está adscripta el centro: - historiador de la ciudad - conservador de la ciudad	Texto	No	U	- historiador de la ciudad - conservador de la ciudad
régimen	Campo que representa el tipo de Funcionamiento que puede ser el centro: - interno - seminterno - externo	Texto	No	m	- interno - seminterno - externo
sede_universitaria	Representa si el centro educacional es sede universitaria o no	Texto	No	u	- si - no
biblioteca	Representa si el centro educacional tiene biblioteca o no	Texto	No	u	- si - no

materiales_est _discapacitado s	Representa si el centro educacional tiene materiales para estudiantes con discapacidad o no	Texto	No	u	- si - no
población	Campo que representa la cantidad estimada de personas que viven en la región donde se encuentra la escuela	Entero	si	u	1950
consejo_salud	Campo que representa el área de salud responsable de brindar estos servicios a la centro	texto(50 caractere s)	Si	u	Este campo puede tener letras(de 0 a 255 caracteres) y números.
correo	Campo que representa el correo del centro	texto(50 caractere s)	Si	u	Usuario@
teléfono	Campo que representa el teléfono del centro	numérico	Si	u	
relación_pais	Representa la relación que existe entre el centro y un país	Texto	Si	u	Campo de selección
actividad_princ ipal	Representa la actividad principal que realiza el centro	texto	No	u	

Tabla 6. Descripción del modelado de dominio. Características Inmueble

Descripción	Representa el número de características existentes para cada inmueble.				
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones

						Clases válidas
cantidad	Cantidad de una características en un determinado inmueble	Numérico	Si	U		10

Tabla 7. Descripción del modelado de dominio. Características_Centro_Educacional

Descripción	Representa si los centros educativos tienen o no determinadas características.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
Clases válidas						
existencia	Representa si existe o no una determinada característica	Texto	Si	U	- Si	- no

Tabla 8. Descripción del modelado de dominio. Características

1 Descripción	Representa los valores que tienen cada una de las entidades.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Clases válidas	
características	Características que puede tener un inmueble: - Capacidad de internos - Capacidad de seminternos	texto	no	U	- Capacidad de internos	- Capacidad de seminternos - Capacidad de externos

- Capacidad de externos
- Cantidad de aulas
- Electricidad
- Fuente de agua
- Agua disponible
- Instalaciones de lava manos
- Adaptación de estudiantes discapacitado
- Servicios higiénicos
- Servicios higiénicos con agua
- Servicios higiénicos de un sexo
- Servicios higiénicos un sexo

- Cantidad de aulas
- Electricidad
- Fuente de agua
- Agua disponible
- Instalaciones de lava manos
- Adaptación de estudiantes discapacitado
- Servicios higiénicos
- Servicios higiénicos con agua
- Servicios higiénicos de un sexo
- Servicios higiénicos un sexo

Tipo_características Representa los tipos de características que existen:

- Por existencia
- Por cantidad

Tabla 9. Descripción del modelado de dominio.Dirección_Nacional

Descripción		Representa todas las características que tienen las direcciones nacionales.			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
Clases válidas					
Código	Representa el código de la estructura	numérico	No	u	
municipio	Representa el municipio al que pertenece la estructura	texto	No	u	Matanzas
provincia	Representa la provincia al que pertenece la estructura	texto	No	u	Colón
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	u	
Consejo_popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	u	
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	u	
Longitud	Representa la longitud a la cual pertenece la estructura	Texto	No	u	

Tabla 10. Descripción del modelado de dominio. Dirección_Provincial

Descripción		Representa todas las características que tienen las direcciones provinciales.			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
Clases válidas					
Código	Representa el código de la estructura	numérico	No	U	
municipio	Representa el municipio al que pertenece la estructura	texto	No	U	Matanzas
Provincia	Representa la provincia al que pertenece la estructura	texto	No	U	Colón
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	U	
Consejo_popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	U	
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	no	U	
Longitud	Representa la longitud a la cual pertenece la estructura	Texto	no	U	

Tabla 11. Descripción del modelado de dominio. Dirección_Municipal

3 Descripción		Representa todas las características que tienen las direcciones municipales.			
Atributos					

Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Código	Representa el código de la estructura	numérico	no	U		
municipio	Representa el municipio al que pertenece la estructura	texto	No	U	Matanzas	
Provincia	Representa la provincia al que pertenece la estructura	texto	No	U	Colón	
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	U		
consejo popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	U		
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	U		
Longitud	Representa la longitud a la cual pertenece la estructura	Texto	No	U		

Tabla 12. Descripción del modelado de dominio. Hogar

3 Descripción		Representa todas las características que tienen los hogares.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
código	Representa el código de la estructura	Numérico	No	U		
municipio	Representa el municipio al que pertenece la estructura	Texto	No	U	Matanzas	
provincia	Representa la provincia al que pertenece la estructura	Texto	No	u	Colón	
Dirección	Representa la dirección donde está ubicada la estructura	Texto	Si	u		
consejo popular	Representa el consejo popular al cual pertenece la estructura	Texto	Si	u		
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	u		
Longitud	Representa la longitud a la cual pertenece la estructura	Texto	No	u		

Tabla 13. Descripción del modelado de dominio. Palacio_Pioneros

3 Descripción		Representa todas las características que tienen los palacios de pioneros.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
código	Representa el código de la estructura	numérico	No	U		
municipio	Representa el municipio al que pertenece la estructura	texto	No	U	Matanzas	
provincia	Representa la provincia al que pertenece la estructura	texto	No	U	Colón	
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	U		
consejo popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	U		
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	U		
Longitud	Representa la longitud a la cual	Texto	No	U		

pertenece la
estructura

Tabla 14. Descripción del modelado de dominio. Campamento_exploradores

3 Descripción		Representa todas las características que tienen los campamentos de exploradores.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Código	Representa el código de la estructura	numérico	No	U		
municipio	Representa el municipio al que pertenece la estructura	texto	No	U	Matanzas	
Provincia	Representa la provincia al que pertenece la estructura	texto	No	U	Colón	
Dirección	Representa la dirección donde está ubicada la estructura	texto	Si	U		
consejo popular	Representa el consejo popular al cual pertenece la estructura	texto	Si	U		
Latitud	Representa la latitud a la cual pertenece la estructura	Texto	No	U		

Longitud	Representa la longitud a la cual pertenece la estructura	Texto	No	U
----------	--	-------	----	---

Tabla 15. Descripción del modelado de dominio. Secretaria_Docente

3	Descripción	Representa todas las características que tienen las secretarias docentes.			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
					Clases válidas Clases no válidas
Código	Representa el código de la estructura	numérico	No	U	

Tabla 16. Descripción del modelado de dominio. Dirección

3	Descripción	Representa todas las características que tienen las direcciones.			
Atributos					
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones
					Clases válidas Clases no válidas
código	Representa el código de la estructura	numérico	No	U	

Tabla 17. Descripción del modelado de dominio. Grupo_Administrativo

3	Descripción	Representa todas las características que tienen los grupos administrativos.			
Atributos					

Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Código	Representa el código de la estructura	numérico	No	U		

Tabla 18. Descripción del modelado de dominio. Organismos_Formadores

4	Descripción	Representa los tipos de organismos formadores existentes en el país.				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
tipo_organismos_formadores	Representa los tipos de Ministerios existentes: - MINED - MINSAP - INDER - MININT - MINAGRI - MINFAR	texto	No	U	Campo de texto	

Tabla 19. Descripción del modelado de dominio. Tipo_Centro

5	Descripción	Representa los tipos de centros existentes en el país.				
Atributos						

Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?
tipo_centro	Representa los tipos de centro: Universidad ESBU	texto	No	U

Tabla 20. Descripción del modelado de dominio. Especialidad

6	Descripción	Representa en que se especializa cada uno de los centros educacionales		
Atributos				
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?
tipo_especialidad	Representa el tipo de especialidad a la que se dedica el centro:	texto	No	U

Tabla 21. Descripción del modelado de dominio. Enseñanza

7	Descripción	Representa los tipos de enseñanza a los que se dedica cada centro		
Atributos				
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?
tipo_enseñanza	Representa el tipo de enseñanza que se realizan en el centro:	texto	No	U

Anexo 2 Descripción de requisitos por proceso.

Tabla 22. Descripción de los requisitos por proceso: crear estructura

Precondiciones		<ul style="list-style-type: none"> • El usuario debe estar autenticado con los parámetros necesarios • Debe existir un inmueble activo para la creación
Flujo de eventos		
Flujo básico <<Crear Estructura>>		
1	El requisito empieza en la página principal cuando el usuario selecciona el módulo de estructura y composición y dentro de esta opción crear estructura.	
2	El sistema mostrara los siguientes campos: tipo de estructura, nombre, hijo de, siglas, categoría de estructura, nivel, fecha de activación y si está activo o no.	
3	El usuario debe rellenar todos los campos, en caso de que los campos sean dejados en blanco ver flujo alternativo 3.a.	
4	El usuario pulsa el botón guardar y crea la estructura.	
5	El sistema vuelve al paso 3.	
Pos-condiciones		
1.	Queda creada la estructura con los campos seleccionados e introducidos.	
Flujos alternativos		
Flujo alternativo 3.a. Campos de filtro de búsqueda vacíos.		
1	El sistema mostrara un mensaje de advertencia "No pueden existir campos en blanco".	
2	El usuario selecciona la opción aceptar y se marcan en rojo los campos que fueron dejados en blanco.	
3	El sistema regresa al paso 2 del flujo básico.	
Pos-condiciones		
	<ul style="list-style-type: none"> • N/A 	

Validaciones			
#	Campo	Posibles valores	Valores no validos
1	Tipo de estructura	Dirección Nacional, Dirección provincia, Dirección municipal, Campamento, Centro Educacional, Hogar, Palacio, Dirección, Secretaria docente	-Seleccione-
2	Nombre	Dirección municipal de Cienfuegos, Dirección provincial de matanzas	Números y caracteres extraños
3	Hijo de	Dirección Nacional	-Seleccione-
4	Siglas	DMM	Números y caracteres extraños
5	Categoría de estructura	Campamento, Centro Educacional, Hogar, Palacio, Dirección, Secretaria docente	-Seleccione-
6	Nivel	Nacional, Provincial, Municipal	-Seleccione-
7	Fecha de activación	5-2-20	Letras y caracteres extraños
8	Activo	- Si - No	(-)
Conceptos	Estructura	Visibles en la interfaz: Tipo de estructura: String	
Requisitos especiales	RnF1, RnF3, RnF4, RnF6, RnF11		
Asuntos pendientes	N/A		
Prototipo elemental de interfaz gráfica de usuario			

Crear Estructura

Tipo de estructura:

-Selecciona-

Nombre:

Hijo de:

-Selecciona-

Siglas:

Categoría de estructura:

-Selecciona-

Nivel:

-Selecciona-

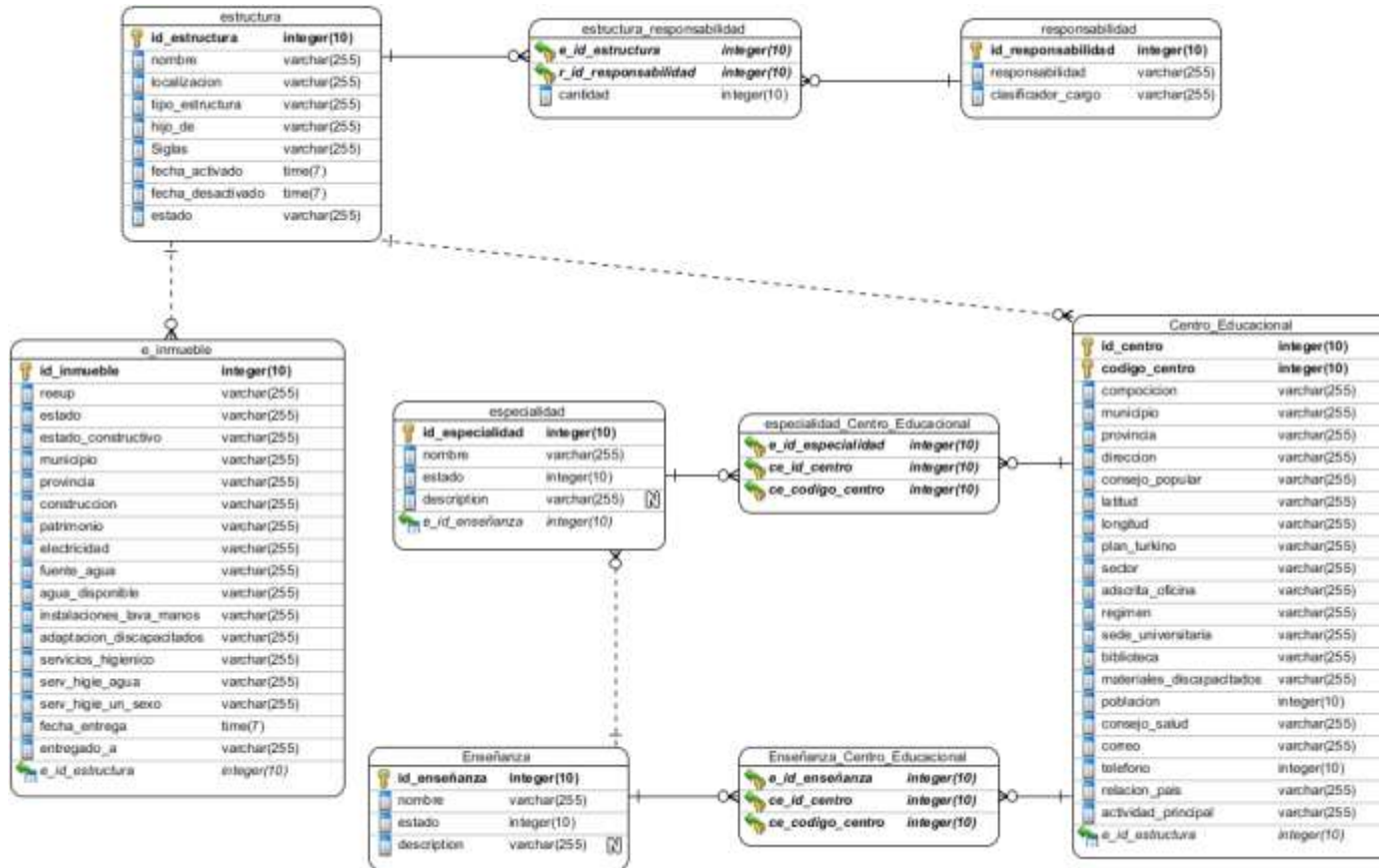
Fecha de activación:

Activo

Aceptar

Cancelar

Anexo 3 Modelo de Datos



Anexo 4 Estrategia de Validación

Tabla 23. Cuadro lógico de IADOV

	Pregunta cerrada
	1

Pregunta cerrada 3	No			No sé			Sí		
	Pregunta cerrada 2								
	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	6
No sé qué decir	2	3	6	3	3	3	6	3	4

Tabla 24. Índice de satisfacción de ladov

Índice de satisfacción	Escala
Máxima satisfacción	+1
Más satisfecho que insatisfecho	0,5
No definido y contradictorio	0
Más insatisfecho que satisfecho	-0,5
Máxima insatisfacción	-1

Anexo 5 Diagrama de Componentes

