



Universidad de las Ciencias Informáticas

Facultad 1

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

Sistema de control y seguimiento de la Planeación
Estratégica de la Facultad 1 de la Universidad de las
Ciencias Informáticas

Autor: Armando Rafael Azcue Fonseca.

Tutor(es): MSc. Yanet Espinal Martín

Ing. Javier Anias Santos

La Habana, 2020

“Año 62 de la Revolución”

Declaración de autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Armando R. Azcue Fonseca

Autor

MSc. Yanet Espinal Martín

Tutor

Ing. Javier Anias Santos

Tutor

Datos de contacto

MSc. Yanet Espinal Martín.

Licenciada en Ciencias de la Computación de la Universidad de Oriente en el año 2005. Profesor asistente del departamento de programación de la facultad 1. Universidad de las Ciencias Informáticas. En el año 2010 defendió su tesis de maestría en Informática Aplicada. Actualmente se desempeña como Jefa de Dpto. del Dpto. de Informática.

Correo electrónico: "Yanet Espinal Martín" <yanete@uci.cu>;

Ing. Javier Anias Santos.

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) en el año 2013. Profesor Instructor prestación de servicios en el Departamento de Informática de la Facultad 1. Universidad de las Ciencias Informáticas. Actualmente se desempeña como director de la Residencia.

Correo electrónico: javier@uci.cu

Dedicatoria

*A mis padres, a mi familia y amigos por apoyarme incondicionalmente
en todos estos años...*

Agradecimientos

A toda mi familia por apoyarme siempre y sobre todo en mi etapa en la UCI. A mis tutores por apoyarme durante el desarrollo del trabajo de diploma. A todos mis compañeros de aula y de apartamento en la UCI que de una forma u otra colaboraron con este trabajo, gracias por los momentos de alegría y de apoyo. A mis profesores y compañeros de los cursos de la Facultad1. A la UCI ...

Resumen

Las organizaciones deben dirigir sus estrategias a fin de alcanzar niveles de competencia aceptables, debe existir una coordinación efectiva entre las funciones de planeación, diseño, distribución y entrega. El desarrollo de un óptimo proceso de planeación estratégica asegura la competitividad futura y presente de la organización, analizando la situación interna y anticipando la evolución del entorno.

En la facultad 1 de la Universidad de las Ciencias Informáticas existen limitantes para la correcta gestión de la Planeación Estratégica, al ser un proceso manual se dificulta la sincronización, seguimiento, control y evaluación de los indicadores y tareas a cumplir por parte de los profesores y trabajadores. La presente investigación se plantea mejorar el control y seguimiento de los objetivos estratégicos de la facultad 1. Para ello propone el desarrollo de un sistema informático que gestione la información que involucra dichos procesos.

Para la propuesta de solución se empleó Visual Paradigm como herramienta de modelado y apoyo a las actividades ingenieriles. El lenguaje de desarrollo fue Python, como framework fue empleado Django, para la persistencia de los datos SQLite para las pruebas y el motor de Postgres para su despliegue, para el desarrollo de las interfaces se utilizó CCS, JS y HTML 5. Como resultado se obtiene un Sistema para el seguimiento y gestión de la Planificación Estratégica de la facultad 1, extensible y modular, con la capacidad de seguir creciendo e incorporarle nuevas funcionalidades.

Palabras clave: control y seguimiento, Facultad 1, indicadores, Objetivos Estratégicos, planeación.

Tabla de contenido

Introducción	1
Capítulo 1: Fundamentación Teórica de la investigación	6
1.1 Conceptos básicos relacionados con el dominio del problema.	6
1.2 La Planeación Estratégica gubernamental en Cuba.	7
1.3 Planeación Estratégica en la educación superior cubana.	8
1.4 Planeación Estratégica en la Universidad de las Ciencias Informáticas.	12
1.5 Análisis del Estado del Arte	13
1.5.1 Análisis de las herramientas identificadas	16
1.6 Metodologías de desarrollo de software	17
1.6.1 Metodología AUP-UCI.	17
1.7 Ambiente de desarrollo.	18
1.8 Conclusiones parciales.	21
Capítulo 2: Características del Sistema	22
2.1 Modelo conceptual	22
2.2 Requisitos de la propuesta de solución	22
2.2.1 Requisitos funcionales de la propuesta de solución	23
2.2.2 Requisitos no funcionales de la propuesta de solución	27
2.3 Descripción de la arquitectura	28
2.3.1 Patrón arquitectónico	29
2.3.2 Patrones de diseño	30
2.4 Diagrama de Clases del Diseño	35
2.5 Diagrama de paquetes	36
2.6 Conclusiones parciales	37

Capítulo 3: Implementación y pruebas.	38
3.1 Modelo de datos	38
3.2 Estándares de codificación.	38
3.3 Normas de Seguridad informática a tener en cuenta	41
3.4 Tratamiento de errores	42
3.5 Modelo de despliegue	42
3.6 Pruebas internas	43
3.7 Conclusiones parciales	50
Conclusiones	51
Recomendaciones	52
Referencias Bibliográficas	53
Anexos	56

Introducción

Los tópicos de la economía moderna corresponden a las áreas del conocimiento, la digitalización, la virtualización, integración, innovación, inmediatez, el capital intelectual, manejo de información en forma oportuna (en cantidad, calidad y tiempo) y la consideración efectiva del cliente como una parte esencial del proceso de negocio. El ritmo acelerado de cambio, exige a las organizaciones verificar la condición de adaptabilidad a las nuevas situaciones, así como las de agilidad y rapidez de respuesta, una administración efectiva de sus recursos humanos y de información, que les permita ser más competitivos y efectivos (Novoa & Iglesias, 2016).

Ante estas características las organizaciones deben dirigir sus estrategias a fin de alcanzar niveles de competencia, debe existir una coordinación efectiva entre las funciones de planeación, diseño, distribución y entrega. El desarrollo de un óptimo proceso de planeación estratégica asegura la competitividad futura y presente de la organización, analizando la situación interna y anticipando la evolución del entorno (Ortiz, 2018).

Todas las organizaciones buscan ser exitosas. Por este motivo, se hace necesario trazar un camino claro que permita alcanzar las metas previstas. Sin embargo, trazar este camino no significa sólo establecer el punto al que se quiere llegar, implica también plantear cuáles serán los pasos para alcanzar esa meta. Es precisamente esto en lo que consiste la Planeación Estratégica de una organización (Vargas et al., 2016; Araya, 2017).

El éxito de una correcta Planeación Estratégica comienza con una correcta definición de los objetivos estratégicos y sus criterios de medida. Estos deben estar distribuidos entre todos los trabajadores de manera que todos tributen a su cumplimiento. La Planeación Estratégica concreta las ideas en planes y programas de actuación, definidos en el tiempo y en el espacio, formulados en términos objetivos, medibles y verificables, cuantificados en términos de costo, inversión y resultados, lo que la convierte en generadora de ventajas competitivas para la organización. A este proceso pueden contribuir notablemente las tecnologías de la información (Hernández et al., 2018).

Durante mucho tiempo, la función de la informática dentro de las empresas fue considerada

como una herramienta para soportar funciones operativas. Actualmente los sistemas de información son vistos además como áreas de oportunidad para lograr ventajas en el terreno de la organización y automatización de los procesos. Son considerados como una herramienta para mejorar la estructura competitiva de las instituciones. Los sistemas de información determinan la demanda de aplicaciones de la organización (Araya, 2017).

Los sistemas de información estratégicos pueden ser considerados como el uso de las Tecnologías de la Información para soportar o dar forma a la estrategia competitiva de la organización. Los sistemas de información estratégicos ayudan a obtener o mantener una ventaja competitiva, establecen un enlace entre la organización y su medio ambiente a fin de lograr la información que se requiere para la formulación, implementación y seguimiento de las estrategias. Además, proveen de un mecanismo para la planeación y toma de decisiones estratégicas (Acosta, Becerra & Jaramillo, 2017).

La globalización, la competitividad y el desarrollo tecnológico, han generado que el sector educacional se transforme para adaptarse y responder a los cambios constantes y de incertidumbre que plantea el entorno. La mayoría de las instituciones de educación usan la planeación estratégica como una herramienta que soporta la gestión. En este sentido, constantemente aparecen organizaciones educativas que muestran transformaciones significativas, dinámicas más constructivas y efectivas en sus funciones sustantivas, administrativas y financieras, apoyadas en procesos exitosos de planeación estratégica, que se caracterizan por lograr estándares que estructuran equipos humanos comprometidos, articulados con la misión, la visión y los objetivos. En tal sentido, las universidades cubanas, de economía presupuestada buscan constantemente mejorar y gestionar eficientemente sus recursos, a partir de una correcta planeación estratégica (Jones & Alderete, 2016).

En Cuba, se ha venido trabajando en la implantación de la Dirección por Objetivos y la introducción paulatina de la Planificación Estratégica a todos los niveles, involucrando en este esfuerzo a la empresa y a las entidades de gobierno en todas las instancias en un proceso lógico, sencillo y caracterizado por una amplia participación de los trabajadores en cada momento.

Una prioridad del trabajo en el Ministerio de Educación Superior (MES) para el periodo que abarca la planificación estratégica 2017-2021, es la labor político-ideológica con toda la comunidad universitaria, en el propósito de mantener vivo el legado del Comandante en Jefe Fidel Castro Ruz y ser fieles al compromiso contraído de cumplir con su concepto de Revolución. Como parte de ello es el llamado a planificar con conciencia los recursos a disposición del sector. El Lineamiento No. 268 de la Política Económica y Social del Partido y la Revolución para el periodo 2016-2021 plantea: Consolidar y perfeccionar el sistema de planificación de objetivos y actividades del gobierno, idea que ha inspirado todo el proceso de elaboración de la planificación estratégica del sistema MES para el periodo 2017-2021. Este lineamiento se aplica en la Universidad de Ciencias Informáticas de la Habana en función de planificar estratégicamente los procesos que se ejecutan (Zaldívar, Campaña & Pérez, 2017; Ministerio de Educación Superior, 2017).

La Universidad de Ciencias Informáticas (UCI) tiene como misión formar profesionales comprometidos con su patria y altamente calificados en la rama de la informática, producir aplicaciones y servicios informáticos a partir del vínculo estudio - trabajo como modelo de formación - investigación - producción, sirviendo de soporte a la industria cubana de la informática. El modelo de formación de la UCI es único de su tipo en Cuba, con alta complejidad entre sus procesos sustantivos, lo que aumenta la necesidad del estudio de mejores formas de Planeación Estratégica

Se hace necesario trazar un camino claro que permita alcanzar las metas previstas en los planes e indicadores de la universidad. Sin embargo, trazar este camino no significa sólo establecer el punto al que se quiere llegar, implica también plantear cuáles serán los pasos para alcanzar esa meta. Y en esta, analizar de forma crítica las lecciones aprendidas de ejecuciones previas para mejorar y aumentar las capacidades de planificación de la institución (Pacheco et al., 2018).

En la UCI se planifican objetivos estratégicos con indicadores que permiten medir su cumplimiento. Por ser una entidad con disímiles niveles de dirección y una cantidad considerable de trabajadores (más de dos mil), se hace muy engorroso el control y seguimiento de los objetivos estratégicos propuestos. Cada trabajador tiene un plan de resultado anual y uno mensual en los cuales debe estar reflejado tareas que tributen al

cumplimiento de los objetivos estratégicos de sus áreas de dirección y estas, a su vez, deben tributar al cumplimiento de los objetivos estratégicos de la universidad.

Controlar todo este grafo de interconexiones y tener una visión general del cumplimiento de los objetivos estratégicos en cualquier área de dirección de manera rápida y precisa es una necesidad para los directivos de la UCI. Hacerlo de manera manual es una tarea compleja, que conlleva gastos en recursos y tiempo para controlar la ejecución de las actividades planificadas.

Cada facultad, en función de las áreas de resultados claves y los objetivos definidos para el cumplimiento de los objetivos estratégicos de la universidad, define pautas, indicadores y criterios de medidas para cumplir su planeación estratégica del año. Estos objetivos se definen con indicadores claros y medibles y deben de estar reflejados en cada uno de los planes de resultados individual de cada profesional de las diferentes áreas de las facultades.

En la facultad 1 este proceso se realiza en jornadas intensas de trabajo o que implica consumo de fondo de tiempo considerable por parte de la dirección de la facultad, además su control y seguimiento se hace muy difícil a los directivos, así como tener una visión general y rápida del nivel de cumplimiento de los objetivos estratégicos de su área de dirección, razón por la cual no pueden tomar decisiones en el momento preciso que le permitan llegar al final del período de planeación estratégica con los objetivos cumplidos. Teniendo en cuenta el éxodo de personal calificado se hace más engorroso y difícil la ejecución de actividades de control y gestión de la planeación estratégica. El documento rector de la planificación estratégica 2017-2021 cuenta entre sus elementos esenciales con la creación de “Directrices de Cambio y proceso de transformaciones de la educación superior”, esto en consonancia con el actual proceso de informatización de la sociedad cubana, impulsado por la dirección del gobierno, propicia a la Facultad 1 a buscar alternativas para lograr una mejor gestión de la Planeación Estratégica.

Como consecuencia de la situación planteada anteriormente se define como **problema a resolver**: ¿Cómo mejorar el control y seguimiento de los objetivos estratégicos de la facultad 1 de la Universidad de las Ciencias Informáticas? El **Objeto de estudio** sobre el

cual se centra el problema es: el proceso de gestión de información de la Planeación Estratégica en la educación superior cubana, enmarcado en el **Campo de acción:** Los sistemas informáticos para la gestión de la planeación estratégica en las universidades cubanas.

Para dar solución al problema identificado se determina como **objetivo general:** desarrollar un sistema de gestión de información, para elevar el control y seguimiento de la Planeación Estratégica de la facultad 1 de la Universidad de las Ciencias Informáticas.

Para dicho resultado se han definido las siguientes tareas de investigación:

- Fundamentación de las bases teóricas de la investigación a partir de los principales conceptos y definiciones asociados el proceso de gestión de información de la Planeación Estratégica en la educación superior cubana.
- Análisis y diseño de un sistema informático de control y seguimiento de la Planeación Estratégica.
- Desarrollo de un sistema informático de control y seguimiento de la Planeación Estratégica.
- Validación a partir de las pruebas de software diseñadas para medir la calidad de la solución.

Para responder al desarrollo de la investigación se emplean los métodos científicos:

Métodos teóricos:

Histórico-Lógico: se utiliza para el análisis de los sistemas de gestión para la planeación estratégica de empresas. Propicia describir el objetivo de dichos sistemas y comprobar si se ajustan al objetivo de la investigación.

Analítico-Sintético: se utiliza para el proceso de análisis y estudio de la bibliografía empleada, arribando así a conclusiones que contribuyan a elaborar la propuesta de solución, y para el estudio y análisis de las herramientas a utilizar.

Métodos empíricos:

Entrevista: método para fundamentar y precisar el problema a resolver. Se utiliza para

establecer las necesidades del cliente, identificar los procesos de planeación y evaluación de la facultad y los profesores de la misma en la UCI y las deficiencias que presentan.

Observación: Posibilita la verificación de la información obtenida a través de las entrevistas realizadas. Este método se emplea para comprobar la manera en que se realiza el proceso de planeación estratégica y la evaluación de esta, así como su funcionamiento en la actualidad.

Modelación: este método permite efectuar el análisis de la realidad existente mediante diagramas que ayudan a comprender y unir los datos referentes a la reservación de transporte, también es utilizado en la elaboración de los diagramas de procesos donde se explican las actividades que se realizan en la reservación de transporte de la Universidad de las Ciencias Informáticas.

Estructura Capitular del documento

Capítulo 1: Fundamentación teórica de la investigación: dedicado a los fundamentos teóricos de la investigación. Se caracteriza el control y seguimiento de la Planeación Estratégica en el Ministerio de Educación Superior y en la facultad 1 de la Universidad de las Ciencias Informáticas. Se analiza el estado actual de los sistemas y herramientas de software para el Control y seguimiento de la Planeación Estratégica. Además, se definen los lenguajes, tecnologías herramientas y metodología de desarrollo de software.

Capítulo 2: Propuesta de Solución: describe metodológicamente las fases de desarrollo de software del sistema de control y seguimiento de la Planeación Estratégica en el Ministerio de Educación Superior y en la facultad 1 de la Universidad de las Ciencias Informáticas. Se especifican los requisitos funcionales y no funcionales, así como los artefactos generados. Se presentan los estándares de codificación y los patrones de diseño empleados

Capítulo 3: Validación de resultados: contiene la evaluación de los resultados alcanzados en la investigación, a través de pruebas de software determinadas por los diseños de casos de prueba.

Capítulo 1: Fundamentación Teórica de la investigación

En el presente capítulo se plantean los conceptos fundamentales relacionados con el objeto de estudio del trabajo, que servirán de guía para desarrollar la solución propuesta. Se realiza un análisis de las herramientas identificadas en la literatura consultada, relacionadas con el campo de acción y el objeto de estudio. Por último, se realiza una caracterización de las tecnologías, herramientas y lenguajes para definir y seleccionar las que serán utilizadas en el desarrollo de este trabajo de diploma.

1.1 Conceptos básicos relacionados con el dominio del problema.

En esta sección se presentan un conjunto de definiciones y conceptos para esclarecer la terminología a utilizar en la base teórica del presente trabajo.

Objetivos estratégicos

Los objetivos estratégicos son los fines o metas desarrollados a nivel estratégico que una organización pretende alcanzar a largo plazo. Están basados en la visión, la misión y los valores de una organización y condicionan las acciones que se llevarán a cabo (Fernández et al., 2016).

Son los resultados que una organización aspira a lograr en el futuro. Todo punto concreto en el tiempo y en el espacio. Son declaraciones de resultados medibles que se deben lograr dentro del marco de tiempo del Plan Estratégico (Santos, 2019).

Características: Medible, razonable, claro, coherente y motivador.

Elementos: Variable, atributo, unidad de medida, umbral (punto final) y horizonte.

Planeación

Según la Real Academia de la Lengua Española (RAE) Planeación es un plan general, metódicamente organizado y frecuentemente de gran amplitud, para obtener un objetivo determinado, tal como el desarrollo armónico de una ciudad, el desarrollo económico, la investigación científica o el funcionamiento de una industria.

La planeación es la acción de la elaboración de estrategias que permiten alcanzar una meta ya establecida, para que esto se pueda llevar a cabo se requieren de varios elementos, primero se debe comprender y analizar una cosa o situación en específica, para luego pasar a la definir los objetivos que se quieren alcanzar, de cierta forma, el planear algo define el lugar o momento en donde se encuentra algo o alguien, plantea a donde se quiere ir e indica paso a paso lo que se debe hacer para llegar hasta allí (Ramírez, 2017). Para la presente investigación se asume esta definición por su pertinencia con el trabajo.

Planeación estratégica

En la planeación estratégica se analiza la situación actual, se establecen los objetivos generales de la empresa, y se diseñan estrategias, cursos de acción y planes estratégicos necesarios para alcanzar dichos objetivos; planes que afectan una gran variedad de actividades y que parecen simples y genéricos. La planeación estratégica se realiza a nivel de la organización, es decir, considera un enfoque global de la empresa, por lo que debe ser elaborada por la cúpula de la empresa, y ser realizada a largo plazo, en teoría, para un periodo de 5 años a más, aunque en la práctica se suele realizar para un periodo de 3 a 5 años, debido a los cambios del mercado (Vargas et al., 2016; Chiavenato, 2017).

Control de las estrategias

Para controlar, primero hay que definir que es: gestión, y después ver de qué forma se controla esa gestión. Gestión es medida y análisis, estas constituyen la base de la acción y esta se hace muy incierta si no se puede visualizar y comprender. Gestión comprende el proceso de técnicas, conocimientos y recursos, para llevar a cabo la solución de tareas eficientemente. La Gestión Empresarial, es un término utilizado para describir el conjunto de técnicas y la experiencia de la organización en procesos como planificación, dirección y control eficiente de las operaciones y de las otras actividades de la empresa (Valdés et al., 2017).

Mientras que el control es una actividad que forma parte de la vida cotidiana del ser humano, conscientemente o no. Es una función que se realiza mediante parámetros establecidos con anterioridad, y el sistema de control es el fruto de la planificación y, por tanto, apunta al

futuro. El control se refiere a la utilización de registros e informes para comparar lo logrado con lo programado (Parra y Dávila, 2016).

1.2 La Planeación Estratégica gubernamental en Cuba.

La aseveración de que los Estados deben alcanzar un elevado nivel de desarrollo social y un crecimiento económico sostenible y equitativo, unido al logro de una utilización eficiente y eficaz de los recursos disponibles hace que las técnicas, herramientas y métodos de dirección se mantengan en un proceso constante de perfeccionamiento (Ministerio de Educación Superior, 2017).

A inicio de la década de los años 80, los presupuestos por programas marcaron hitos importantes en la evolución de las técnicas presupuestarias, en respuesta a la necesidad de mejorar los procesos de asignación de recursos. En 1989, aparecen experiencias orientada al seguimiento de objetivos que integraban los cuatro procesos clásicos del ciclo financiero: planificación, elaboración de presupuestos, gestión y control.

Hasta ese momento se reconocen resultados que han trascendido a los procesos contemporáneos de presupuesto y gestión, entre ellos:

- Los objetivos constituyen el basamento fundamental sobre el que se construyen los programas sociales y sirven de base para construir los sistemas control de los resultados.
- Posibilitan un enlace entre los elementos considerados en los presupuestos y los sistemas de gestión de los estados.
- Integra los resultados del proceso presupuestario.
- Permite una evolución positiva en el proceso de negociación entre los entes presupuestarios y los administradores de los programas.

Sin embargo, a pesar de estos resultados, se reconoce que está siendo una tarea compleja, que requiere tiempo y preparación de los funcionarios públicos.

En Cuba, se ha venido trabajando en la implantación de la dirección por objetivos y la introducción paulatina de la Planificación Estratégica a todos los niveles, involucrando en

este esfuerzo a la empresa y a las entidades de gobierno en todas las instancias en un proceso lógico, sencillo y caracterizado por una amplia participación de los trabajadores en cada momento.

Muchos han sido los debates de los científicos, profesores, profesionales y trabajadores sobre cada uno de los elementos de este proceso, sin embargo, esa reflexión colectiva hace que el enfoque cubano sobre estas técnicas sea una contribución de criterios, definiciones, sugerencias y experiencias en la conformación de la Escuela Cubana de Dirección.

En el proceso de modernización de la Administración Financiera del Estado en Cuba, estas herramientas jugaron un papel significativo, al contribuir en el diseño de los procesos de gestión de las finanzas públicas con un enfoque más integral y asimilar experiencias internacionales en este sentido (Fernández, Juvinao and Solano, 2016).

1.3 Planeación Estratégica en la educación superior cubana.

Actualmente, la confección del plan de la economía se hace mucho más compleja dado el grado de incertidumbre con que deben ser elaboradas las proyecciones, la diversificación de la estructura de la propiedad, la operatoria descentralizada de los recursos, la presencia de la dualidad monetaria, las restricciones externas dadas por la complejidad del entorno mundial y el bloqueo económico que se nos ha impuesto, así como la ausencia de proyecciones a mediano plazo esta etapa (Ministerio de Educación Superior, 2017).

Ya no se parte de enmarcamientos sino de potenciales, restricciones y alternativas de cada esfera, para después construir variantes compatibles; se somete a análisis la viabilidad financiera de las proyecciones ramales, en función de la nueva forma de asignar los recursos, se emplean análisis de sensibilidad; los cambios organizativos e institucionales pasan a formar parte de la propia proyección; se concibe la elaboración de variantes en cuanto a logros posibles, decisiones y factores externos; se elaboran estrategias respecto a los factores externos y los agentes económicos.

Para cualquier entidad en las condiciones actuales es vital introducir, definir y controlar sus metas. Lograrán un paso de avance si para ello estructuran un modelo de gestión basado

en la dirección por objetivos. Los objetivos constituyen una vía para expresar las metas o propósitos que pretende alcanzar una organización.

Para las entidades que dependen del Estado en el caso de estudio, cada día se reafirma la posibilidad de utilizar estas herramientas. Primero, hasta la década de los años 80, dada las características de la economía y su planificación que posibilitaban proyectar y evaluar el resultado productivo a corto y largo plazo con relativa estabilidad.

Hasta este momento, se formulaban metas y se utilizaban sistemas de control acorde al desarrollo de los procesos de dirección. En todo este proceso, el Presupuesto del Estado, como instrumento de dirección, reflejaba año tras año, los objetivos de desarrollo económico - social del país para cada período.

La educación superior cubana es un bien público y su realización máxima se obtiene cuando sus procesos son capaces de impactar en el desarrollo socioeconómico del entorno donde se desarrolla. El proceso de actualización del modelo económico cubano, así como el programa de desarrollo económico y social hasta el año 2030 requiere de instituciones de educación superior integradas, innovadoras y pertinentes (Ministerio de Educación Superior, 2017).

Por otra parte, los objetivos de trabajo del Ministerio de Educación Superior para el año 2019 forman parte de la Planificación Estratégica del organismo para el periodo 2017-2021. En ellos se han contemplado los criterios de medida plasmados en los grados de consecución definidos para este año. Los que aquí se presentan han sido valorados a partir de los resultados esperados y alcanzados en el año 2018, así como los eventos y acontecimientos que tienen lugar en las actuales circunstancias nacionales e internacionales, que ejercen influencia sobre la educación superior cubana. La figura 1 muestra un esquema metodológico general de la Planeación estratégica del MES (Ministerio de Educación Superior, 2018).



Figura. 1 Equema metodológico general de la Planeación estratégica del MES (Ministerio de Educación Superior, 2017).

Entre esos elementos ocupa un lugar central el proceso de perfeccionamiento que se viene desarrollando en el actual periodo en el Ministerio de Educación Superior y en todas sus entidades. Se han tenido en cuenta, asimismo, las importantes transformaciones que hoy se están introduciendo en la educación superior cubana y su impacto en la sociedad. Factores esenciales de todo el proceso de definición de los objetivos y los criterios de medida son las importantes tareas que han ido asignando el Partido y el Gobierno a la educación superior y en particular a nuestro Ministerio, en la implementación de los Lineamientos de la Política Económica y Social del Partido y la Revolución, en la conceptualización del Modelo Económico y Social, así como en las Bases para el Plan Nacional de Desarrollo Social hasta 2030 (Ministerio de Educación Superior, 2018).

Como parte de perfeccionamiento del modelo económico cubano se introduce mecanismos de mercado, basado en la planificación socialista, siendo la empresa estatal socialista la principal forma de la economía nacional, aunque se promueve también las cooperativas, las pequeñas propiedades agrícolas, los usufructuarios y los trabajadores independientes, todo ello acompañado de una mejor eficiencia en la gestión de las actividades económicas tanto presupuestadas como empresariales, dando a esta última mayores facultades en su gestión; se identifica como prioridad la producción de alimentos, se enfatiza la necesidad de incrementar la cultura económica en la población y perfeccionar la educación, la salud, la cultura y el deporte, también se orienta a trabajar en una política de precios centralizada, acompañada de una política fiscal que garantice que se cumplan los objetivos estratégicos trazados.

De los 274 lineamientos aprobados, 81 cuentan con la participación del MES para un 26 por ciento, de ellos, 21 pertenecen al capítulo 1 "Modelo de Gestión Económica" y 20 al capítulo VI "Política Social" para un 26 y 25 por ciento respectivamente (Ministerio de Educación Superior, 2017).

Principales impactos:

Integración de las Universidades (Lineamientos 140, 143, 145, 146, 147 y 152)

- En todas las universidades se manifiesta un incremento de la calidad del proceso docente educativo y sus indicadores.
- Los claustros han sido impactados positivamente en aspectos tan importantes como el trabajo metodológico, en la composición de los departamentos básicos, en las categorías docentes y los grados científicos, y en la preparación pedagógica y profesional.
- Disminución de más del 40% de los cargos de cuadros que existía en las instituciones antes de la integración y de más del 60% en los Centros Universitarios Municipales (CUM). Disminuyen los cuadros nombrados por excepción y mejora significativamente el completamiento de la plantilla de cargos.

- El enfoque multidisciplinario del trabajo de ciencia e innovación ha potenciado que sea más vinculado a los problemas del territorio, provincias y municipios, y exhibe mejores indicadores.
- Se incrementaron los cursos de postgrado y la cobertura de los mismos creciendo su interdisciplinariedad y los programas de postgrados académicos, sobre todo maestrías y diplomados.
- La extensión universitaria se vio también beneficiada con un enfoque multidisciplinario, una respuesta más integral a los problemas comunitarios y una mayor incorporación de los estudiantes a las tareas de impacto y a los movimientos cultural y deportivo.

1.4 Planeación Estratégica en la Universidad de las Ciencias Informáticas.

La universidad es un sistema de procesos interrelacionados que cuenta con tres niveles de decisión: nivel Rector, Facultades–Centros/Filiales universitarias municipales (CUM/FUM) y departamentos (ver figura 2). El nivel rector tiene la función de dirigir los procesos desde la perspectiva global de la universidad, por lo que se trazan estrategias y objetivos que se derivan a nivel de facultades y departamentos. Se refleja como la gestión de los procesos estratégicos es responsabilidad del nivel rector con la participación de las facultades y los Consejos Universitarios Municipales (CUM), se consideran como procesos que se deben gestionar con un enfoque estratégico: la gestión de la calidad, y la gestión del capital humano (GCH), debido a que estos son los que proporcionan el desarrollo de la universidad (Ortiz, 2014).



Figura. 2. Niveles de decisión de la gestión de los procesos universitarios. (Zaldívar, Campaña & Pérez, 2017)

La Universidad de las Ciencias Informáticas forma un profesional con valores ético-morales y humanistas, comprometido con la Revolución y su Patria, que posee cultura general, habilidades y conocimientos al nivel de las mejores universidades de América Latina.

La UCI cuenta con un claustro estable y sus profesores luchan por la excelencia, por dominar y estar actualizados en las ciencias de su experticia y en las categorías y métodos marxistas, tienen prestigio científico y son un ejemplo de ética y de revolucionario. Desarrollan en sus clases el debate científico, político e ideológico. Los profesores de más experiencia forman nuevas generaciones de docentes.

Los profesionales de la rama de las ciencias informáticas del país encuentran en la UCI, la posibilidad de actualización sistemática al más alto nivel, con un enfoque marxista y ajustada a los intereses de la política nacional.

La UCI, líder nacional científico técnico en informática, sobre la base de la fusión en los procesos de formación-producción-investigación como modelo en la gestión, internalización y generalización de los resultados de los proyectos de investigación, a través de una red de

centros de investigación, desarrollo e innovación. La investigación científica es realizada por profesores, investigadores y estudiantes, vinculada estrechamente a las demandas de los Organismos de la Administración Central del Estado (OACE), generando impactos en lo económico y lo social, en áreas priorizadas.

La UCI a nivel latinoamericano es líder en productos de software y soluciones informáticas, tiene relaciones con empresas de software de primer nivel en el mundo y cuenta con un modelo de producción de software basado en modelos internacionales de calidad, es centro de referencia nacional de calidad de software y se sustenta sobre plataformas de software libre.

La Planeación Estratégica de la UCI para el período 2017-2021 recoge 5 Áreas de Resultados claves y estas a su vez tienen objetivos y criterios de medida, para garantizar el cumplimiento de las metas y proyecciones de la universidad

- ARC1: Formación de pregrado
- ARC 2: Formación de posgrado
- ARC 3: Ciencia, Tecnología e Innovación
- ARC 4: Recursos humanos
- ARC 5: Desarrollo y comercialización de aplicaciones y servicios informáticos y académicos

Las facultades realizan una desagregación de los objetivos de la universidad asumiendo la responsabilidad en el cumplimiento de una parte de los resultados.

1.5 Análisis del Estado del Arte

ISOTools Estrategia

Software online que permite establecer relaciones causa-efecto entre ellos y ver gráficamente dicha relación. Propicia conocer el grado de cumplimiento de cada uno de los objetivos y sabrá con exactitud el estado de la estrategia. Además, podrá consultar la relación entre los objetivos, los indicadores y los proyectos asociados. ISOTools permite crear múltiples cuadros de mando, relacionados o no entre sí (ISOTools, 2018).

Entre sus funcionalidades se encuentra:

Agenda de actividades

- Comunicación interna
- Entidades
- Formatos y Registros
- Gestor documental
- Indicadores



Figura. 3. Interfaz Principal de la herramienta online ISOTools Estrategia.

Existe un grupo de herramientas desarrolladas bajo la metodología **Balanced Scorecard Collaborative Inc**, internacionalmente adoptada para la gestión de los objetivos estratégicos de una institución, entre las principales destacan:

- Cognos: Cognos Metrics Manager Como su nombre lo indica, el software está muy orientado y tiene fortaleza hacia la métrica (performance scorecarding, KPI's). Tiene un muy buen visualizador para el dashboard.
- Corvu: CORVU BSC Solutions, uno de los primeros en el mercado, con una solución de mucha fortaleza. Inicialmente muy focalizado a la gestión y medición. Hoy en día una solución completa.

- Crystal Decisions Crystal Performance Scorecard: una aplicación que se integra con otras soluciones de la misma empresa y soluciones ERP de otros proveedores.
- Fiber FlexSI: Solución originaria de Brasil. No se obtuvo mayor información puesto que no es pública.

Existe una variedad de aplicaciones Android para la gestión de los objetivos estratégicos y la planeación estratégica de una organización, se seleccionaron 3 de acuerdo al grado de aceptación de los mismos en las plataformas donde se encuentran disponibles.

QuickScore

QuickScore es una solución de gestión de rendimiento estratégico que emplea metodologías de indicadores estratégicos equilibrados para automatizar los procesos en toda la organización. La plataforma se puede implementar de manera local o alojar en un entorno en la nube. Permite a los usuarios agregar metas/objetivos y compararlos con los valores de rendimiento reales.

Las principales funciones de QuickScore incluyen automatización de indicadores estratégicos equilibrados, informes personalizados, alertas, medición del rendimiento y evaluación comparativa. Los usuarios pueden generar automáticamente mapas de estrategia, redimensionar o personalizar la apariencia, agregar logotipos y obtener visibilidad del rendimiento en las distintas perspectivas, medidas u objetivos. Además, las empresas pueden usar la plataforma para evaluar las medidas de bajo nivel en relación con las medidas ponderadas empleando fuentes de datos automatizadas, puntuaciones o códigos de color (QuickScore, 2019).

KPI Fire

KPI Fire es un software integral de planificación y ejecución de estrategias para gestión de proyectos, creación de tableros de KPI, establecimiento y seguimiento de objetivos, colaboración en ideas, y mucho más. La solución basada en la nube es adecuada para todo tipo de equipos en los sectores de fabricación, servicios y atención médica, entre otros. *KPI Fire* está diseñado para ayudar a las empresas a alinear a los equipos y lograr los objetivos al proporcionar a los usuarios las herramientas para ingresar sus objetivos anuales y planes

estratégicos, y luego descomponer esos objetivos en objetivos departamentales, objetivos de equipo y, finalmente, proyectos.

KPI Fire proporciona información general de los indicadores de rendimiento clave (KPI) en tiempo real a través de paneles de KPI personalizados que permiten a todos los usuarios realizar un seguimiento del progreso del objetivo. Las empresas pueden personalizar completamente el panel de KPI para mostrar las métricas que son más importantes para su estrategia y cualquier cambio realizado a los KPI, o datos ingresados en el sistema, se actualizan en tiempo real para todos los usuarios. Gracias a la presentación visual de datos que ofrecen los paneles de *KPI Fire*, los usuarios pueden identificar qué proyectos necesitan más trabajo y dónde deben realizarse ajustes para alcanzar los objetivos (*KPI Fire*, 2019).

Adaptive Insights

Adaptive Insights, es una solución de planificación empresarial para elaboración de presupuestos, pronósticos e informes. Con *Adaptive Insights for Finance*, se obtiene un proceso de planificación activo para impulsar el éxito empresarial. Un proceso de planificación activo permite colaborar con los socios comerciales, proporciona tableros integrales y capacidades de modelado y permite hacer planes y elaborar pronósticos de forma continua. Más de 3800 empresas utilizan este software para modelar, supervisar, analizar e informar sobre el rendimiento de la organización. *Adaptive Insights* se posicionó como uno de los líderes en el Cuadrante Mágico de 2019 de Gartner para Soluciones en la nube de planificación y análisis financieros (*Adaptive Insights*, 2019).

1.5.1 Análisis de las herramientas identificadas

Las herramientas analizadas anteriormente son importantes en su área de acción, pues solucionan el problema para el que fueron desarrolladas, sin embargo, son soluciones a la medida. Entre las limitantes fundamentales asociadas a estas herramientas se encuentra el limitado acceso a su documentación ingenieril o código fuente. Esto provoca que no sea posible desarrollar extensiones que propicien adaptarlas a las necesidades de esta investigación. Cuba aspira a partir del proceso de informatización la independencia tecnológica de sus plataformas y sistemas de información a nivel de país.

Al realizar un análisis funcional sobre las herramientas se afirma que los requerimientos funcionales básicos especificados en la mayoría de estas herramientas se dividen en 6 grupos. En todo caso parte del diseño de un Cuadro de Mando Integral (CMI), las aplicaciones permiten desarrollar todas las etapas del diseño del CMI:

- 1) ver las estrategias desde distintas perspectivas.
- 2) identificar los objetivos estratégicos para cada perspectiva.
- 3) asociar medidas con objetivos estratégicos.
- 4) encadenar objetivos estratégicos con relaciones causa efecto.
- 5) asignar metas a medidas.
- 6) listar iniciativas estratégicas.

Estos elementos son de importancia para esta investigación debido a que propicia una idea del comportamiento de herramientas internacionales en la planeación estratégica de organismos e instituciones. Los cuales pueden ser reutilizados en el desarrollo de la propuesta.

1.6 Metodologías de desarrollo de software

El desarrollo de software, es uno de los sectores tecnológicos más competitivos, sin embargo, ha tenido una evolución constante en lo que se refiere a las metodologías con el objetivo de mejorar, optimizar procesos y ofrecer una mejor calidad. Las metodologías constituyen un enfoque estructurado para el desarrollo de software que incluyen modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos. Estas metodologías se clasifican en dos grandes grupos: tradicionales y ágiles (Gómez, López y Bacalla, 2010).

Las metodologías tradicionales se caracterizan por (Ojeda y Fuentes, 2012):

- Documentación exhaustiva de todo el proyecto.
- Los costos son altos al implementar un cambio.
- No es una buena solución para entornos volátiles.

- El equipo de desarrollo debe ser grande.

Mientras que las metodologías ágiles se caracterizan por (Canós, 2012):

- Buena solución para proyectos a corto plazo.
- Facilidad de respuesta a cambios repentinos en el desarrollo.
- Equipos de desarrollo pequeños.
- Ciclos de desarrollo cortos.

1.6.1 Metodología AUP-UCI.

Al no existir una metodología de *software* universal, debido a que toda metodología debe ser adaptada a las características de cada proyecto, se decide hacer una variación de la metodología Proceso Unificado Ágil (AUP, por sus siglas en inglés) en la UCI, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la universidad. De las cuatro fases que propone AUP: Inicio, Elaboración, Construcción y Transición, se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola que es Ejecución y se agrega la fase de Cierre (Garay, 2016).

AUP propone siete disciplinas: Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno, por lo que se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también (Sánchez, 2014). A partir de que la disciplina de Modelado de negocio propone tres variantes a utilizar en los proyectos: casos de uso del negocio, descripción de proceso de negocio y modelo conceptual, y existen tres formas de encapsular los requisitos: casos de uso del sistema, historias de usuario y descripción de requisitos por proceso, surgen cuatro escenarios para modelar el sistema en los proyectos, quedando de la siguiente forma:

Escenario 1: proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario 2: proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario 3: proyectos que modelen el negocio con descripción de proceso de negocio solo pueden modelar el sistema con descripción de requisitos por proceso.

Escenario 4: proyectos que no modelen negocio solo pueden modelar el sistema con HU.

En la presente investigación se utilizará AUP-UCI como metodología de desarrollo de software en su escenario 2 debido a las características del problema de investigación y su entorno de solución, el alcance del proyecto y cantidad de recursos humanos. Además, se tiene en cuenta la estandarización y experiencia productiva en la universidad aplicando esta metodología.

1.7 Ambiente de desarrollo.

En este epígrafe de caracterizan las Tecnologías, herramientas y lenguajes de desarrollo. Se adoptan las tecnologías y herramientas definidas por la Dirección de Informatización de la UCI, en específico las definidas para el desarrollo del Sistema de control y seguimiento de la Planeación Estratégica de la Facultad 1, a la que se desea integrar el presente trabajo. Como lenguaje de programación Python, Django como marco de trabajo, SQLite como gestor de base de datos, las cuales se describen a continuación.

SQLite

SQLite está construida en C, lo cual facilita la migración a diversas plataformas de sistemas operativos y de dispositivos. Dado que una base de datos de SQLite se almacena por completo en un solo archivo, está puede ser exportada a cualquier otra plataforma y tener interoperabilidad al 100% sin ningún requerimiento de programación adicional o cambios de configuración (Fernández Rabilero & Varona Carmenates, 2016).

Gracias a que SQLite es software libre, es posible encontrar una gran cantidad de componentes, librerías y drivers para interactuar con SQLite desde una gran diversidad de lenguajes y plataformas de programación. Ya sea que estemos utilizando lenguajes modernos como Java, Perl, Python, PHP, Ruby, C#, lenguajes más antiguos como Pascal, SmallTalk, Clipper, o lenguajes poco conocidos como Suneido, REXX, S-Lang, para todos podemos encontrar librerías y ejemplos de código para SQLite.

- SQLite puede ser utilizado de dos formas diferentes que se muestran a continuación Como gestor de base de datos local en un PC. De esta forma podemos gestionar bases de datos con SQLite igual que si estuviéramos trabajando con un sistema gestor de base de datos como MySQL únicamente instalando un archivo ejecutable de tamaño considerablemente pequeño.
- Como una extensión de PHP. La ventaja de esta opción es que no necesita tener instalado o conectar con un servidor de base de datos dado que ofrece una interfaz prácticamente con las mismas funcionalidades que un gestor de bases de datos cualquiera donde los datos serán almacenados en un texto plano.

HTML 5

HTML 5 es la versión más nueva del lenguaje de marcado HTML (*HyperText Markup Language*) y es usado para la estructuración y organización de los contenidos de una página, se podría decir que es el esqueleto de una página web.

Permite escribir texto de forma estructurada y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento. Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, y el resultado puede considerarse como un documento multimedia. Los documentos HTML deben tener la extensión HTML o htm, para que puedan ser visualizados en los navegadores. Los navegadores se encargan de interpretar el código HTML de los documentos y de mostrar a los usuarios las páginas web resultantes del código interpretado. El HTML desarrolla el lenguaje HTML con mecanismos para hojas de estilo, ejecución de scripts, marcos, objetos incluidos, soporte mejorado para texto de derecha a izquierda y direcciones mezcladas, tablas más ricas y mejoras en formularios, ofreciendo mejoras de accesibilidad para personas con discapacidades (Fernández Rabilero & Varona Carmenates, 2016).

Con CCS3 propicia hacer una variedad de animaciones y efectos para realizar diseños sofisticados y sorprendentes con mínimos recursos de procesamiento. Se ha ampliado para ser capaz de manejar elementos de estilo de una manera mucho más compleja. Se pueden agregar sombras, bordes redondeados, transiciones, uso de tipografías externas, mejoras en el manejo de columnas y cuadros de distribución flexible.

Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma (Lutz, 2001).

- Python usa tipado dinámico y conteo de referencias para la administración de memoria.
- Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).
- Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.
- Aunque la programación en Python podría considerarse en algunas situaciones hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme.

Django

Los orígenes de Django en la administración de páginas de noticias son evidentes en su diseño, ya que proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas hechas con Django a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos).

La distribución principal de Django también aglutina aplicaciones que proporcionan un sistema de comentarios, herramientas para syndicar contenido via RSS y/o Atom, "páginas planas" que permiten gestionar páginas de contenido sin necesidad de escribir controladores o vistas para esas páginas, y un sistema de redirección de URL (Holovaty, & Kaplan-Moss, 2009).

Otras características de Django son:

- Un mapeador objeto-relacional.
- Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django.
- Una API de base de datos robusta.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URL basado en expresiones regulares.
- Un sistema "middleware" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes middleware que proporcionan cacheo, compresión de la salida, normalización de URL, protección CSRF y soporte de sesiones.
- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

Aunque Django está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador, sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar estrictamente ningún paradigma particular, y en cambio prefieren hacer "lo que les parece correcto". Como resultado, por ejemplo, lo que se llamaría "controlador"

en un "verdadero" *framework* MVC se llama en Django "vista", y lo que se llamaría "vista" se llama "plantilla".

Modelo: Una de las partes más potentes de Django, su modelo de datos. Cada uno de los modelos creados se mapean en diferentes tablas en la Base de Datos. Esto permite aislar la Base de Datos del código y olvidarte de los diferentes *select* y *updates* a veces tan tediosos.

Vista: La capa de presentación se basa en plantillas *HTML*. Django presenta un *template engine* y un *template loader* muy potente que permite presentar al usuario diversas páginas *HTML* usando una base como plantilla. Esto es posible porque en cada una de las plantillas se pueden introducir determinadas etiquetas Django que el *template loader* se encargará de interpretar.

Controlador: Es lo que en Django se llama *views*. Puede llevar a la confusión, aunque Django lo llame *views*, éstas son las que actúan como controlador. Escritas en puro código Python cada *view* atenderá una petición *HTML* según el mapeo de *URL* del que ya se hablará más adelante.

1.8 Conclusiones parciales.

El análisis de las herramientas existentes para la planeación de los objetivos estratégicos posibilita la identificación de que ninguna de estas presenta las funcionalidades necesarias para su gestión en la Facultad 1 de la UCI, sin embargo, se identificaron funcionalidades de interés, por lo que se decide el desarrollo de un sistema que permita cubrir esta problemática.

El análisis de las tecnologías utilizadas para el desarrollo de aplicaciones de estas características, propicia recopilar los basamentos teóricos para su selección.

Se realiza la descripción del proceso de planeación estratégica en Cuba, lo que permite el análisis y argumentación del comportamiento funcional de la misma a diferentes niveles.

Capítulo 2: Características del Sistema

En el presente capítulo se presentan los objetivos estratégicos de la organización, el flujo actual de los procesos involucrados en el campo de acción, descripción de los procesos que serán objeto de automatización, se modela el negocio propuesto, así como las particularidades del diseño. Se plantean los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso del sistema.

1.1 Modelo conceptual

Un modelo conceptual es la relación entre los conceptos de todos los temas relacionados con un problema específico. En él se describen las distintas entidades, sus atributos, papeles y relaciones, además de las restricciones que rigen el dominio del problema (Pressman, 2010).

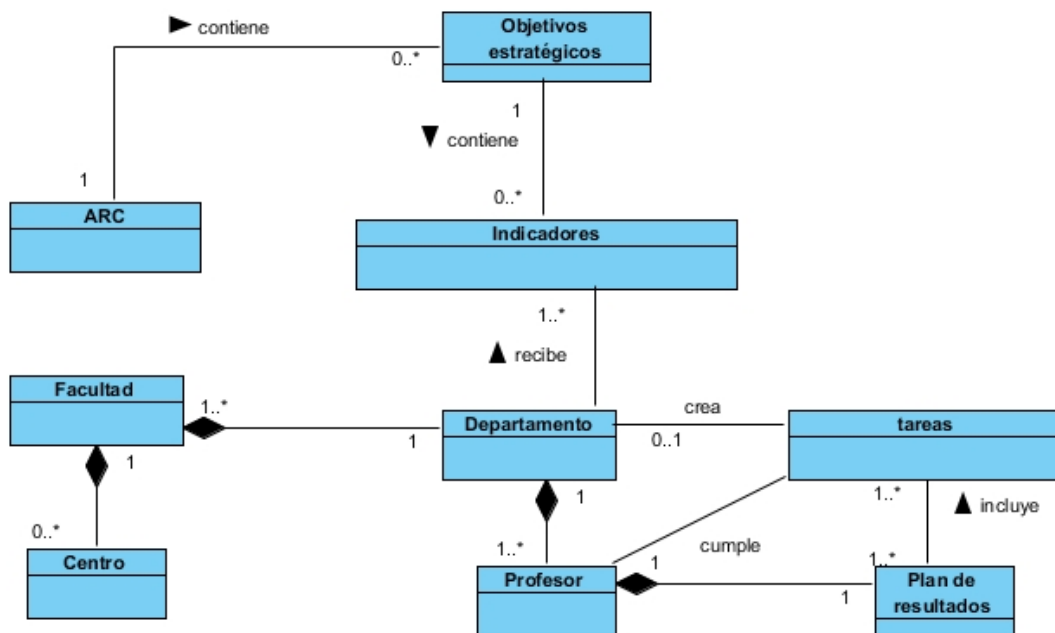


Figura. 4. Modelo conceptual. Elaboración propia

La figura 4 describe el modelo conceptual de la propuesta de solución. A la facultad se le asocian departamentos y centros de desarrollos, a los departamentos se le asocian profesores que tienen un Plan de resultados y tareas a ejecutar para su cumplimiento. La

facultad y los departamentos trabajan para el cumplimiento de indicadores asociados a objetivos estratégicos que a su vez responden a Áreas de Resultados Claves.

1.2 Requisitos de la propuesta de solución

Según Sommerville (2005) los requisitos para un sistema de software determinan lo que debe hacer el sistema y definen las restricciones en su funcionamiento e implementación, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo.

Técnicas de obtención de requisitos utilizadas:

Existe un número importante de técnicas para obtener requerimientos. A continuación, se describen las técnicas utilizadas en este proyecto.

Entrevista: La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista.

Análisis documental: Varios tipos de documentación, como manuales y reportes, pueden proporcionar al analista información valiosa con respecto a las organizaciones y a sus operaciones. La documentación difícilmente refleja la forma en que realmente se desarrollan las actividades, o donde se encuentra el poder de la toma de decisiones. Sin embargo, puede ser de gran importancia para introducir al analista al dominio de operación y el vocabulario que utiliza.

1.2.1 Requisitos funcionales de la propuesta de solución

Los requisitos funcionales son definidos por el equipo de desarrollo teniendo en cuenta la información brindada por el cliente. Partiendo de esta información se elaboran los documentos de especificación de requisitos y finalmente se valora y valida la información en busca de errores, inconsistencias o faltas para evitar omitir algún requerimiento del

cliente (Pressman, 2010). A partir de las técnicas de captura de requisitos descritas anteriormente se obtuvieron 38 requisitos funcionales, resumidos en la Tabla 1.

Tabla 1. Requisitos funcionales de la propuesta de solución.

No.	Nombre	Descripción	Prioridad	Complejidad
RF1	Crear usuario	El sistema permite crear un usuario con los permisos asociados.	Media	Baja
RF2	Modificar usuario	El sistema permite modificar los datos de un usuario.	Media	Baja
RF3	Eliminar usuario	El sistema permite eliminar los datos de un usuario.	Media	Baja
RF4	Listar usuarios	El sistema permite generar un listado de los usuarios registrados	Media	Baja
RF5	Ver detalles de usuario	El sistema permite visualizar los datos asociados a un usuario.	Media	Baja
RF6	Autenticar en el sistema	El sistema permite autenticar a un usuario registrado usando sus credenciales.	Media	Baja
RF7	Cerrar sesión	El sistema permite cerrar una sesión activa.	Media	Baja
RF8	Cambiar contraseña	El sistema permite a un usuario cambiar el valor de sus credenciales	Media	Baja
RF9	Insertar Área	El sistema propicia la inserción de un área de la facultad (departamento, centro de desarrollo)	Media	Baja
RF10	Modificar Área	El sistema propicia la modificación de un área de la facultad (departamento, centro de desarrollo)	Media	Baja
RF11	Eliminar Área	El sistema permite la eliminación de un área de la facultad (departamento,	Media	Baja

		centro de desarrollo)		
RF1 2	Listar Áreas	El sistema permite generar un listado de las áreas asociadas a la facultad (departamento, centro de desarrollo)	Media	Baja
RF1 3	Adicionar proceso	El sistema permite al jefe de departamento adicionar un proceso (área de resultado clave).	Alta	Media
RF1 4	Ver detalles de proceso	El sistema permite visualizar los datos asociados a un proceso.	Media	Baja
RF1 5	Modificar proceso	El sistema permite la modificación de un proceso de la facultad (departamento, centro de desarrollo)	Media	Baja
RF1 6	Listar procesos	El sistema permite generar un listado de los procesos.	Media	Baja
RF1 7	Adicionar objetivo estratégico	El sistema permite al jefe de departamento adicionar un objetivo estratégico.	Alta	Media
RF1 8	Ver detalles de un objetivo	El sistema permite visualizar los datos asociados a un objetivo.	Media	Baja
RF1 9	Asociar indicador a objetivo estratégico	El sistema permite al jefe de departamento adicionar un indicador a objetivo estratégico.	Alta	Media
RF2 0	Modificar Indicador	El sistema permite al jefe de departamento modificar un indicador.	Baja	Baja
RF2 1	Eliminar Indicador	El sistema permite al jefe de departamento eliminar un indicador.	Baja	Baja
R F	Mostrar porcentaje de cumplimiento de indicadores de	El sistema permite mostrar los porcentajes de cumplimiento de los indicadores asociados a un	Media	Baja

2 2	un profesor	profesor		
R F 2 3	Mostrar porcentaje de cumplimiento de indicadores del departamento	El sistema permite mostrar los porcentos de cumplimiento de los indicadores asociados a un departamento	Media	Baja
R F 2 4	Mostrar porcentaje de cumplimiento de indicadores de la facultad	El sistema permite mostrar los porcentos de cumplimiento de los indicadores asociados a la facultad. (proceso, objetivo, indicadores, meta, % de cumplimiento)	Media	Baja
R F 2 5	Asignar un indicador al profesor	El sistema permite al jefe de departamento asignar el cumplimiento de los indicadores de cada uno de sus profesores.	Alta	Media
R F 2 6	Modificar indicador asociado a profesor	El sistema permite modificar los datos de un indicador asociado a un profesor	Baja	Baja
R F 2 7	Desvincular indicador asociado a un profesor	El sistema permite un indicador asociado a un profesor	Baja	Baja
RF2 8	Asignar tarea a profesor	El sistema permite asignar una nueva tarea a un profesor	Alta	Media
RF2 9	Modificar tarea asignada a profesor	El sistema permite modificar una tarea existente a un profesor	Alta	Media
RF3 0	Eliminar tarea asignada a profesor	El sistema permite eliminar una tarea existente a un profesor	Baja	Baja

R F 3 1	Revisión de tareas por el jefe de departamento	El sistema permite al jefe de departamento evaluar y revisar el % de cumplimiento de la tarea.	Alta	Media
R F 3 2	Listar cumplimiento de las tareas asignadas	El sistema permite mostrar el estado en que se encuentran las tareas (cumplida o no cumplida, y % de cumplimiento)	Media	Baja
R F 3 3	Modificar porcentaje de cumplimiento de una tarea	El sistema permite modificar el % de cumplimiento de una tarea	Alta	Media
R F 3 4	Generar plan de resultados del profesor	El sistema permite gestionar y construir el plan de resultados del profesor (está constituido por objetivos, indicadores y tareas)	Alta	Media
R F 3 5	Modificar plan de resultados del profesor	El sistema permite modificar la información asociada al plan de resultados del profesor	Alta	Media
R F 3 6	Eliminar información del plan de resultados del profesor	El sistema permite eliminar información del plan de resultados del profesor	Alta	Media
R F 3 7	Notificar plazo de cumplimiento de la tarea	El sistema notifica o alerta al profesor el plazo de cumplimiento de una tarea dada.	Baja	Media
R F	Mostrar vista general de la planeación	El sistema muestra cómo queda organizado el plan estratégico de la facultad	Alta	Media

3 8	estratégica de la facultad	(proceso, objetivo, indicadores, meta, % de cumplimiento)		
--------	----------------------------	---	--	--

Para validar los requisitos especificados se empleó la técnica de Prototipos, además de aplicar los Criterios para validar los requisitos del cliente y realizar el Acta de aceptación de los requisitos, según el modelo de desarrollo de AUP UCI.

1.2.2 Requisitos no funcionales de la propuesta de solución

Los requisitos no funcionales especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características del funcionamiento (Pressman, 2010).

Las características del sistema son propiedades o cualidades que el producto presenta. A diferencia de las funcionalidades, estas son de mayor peso para la arquitectura del sistema y de no cumplirse afectan el funcionamiento del software. Definen apariencia, usabilidad, rapidez y confiabilidad.

Usabilidad: Cada sección del sistema, en la interfaz, debe comenzar con un título o encabezamiento que describa el contenido de la pantalla. En procesos con formularios y pantallas múltiples, cada página debe estar etiquetada para mostrar su relación con las otras. Cuando se introducen datos en la pantalla la terminología utilizada debe ser familiar para los usuarios.

Apariencia o interfaz externa: La presente investigación genera el artefacto Arquitectura de Información para la aplicación con la finalidad de estructurar, organizar y etiquetar el contenido y los elementos de navegación para facilitar la navegación de los usuarios.

Seguridad: La información manejada por el sistema está protegida de acceso no autorizado y divulgación. Existe un único usuario con el rol de administrador del sistema que otorga los permisos a los demás usuarios según su rol.

Portabilidad: La aplicación debe ser multiplataforma, es decir, debe poder ejecutarse sobre los Sistemas Operativos Windows y GNU Linux.

RNF1: Soporte

- El sistema debe ejecutarse sobre los gestores de bases de datos MySQL y PostgreSQL.
- El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

RNF3: Seguridad

- Se define el acceso al sistema y sus funcionalidades mediante la asignación de permisos por roles de usuarios.
- La carpeta donde se encuentre el sistema solo tendrá permiso de lectura.
- Los errores deben mostrar la menor cantidad de detalles posible, para evitar brindar información que comprometa la seguridad e integridad del sistema.

RNF5: Apariencia o interfaz externa

- El sistema debe ser compatible con los navegadores Chrome, Firefox, Safari, Opera e Internet Explorer a partir de su versión 9.
- El sistema debe poseer un diseño web adaptable.

1.3 Diagrama de casos de uso del sistema

Las interacciones entre el sistema y sus usuarios, son representadas por los diagramas de caso de uso, los cuales determinan las funciones a ejecutar por el sistema. Las Figuras 5 y 6 muestran los Diagramas de Caso de Uso correspondientes a Gestionar Objetivos Estratégicos y Gestionar Indicadores.



Figura 5. Diagrama de Caso de Uso correspondiente a Gestionar Objetivos Estratégicos.

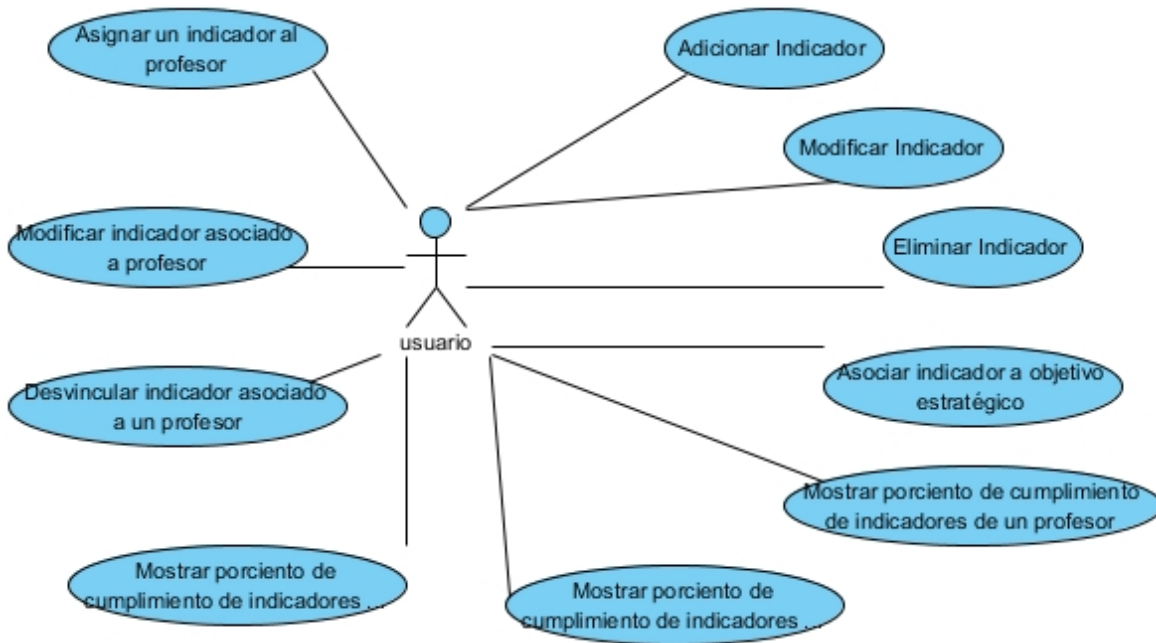


Figura 5. Diagrama de Caso de Uso correspondiente a Gestionar Indicadores.

Tabla 2. Descripción del caso de uso Gestionar Objetivos Estratégicos.

Caso de uso	Gestionar Objetivos Estratégicos
Objetivo	El caso de uso permite gestionar la información asociada a los objetivos estratégicos de la facultad.
Actores	Decano, Jefe de Departamento.
Resumen	El caso de uso comienza cuando el Decano o el Jefe de Departamento acceden al sistema para gestionar la información asociada a los objetivos.
Referencias	RF17, RF18 y RF19
Precondición	El usuario debe haberse autenticado al sistema y accedido a la gestión de objetivos
Postcondición	Se gestionó la información asociada a los objetivos
Flujo de eventos	
Flujo básico "Gestionar Objetivos Estratégicos"	
	<ul style="list-style-type: none"> El usuario accede a las opciones del sistema:

- Adicionar Objetivo Estratégico, ver flujo alternativo 1.
- Ver detalles de Objetivo, ver flujo alternativo 2.
- Asociar indicador a objetivo, ver flujo alternativo 3.

Flujo alternativo 1 "Adicionar Objetivo estratégico"

El Sistema brinda la posibilidad de introducir los siguientes datos:

- Área
- Detalles
- Nombre

El usuario introduce los datos y acepta.

Flujo alternativo 2 "Ver detalles Objetivo estratégico"

El Sistema brinda la posibilidad de ver la información asociada a un objetivo:

- Detalles
- Nombre

El usuario acepta.

El sistema regresa a la interfaz anterior.

Flujo alternativo 3 "Asociar Indicador a objetivo"

El Sistema brinda la posibilidad de asociar indicadores a objetivo:

El usuario selecciona el indicador y asocia.

El usuario acepta y regresa a la interfaz anterior.

1.4 Descripción de la arquitectura

La arquitectura utilizada para la propuesta de solución es la Arquitectura Cliente-Servidor, un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas. El modelo Cliente/Servidor permite diversificar el trabajo que realiza cada aplicación, de forma que los Clientes no se sobrecarguen, cosa que ocurriría si ellos mismos desempeñan las funciones que le son proporcionadas de forma directa y transparente. En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes. (Provoste, 2019). La figura 7 muestra la representación de la

arquitectura:

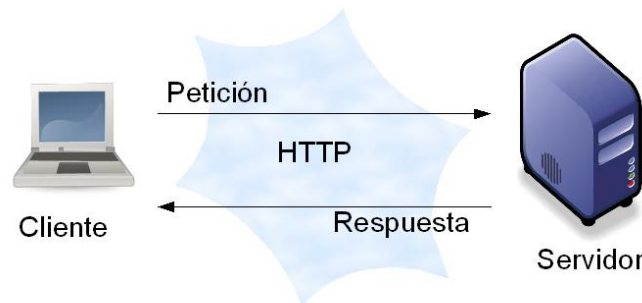


Figura 7. Arquitectura Cliente-Servidor

Cliente: es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de los recursos que desea obtener mediante HTTP. La parte cliente de las aplicaciones web suele estar formada por el código HTML que forma la página web más algo de código ejecutable realizado en lenguaje de script del navegador. La misión del cliente web es interpretar las páginas HTML y los diferentes recursos que contienen (Luján-Mora, 2001).

Servidor: es un programa que está esperando permanentemente las solicitudes de conexión por parte de los clientes web. La parte servidor de las aplicaciones web está formada por páginas estáticas que siempre muestran el mismo contenido y por programas o scripts que son ejecutados por el servidor web cuando el navegador del cliente solicita algunas páginas. La salida de este script suele ser una página HTML estándar que se envía al navegador del cliente (Luján-Mora, 2001).

1.4.1 Patrón arquitectónico

La arquitectura de la solución propuesta está basada en el patrón arquitectónico Modelo Vista Controlador (MVC). Este patrón garantiza la reducción del esfuerzo de programación, la cual es necesaria en la implementación de sistemas múltiples. Permite separar cada una de las capas del sistema: el modelo, donde se encuentran los datos y las reglas del negocio; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del usuario. (Sellarès, 2011)

A partir de la descripción realizada de las tecnologías en el anterior acápite, se evidencia la utilización del MVC de la siguiente manera:

- **El Modelo:** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista:** o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos de interacción con éste.
- **El Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

Este modelo de arquitectura presenta varias ventajas:

- Separación clara entre los componentes de un programa; lo cual permite su implementación por separado.
- Interfaz de Programación de Aplicaciones API (*Application Programming Interface*) muy bien definida; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- Conexión entre el Modelo y sus Vistas dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.

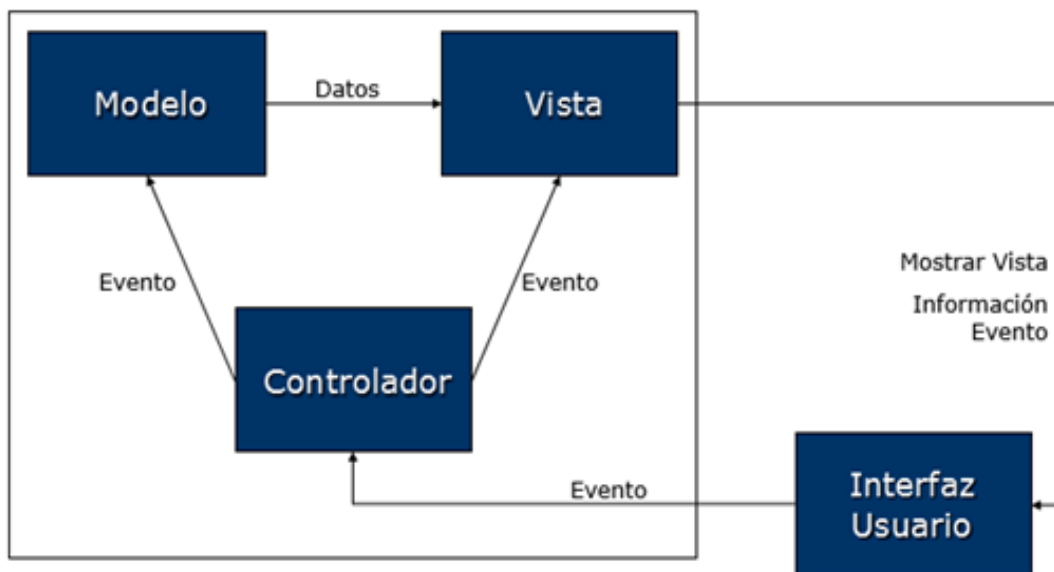


Figura 8. Arquitectura MVC. Fuente: elaboración propia.

1.4.2 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema. Esta solución provee una forma confiable, segura y probada para resolver problemas recurrentes en el diseño de software. Los patrones de diseño tienen como finalidad precisar en detalle los subsistemas y componentes de la aplicación. (Visconti y Astudillo, 2015)

Patrones GRASP

Los Patrones Generales de Software para Asignación de Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. Ayudan a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Las responsabilidades están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento (Tabares, 2014). A continuación, se describen los patrones GRASP utilizados en el desarrollo de la propuesta de solución:

Experto: propone asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplirla. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo. El uso del patrón se evidencia en las clases librerías, que son las que cuentan con la información necesaria para cumplir las responsabilidades sobre los elementos de negocio. Un ejemplo del uso del patrón en la propuesta de solución es la librería encargada de la información asociada a las tareas por indicadores, la que posee las responsabilidades relacionadas con la información de las tareas, ejemplo el cálculo del porciento con la implementación del método *“percent”*.

```
@admin.register(Task)
class TaskAdmin(admin.ModelAdmin):
    inlines = [ProgressInline]
    fields = ['name', 'details', 'indicator', 'professor']

    def percent(self, instance: Task):
        percent = 0
        try:
            percent = instance.progress
        except Exception as e:
            print(type(e))
            percent = 0
        finally:
            return percent
```

Figura. 9. Fragmento de código para mostrar el patrón Experto. Fuente: elaboración propia.

Creador: expresa la asignación a una clase la responsabilidad de crear una instancia de otra. El uso del patrón se evidencia en las clases *NormalTaskFactory* y *PermamentTaskFactory* (Figura 10) que se encargan de crear tareas específicas relacionadas a un dueño (“*owner*”). En la propuesta de solución cada clase controladora crea una instancia de las librerías y métodos que necesita para su funcionamiento y además utiliza campos propios para crear dichas instancias.


```

class NormalTask(Task):
    def __init__(self, owner):
        super().__init__(owner)
        self._percent = 0
    @property
    def completion_percent(self):
        return self._percent

class PermanentTask(Task):
    @property
    def completion_percent(self):
        return 0

class TaskFactory(ABC):
    def __init__(self, owner):
        self._owner = owner
    @abstractmethod
    def factory_method(self):
        pass
    def create(self):
        return self.factory_method()

class NormalTaskFactory(TaskFactory):
    def factory_method(self):
        return NormalTask(self._owner)

class PermanentTaskFactory(TaskFactory):
    def factory_method(self):
        return PermanentTask(self._owner)
  
```

Figura. 10. Fragmento de código para mostrar el patrón Creador. Fuente: elaboración propia.

Controlador: define que se le debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. El uso del patrón se evidencia en las clases controladoras que se encargan de obtener los datos y enviarlos a las librerías y las vistas, así como manejar los posibles errores o mensajes que se muestran. En la Figura 11, el menú de la izquierda define cual es la clase la controladora de la propuesta de solución, dicha controladora es la que recibirá las peticiones asociadas a esta funcionalidad.

```

35 @admin.register(KeyResultArea)
36 class KeyResultAdmin(admin.ModelAdmin):
37     inlines = [GoalInline]
  
```

Figura. 11. Fragmento de código para mostrar el patrón Controlador. Fuente: elaboración propia.

Alta cohesión: la cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una clase con baja cohesión hace muchas tareas no afines o un trabajo excesivo. En el ejemplo se ilustra un modelo cuya única responsabilidad es vincular las instancias de la clase `Center` con los objetos de la base de datos siendo parte del sistema ORM (Object Relational Model).

```
11 # Representación de un centro
12 class Center(models.Model):
13     name = models.CharField(max_length=250, verbose_name='Nombre')
14
15     class Meta:
16         verbose_name = 'Centro o departamento'
17         verbose_name_plural = 'Centros o departamentos'
18
19     def __str__(self):
20         return self.name
21
```

Figura. 12. Fragmento de código para mostrar el patrón Alta cohesión. Fuente: elaboración propia.

Patrones GoF

Los patrones Banda de los Cuatro (*Gang-of-Four*) describen las formas comunes en que diferentes tipos de objetos, pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases la formación de estructuras de mayor complejidad. Además, permiten crear grupos de objetos para ayudarnos a realizar tareas complejas. Existen tres tipos de patrones: de creación, estructurales y de comportamiento. Los patrones de creación abstraen la forma en la que se crean los objetos, permitiendo tratar las clases a crear de forma genérica dejando para más tarde la decisión de qué clases crear o cómo crearlas (Peña, 2016). El patrón *GoF* que se utilizó en el desarrollo del componente es:

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 (Peña, 2016). El objeto *DatabaseConector* tiene los atributos y métodos necesarios para crear la conectar el software con la base de datos, puesto que esta conexión es única (el software utiliza una sola base de datos) tiene sentido hacer de la clase una única instancia.

```

class MetaSingleton(type):
    instances={}
    def __call__(cls,*args, **kwargs):
        if cls not in cls.instances.keys():
            cls.instances[cls] = super().__call__(*args, **kwargs)
        return cls.instances[cls]

class DataBaseConector(SqliteConector, metaclass=MetaSingleton):
    pass
  
```

Figura. 13. Fragmento de código para mostrar el patrón Singleton. Fuente: elaboración propia

1.5 Diagrama de Clases del Diseño

El diagrama de clases del diseño especifica la estructura de clases del sistema con relaciones entre clases y estructuras de herencia. Es desarrollado buscando una solución ideal (Sellarès, 2011). A continuación, se muestra un fragmento del diagrama de clases del diseño del CU Gestionar Procesos y Gestionar Objetivos Estratégicos, basado en estereotipos web, que se realiza durante el proceso de desarrollo.

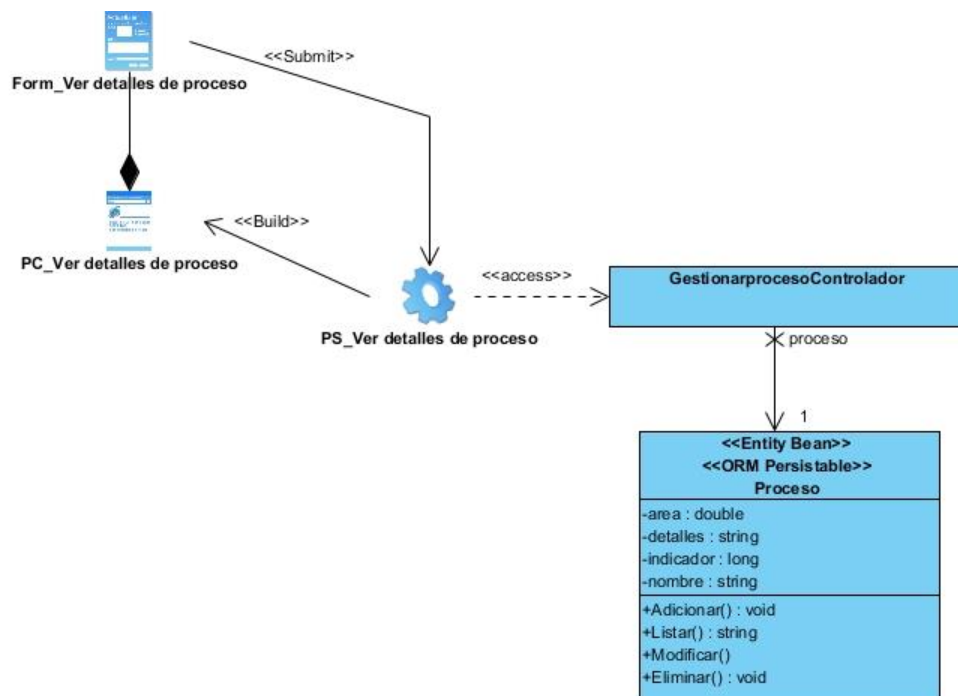


Figura. 14. Diagrama de clases del diseño del CU Gestionar Procesos. Fuente: elaboración propia.

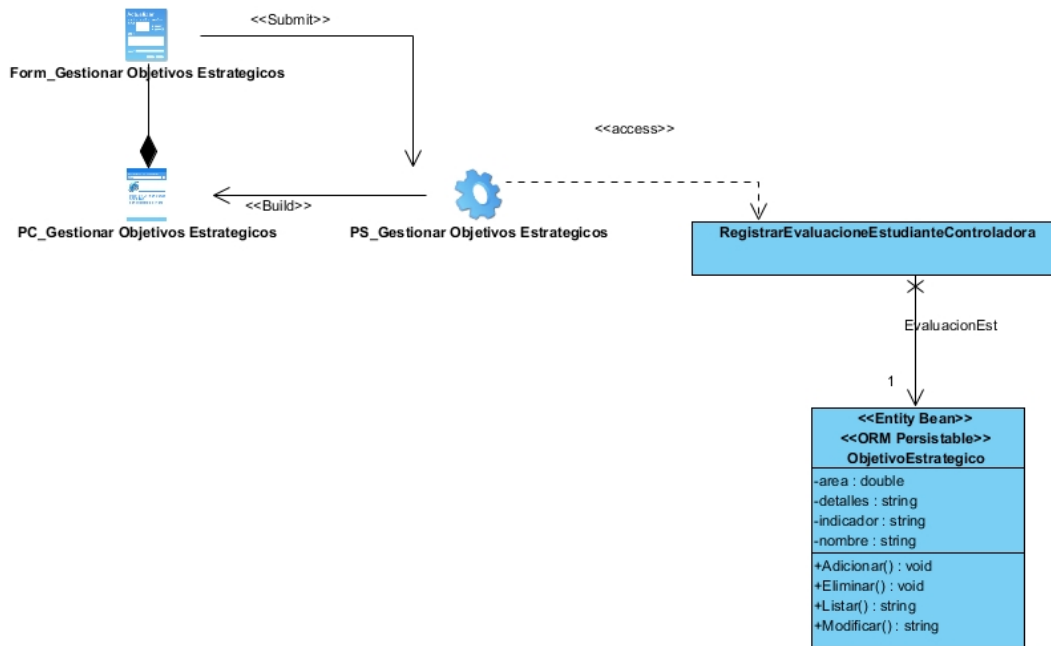


Figura. 15. Diagrama de clases del diseño del CU Gestionar Objetivos Estratégicos. Fuente: elaboración propia.

1.6 Diagrama de paquetes

Para la confección del modelo del diseño, se plantea una estructura de paquetes que sea manejable para la implementación. Cada uno de estos paquetes está compuesto por diversos subpaquetes que a su vez contienen los diagramas de clases del diseño. Todas las clases están agrupadas en el paquete Repositorio de clases. En Sesiones se encuentran todas las clases controladoras agrupadas en paquetes, donde un paquete tiene las controladoras autogeneradas, otro las personalizaciones que se hacen sobre las controladoras autogeneradas y uno para las controladoras propias del proceso. El paquete Entidades contiene a su vez otros paquetes con las entidades autogeneradas y personalizadas. Todas las vistas están contenidas en el paquete Vistas. Estos paquetes se relacionan entre ellos ya que las vistas consultan y actualizan las entidades e invocan a las controladoras y estas modifican las entidades.

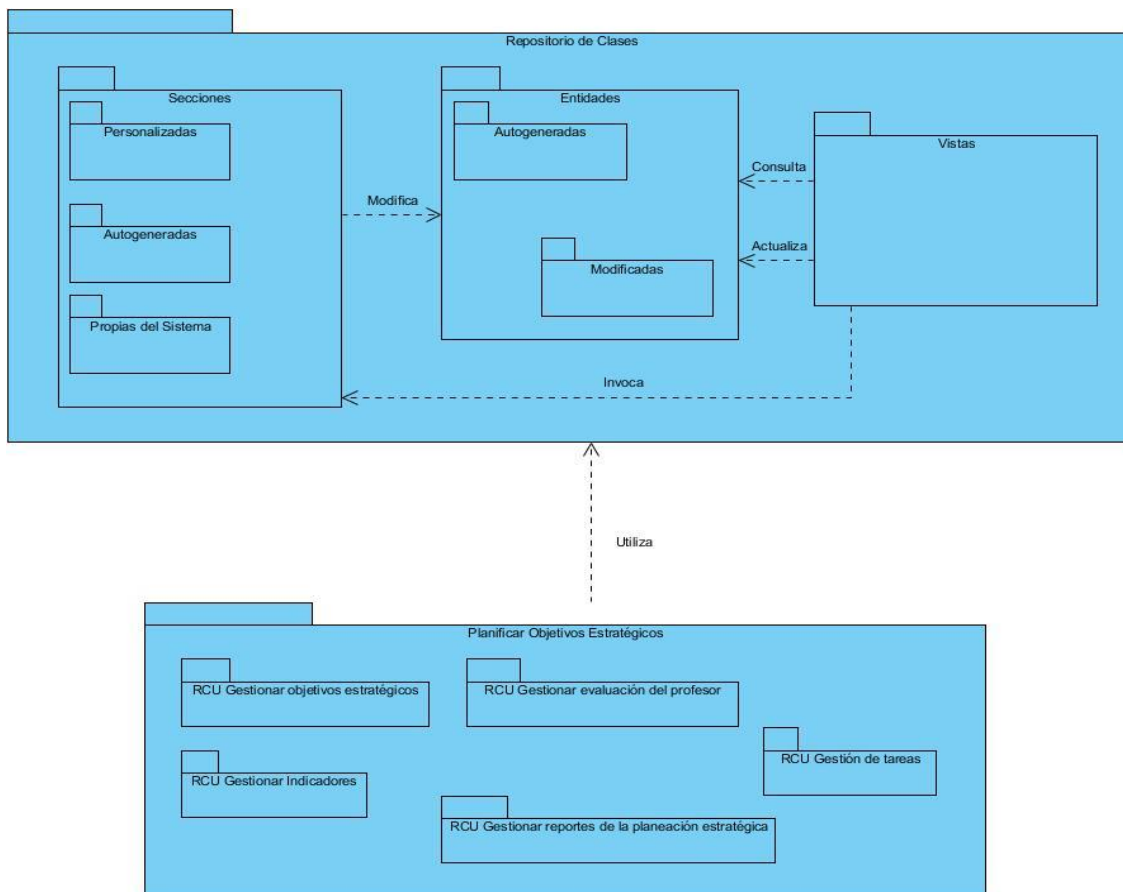


Figura. 16. Diagrama de paquetes. Fuente: Elaboración propia.

1.7 Conclusiones parciales

Luego de describir las características que debe cumplir el sistema, basado en tecnologías

libres y de realizar el análisis y diseño correspondiente, se especificaron las técnicas para la obtención de requisitos, las cuales Propician la comprensión, identificación y descripción de las exigencias funcionales y no funcionales que deberá cumplir la solución.

La utilización de los patrones de diseño y definición de la arquitectura permite el diseño de una solución robusta, flexible y escalable. A partir del análisis, el diseño realizado y de los artefactos generados quedan sentadas las bases para la implementación y validación de la solución propuesta.

Capítulo 3: Implementación y pruebas

En el presente capítulo después del diseño realizado en el capítulo anterior, se presenta el diagrama asociado al Modelo de Datos y Diagrama de Despliegue con el objetivo de facilitar una vista general de la solución. Se realizan las pruebas correspondientes para la validación del buen funcionamiento de la aplicación desarrollada y mostrar la calidad del resultado obtenido.

2.1 Modelo de datos

Un modelo de datos es un conjunto de conceptos que sirven para describir la estructura de una base de datos: la información, las relaciones y las restricciones que deben cumplirse. Los modelos de datos contienen también un conjunto de operaciones básicas para la realización de consultas y actualizaciones de datos. (Pressman, 2010)

El modelo de la base de datos consta de tres fases: diseño conceptual, lógico y físico de la base de datos. La primera fase consiste en la producción de un esquema conceptual que es independiente de las consideraciones físicas. Luego el modelo se refina en un esquema lógico eliminando las construcciones que no se puede representar en el modelo de base de datos escogido. En la tercera fase, el esquema lógico se transforma en un esquema físico para el sistema gestor de base de datos elegido. La fase de diseño físico considera las estructuras de almacenamiento y los métodos de acceso necesarios para proporcionar un acceso eficiente a la base de datos. A continuación, se muestra la Figura 17 que representa el modelo físico de la propuesta de solución.

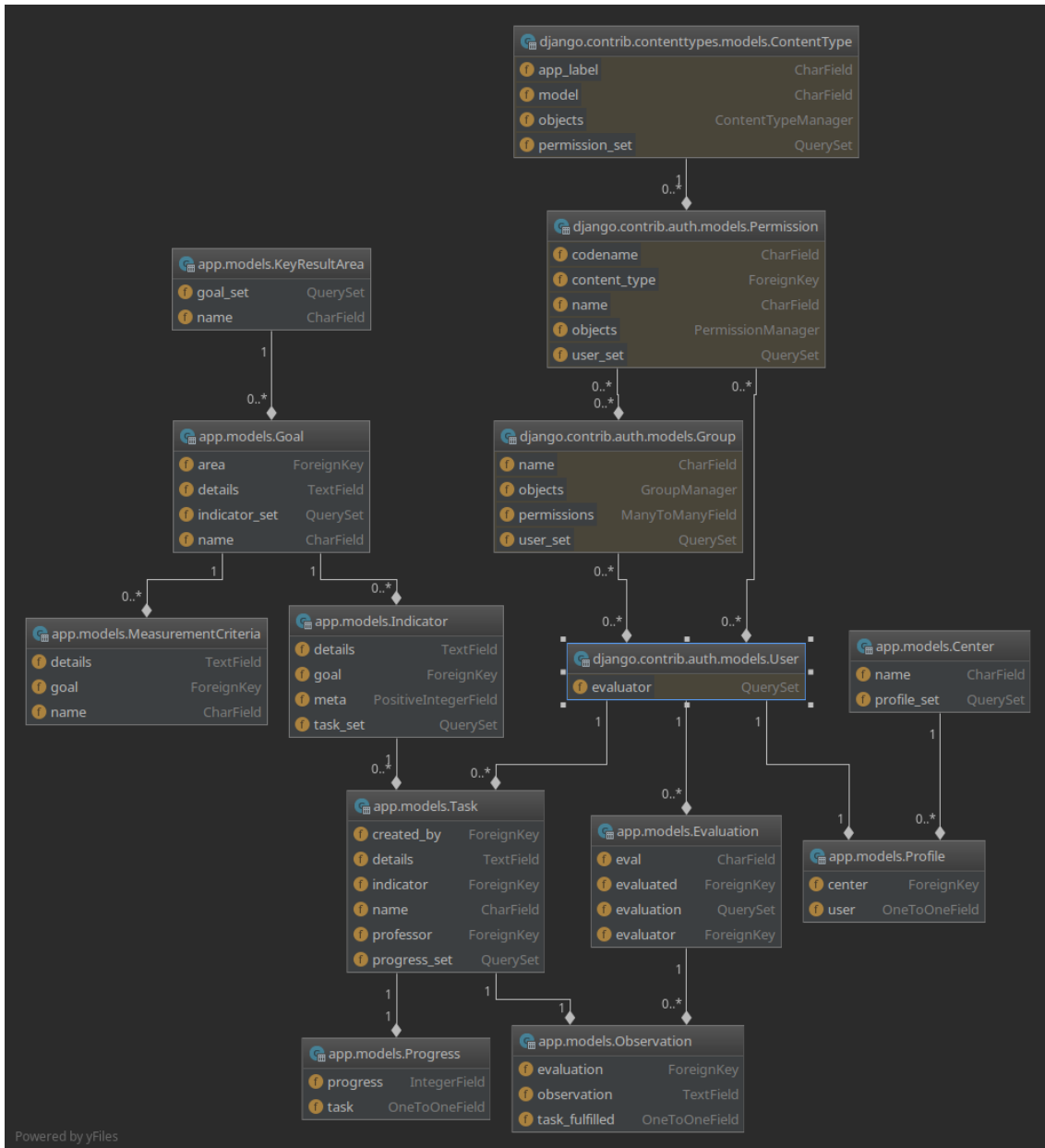


Figura. 17. Modelo de Datos. Fuente: elaboración propia.

2.2 Estándares de codificación.

Los estándares de codificación son reglas que se aplican para lograr uniformidad en el código producido por un grupo de desarrollo de un sistema. Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares

de codificación no destapan problemas existentes, evitan más bien que los errores ocurran, lo que permite obtener un código de alta calidad.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación, es de gran importancia para la calidad del software. La aplicación de estándares de codificación además posibilita que el software que se obtiene sea fácil de comprender y de mantener en el tiempo. El uso de los mismos tiene ventajas tales como:

- Asegurar la legibilidad del código entre distintos programadores, facilitando la depuración del mismo.
- Proveer una guía para el encargado de mantenimiento/actualización del sistema, con código claro y bien documentado.
- Facilitar la portabilidad entre plataformas y aplicaciones.

A continuación, se presentan algunos de los estándares de codificación utilizados en la solución propuesta. Se debe utilizar como idioma el español, las palabras no se acentuarán. 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. Además, antes de un comentario de bloque o de un comentario de una línea y entre las distintas secciones lógicas de un método para facilitar la lectura. Se debe dar un espacio en blanco entre una palabra clave del lenguaje y un paréntesis.

Respecto a las normas de inicialización, declaración y colocación de variables, constantes, clases y métodos:

- ✓ Todas las instancias y variables de clases o métodos empezarán con minúscula. Las palabras internas que lo forman, si son compuestas, empiezan con su primera letra en mayúsculas. Los nombres de variables no

deben empezar con los caracteres subguión "_" o signo de peso "\$", aunque ambos están permitidos por el lenguaje.

- ✓ Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúscula. Mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas.

Respecto a la indentación y longitud de la línea se deben emplear cuatro espacios como unidad de indentación. Los tabuladores deben ser exactamente cada 8 espacios. Y evitar las líneas de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.

Importaciones:

- Las importaciones deben estar en líneas separadas.
- Siempre deben colocarse al comienzo del archivo.
- Deben quedar agrupadas de la siguiente forma:
 - Importaciones de la librería estándar.
 - Importaciones terceras relacionadas.
 - Importaciones locales de la aplicación/librerías.
- Cada grupo de importaciones debe estar separado por una línea en blanco.

Un ejemplo de código que muestra la aplicación de este estándar de código se muestra en la figura 18.

Comentarios:

- Los comentarios deben ser oraciones completas.
- Si un comentario es una frase u oración, su primera palabra debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.
- Si un comentario es corto, el punto final puede omitirse.

```
20
21 class DefaultConnectionProxy:
22     """
23     Proxy for accessing the default DatabaseWrapper object's attributes. If you
24     need to access the DatabaseWrapper object itself, use
25     connections[DEFAULT_DB_ALIAS] instead.
26     """
27     def getattr (self, item):
```

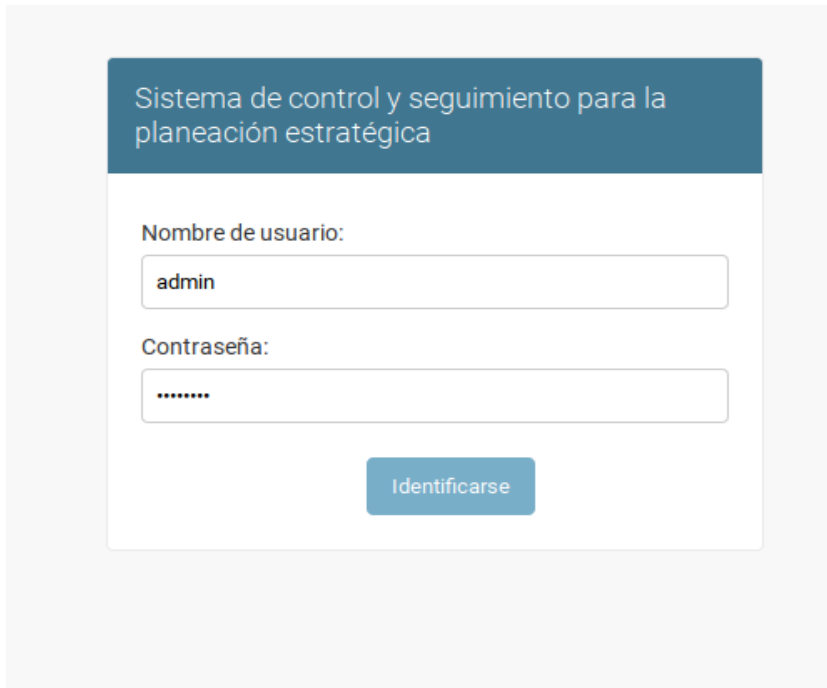
Figura. 18. Fragmento de código que muestra el uso de comentarios. Fuente: Elaboración propia

2.3 Normas de Seguridad informática a tener en cuenta

Todos los componentes de un sistema informático están expuestos a un ataque (hardware, software y datos) son los datos y la información los sujetos principales de protección de las técnicas de seguridad. La seguridad informática se dedica principalmente a proteger la confidencialidad, la integridad y disponibilidad de la información.

La seguridad es un tema de gran impacto en el sistema propuesto, pues es de vital importancia el control de la información que se almacena y se visualiza para garantizar la confidencialidad de la información registrada. Es por ello, que todo usuario que interactúe con la solución propuesta deberá autenticarse para realizar alguna acción sobre la misma. En este apartado se describen detalladamente las funcionalidades del Sistema de Planeación Estratégica:

- En el sistema se dan los permisos de acuerdo con la función que ocupa el usuario en el mismo, lo que permite solo tener acceso a las secciones, páginas, opciones del menú y funcionalidades que respondan directamente a su rol.
- El sistema brinda la posibilidad de asignar o denegar permiso a roles y usuarios en las funcionalidades.



The image shows a login form titled "Sistema de control y seguimiento para la planeación estratégica". It contains two input fields: "Nombre de usuario:" with the text "admin" and "Contraseña:" with masked characters "*****". Below the fields is a blue button labeled "Identificarse".

Figura 19. Interfaz de autenticación. Fuente: Elaboración propia.

2.4 Tratamiento de errores

Durante el tiempo de ejecución de un sistema pueden fracasar diferentes rutinas; es esto a lo que comúnmente se le llama excepción. Mediante el tratamiento de excepciones se restaura a un estado en el que la rutina pueda seguir la ejecución, lo que permite obtener un sistema robusto y fiable.

En el sistema propuesto, el control de las excepciones se lleva a cabo a toda porción de código donde pueda surgir alguna situación inesperada, especialmente donde se ejecutan sentencias que manipulan los datos que viajan desde y hacia la base de datos. También se controlan los errores en la validación de datos provenientes de la interfaz de usuario, puesto que encierran una lógica compleja en cierta medida.

Para el manejo de las excepciones o errores, en las clases controladoras de procesos, se utiliza el bloque try para detectar cuándo ocurra algún fallo y un bloque catch donde se manejarán dichas excepciones, mediante mensajes que se muestran en la interfaz de

usuario.

Existe un archivo denominado page.xml, que engloba la configuración de todos los mensajes que se deben mostrar por cada tipo de excepción, así como la página a la que el sistema redirecciona en caso de la aparición de un error sorpresivo.

2.5 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. La figura 20 muestra el diagrama de despliegue definido para la propuesta de solución:

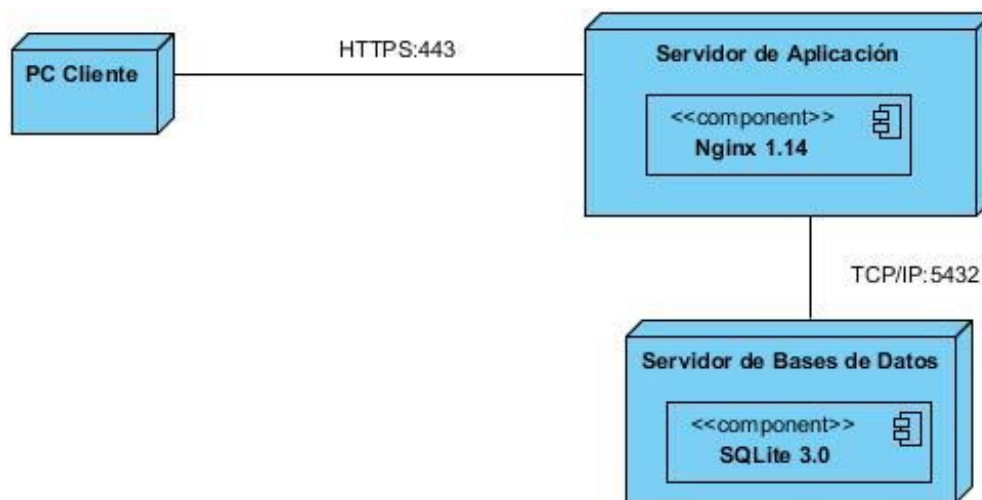


Figura 20. Diagrama de despliegue. Fuente: Elaboración propia.

- PC Cliente: ordenador desde donde los usuarios acceden a la aplicación.
- Servidor de aplicación: ordenador donde se encuentra el servidor web Nginx. Es de donde se gestiona todo el contenido de la aplicación. Las máquinas de los clientes acceden a él, a través del navegador web.
- Servidor de Bases de Datos: servidor donde se ubica la base de datos que almacena toda la información generada por el sistema y que va a ser utilizada por los servicios que brinda el módulo.

- HTTPS: Protocolo transferencia de hipertexto seguro, por sus siglas en inglés, Hypertext Transfer Secure Protocol (HTTPS) es un protocolo de aplicación basado en el protocolo HTTP que permite la transferencia de segura de datos de hipertexto siguiendo el esquema petición-respuesta entre un cliente y un servidor.
- TCP/IP: es una familia de protocolos de internet que permite la conexión entre el servidor web y el servidor de bases de datos.

2.6 Pruebas internas

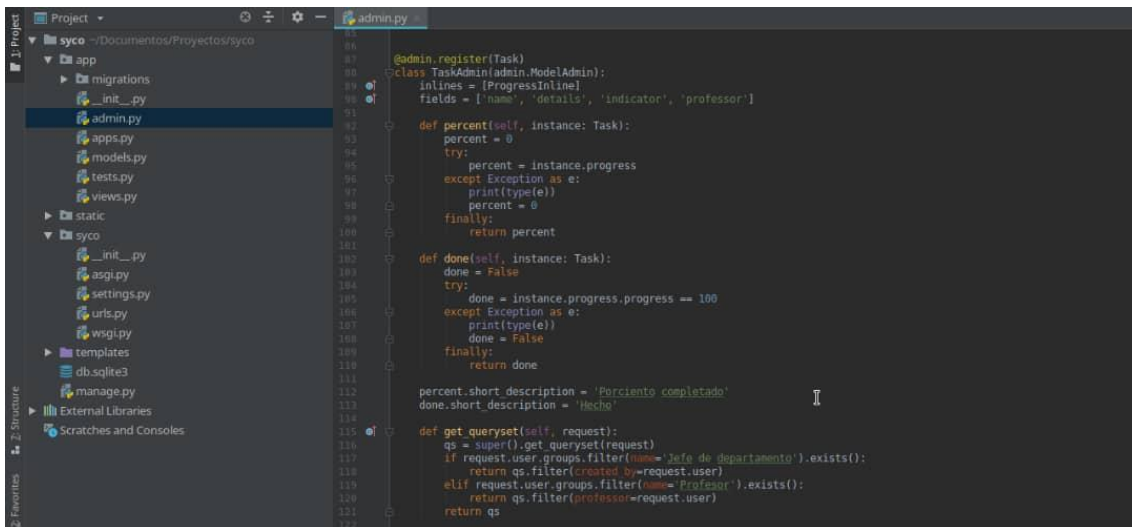
Durante esta etapa se prueban los componentes del producto con el objetivo de medir la calidad del software. El proceso de pruebas está encaminado a medir el cumplimiento de las funcionalidades establecidas por el cliente, reduciendo de esta manera el número de errores no detectados. Entre las pruebas internas se decide hacer pruebas unitarias, de caja blanca y de camino básico. También se emplea la herramienta *JUnit* que permite correr un conjunto de pruebas de forma automática e informa de aquellas que han fallado (Montes Ramírez et al. 2016).

Las pruebas unitarias o prueba de unidad enfocan los esfuerzos de verificación en la unidad más pequeña del diseño de software: el componente o módulo de software. Al usar la descripción del diseño de componente como guía, las rutas de control importantes se prueban para descubrir errores dentro de la frontera del módulo. La relativa complejidad de las pruebas y los errores que descubren están limitados por el ámbito restringido que se establece para la prueba de unidad. Las pruebas de unidad se enfocan en la lógica de procesamiento interno y de las estructuras de datos dentro de las fronteras de un componente. Este tipo de pruebas puede realizarse en paralelo para múltiples componentes. (Sommerville 2015)

Las pruebas realizadas mediante el método de caja blanca se basan en el examen cercano de los detalles de procedimiento. Las rutas lógicas en el software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. En ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. (Pressman 2010)

La prueba de camino básico permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. (Sommerville 2015) La complejidad ciclométrica se aplica a todos los métodos de la herramienta, pero en la investigación se analizará la funcionalidad de evaluación del profesor. Al calcular la complejidad ciclométrica a partir del grafo realizado del método en cuestión, se llegó a la conclusión de que se necesitan 3 casos de prueba para validar todas las posibles variantes del método. El algoritmo de evaluación del profesor posee poco riesgo debido a que el resultado arrojado por la métrica pertenece al intervalo entre 1-6.

Después de aplicar esta complejidad ciclométrica se procede a la realización de la técnica de prueba automatizada, pues esto facilita identificar funciones que no ofrecen una salida acorde con la lógica que se deseaba implementar. A continuación, la figura 21 representa un fragmento de código del sistema al cual se le realiza una prueba unitaria.



```

86
87
88 @admin.register(Task)
89 class TaskAdmin(admin.ModelAdmin):
90     inlines = [ProgressInline]
91     fields = ['name', 'details', 'indicator', 'professor']
92
93     def percent(self, instance: Task):
94         percent = 0
95         try:
96             percent = instance.progress
97         except Exception as e:
98             print(type(e))
99             percent = 0
100         finally:
101             return percent
102
103     def done(self, instance: Task):
104         done = False
105         try:
106             done = instance.progress.progress == 100
107         except Exception as e:
108             print(type(e))
109             done = False
110         finally:
111             return done
112
113     percent.short_description = 'Porcentaje completado'
114     done.short_description = 'Hecho'
115
116     def get_queryset(self, request):
117         qs = super().get_queryset(request)
118         if request.user.groups.filter(name='Jefe de departamento').exists():
119             return qs.filter(created_by=request.user)
120         elif request.user.groups.filter(name='Profesor').exists():
121             return qs.filter(professor=request.user)
122         return qs
  
```

Figura 21. Fragmento de código de la prueba realizada en *JUnit*. Fuente: Elaboración propia

En el caso de la prueba que se muestra en la Figura 20, no se detectaron errores. La prueba se realiza en 2.121 segundos y el método *testConstruirConsultas* 2.468 segundos.

Una vez realizada la corrección de los errores detectados, los métodos de cada clase se ejecutaron de forma correcta, obteniendo de ellos los resultados esperados. Se tomaron precauciones para evitar errores similares y se definieron nuevos casos de prueba para

verificar el correcto desempeño de las funcionalidades.

También se emplearon pruebas de caja negra para probar las funcionalidades tomando como unidad a cada interfaz. Aquí no se centra en cómo se generan las respuestas del sistema, solo se analizan los datos de entrada y los resultados obtenidos.

Este método contiene la técnica de partición de equivalencia que consiste en clasificar las entradas de datos del sistema en grupos que presentan un comportamiento similar, por lo cual serán procesados de la misma forma. Se pueden definir particiones tanto para valores válidos como inválidos.

A continuación, se muestran los casos de pruebas de caja negra aplicado al CU correspondiente a la evaluación del profesor.

Tabla 3. Caso de prueba de caja negra “Gestionar Plan de Resultados del profesor”. Fuente: Elaboración propia

Escenario	Variables				Descripción	Respuesta del sistema	Flujo Central
	1	2	3	4			
Sección 1: Asignar cumplimiento de un indicador al profesor							
EC 1.1 Asignar cumplimiento de un indicador al profesor satisfactoriamente	V	V	V	V	El analista introduce todos los datos correctamente, permitiendo el sistema la asignar cumplimiento del indicador.	Se muestra un mensaje de información “El profesor tiene asignado el cumplimiento de un indicador”.	1.-Vincular profesor con indicador, seleccionar % del indicador, fecha de cumplimiento. 2.- Presionar el botón “Aceptar”.
EC 1.2 Datos incorrectos.	I	I	I	I	El analista introduce uno o varios datos	Muestra un mensaje de error: “Los	1.- Se introduce uno
	V	I	V	I			
	I	V	I	I			

	V	V	I	I	incorrectos, el sistema no permitirá asignar indicador al profesor.	datos introducidos son incorrectos".	o varios datos incorrectos. 2.- Presionar el botón "Aceptar".
	I	I	V	I			
	V	I	V	I			
	I	V	V	I			
	V	V	V	I			
	I	I	I	V			
	V	I	I	V			
	I	V	I	V			
	V	V	I	V			
	I	I	V	V			
	V	I	V	V			
	I	V	V	V			
	V	V	V	V			
EC 1.3 Campos vacíos.	I	I	I	I	El analista deja uno o varios campos vacíos, el sistema no permitirá asignar indicador al profesor	Muestra un mensaje de error: "Existen campos obligatorios vacíos. Por favor, complete estos campos".	1.- Se dejan uno o varios campos vacíos. 2.- Presionar el botón "Aceptar".
	V	I	V	I			
	I	V	I	I			
	V	V	I	I			
	I	I	V	I			
	V	I	V	I			
	I	V	V	I			
	V	V	V	I			
	I	I	I	V			
	V	I	I	V			
	I	V	I	V			
	V	V	I	V			
	I	I	V	V			
	V	I	V	V			
	I	V	V	V			
	V	V	V	V			
Sección 2: Modificar cumplimiento de un indicador al profesor							

<p>EC 2.1 Modificar cumplimiento de indicador</p>	V	V	V	V	<p>El analista introduce todos los datos correctamente, permitiendo el sistema la modificar cumplimiento del indicador.</p>	<p>Se muestra un mensaje de información “El indicador asociado al profesor ha sido actualizado correctamente”.</p>	<p>1.- Seleccionar % del indicador, fecha de cumplimiento y parámetros de evaluación. 2.- Presionar el botón “Aceptar”.</p>
<p>EC 2.2 Datos incorrectos.</p>	I	I	I	I	<p>El analista introduce uno o varios datos incorrectos, el sistema no permitirá modificar el indicador asociado al profesor.</p>	<p>Muestra un mensaje de error: “Los datos introducidos son incorrectos”.</p>	<p>1.- Se introduce uno o varios datos incorrectos. 2.- Presionar el botón “Aceptar”.</p>
V	I	V	I				
I	V	I	I				
V	V	I	I				
I	I	V	I				
V	I	V	I				
I	V	V	I				
V	V	V	I				
I	I	I	V				
V	I	I	V				
I	V	I	V				
V	V	I	V				
I	I	V	V				
V	I	V	V				
I	V	V	V				
V	V	V	V				
<p>EC 2.3 Campos vacíos.</p>	I	I	I	I	<p>El analista deja uno o varios campos vacíos, el</p>	<p>Muestra un mensaje de error: “Existen campos</p>	<p>1.- Se dejan uno o varios campos vacíos.</p>
V	I	V	I				
I	V	I	I				
V	V	I	I				

	I	I	V	I	sistema no permitirá modificar indicador asociado al profesor	obligatorios vacíos. Por favor, complete estos campos".	2.- Presionar el botón "Aceptar".
	V	I	V	I			
	I	V	V	I			
	V	V	V	I			
	I	I	I	V			
	V	I	I	V			
	I	V	I	V			
	V	V	I	V			
	I	I	V	V			
	V	I	V	V			
	I	V	V	V			
	V	V	V	V			
EC 3.1 Desvincular un indicador del profesor	El analista selecciona un indicador para desvincularlo del profesor				Muestra un mensaje de confirmación "¿Seguro que desea desvincular el indicador seleccionado?"	1.- Presionar el botón "Aceptar".	

Se ejecutarán 2 casos de pruebas, para el primer caso de prueba se diseñaron un total de 7 escenarios como muestra la Tabla 4. El resultado de estas pruebas se evidenciará en la Tabla 3. Al concluir la solución de las NC de cada iteración, se realizan pruebas de regresión.

Este proceso de prueba permitirá verificar el cumplimiento de los requisitos funcionales del software, donde los resultados de las pruebas que no fueron satisfactorios pasaron a ser no conformidades.

Tabla 4. Resultados de las pruebas de caja negra. Elaboración propia

No. Iteración	NC detectadas	Funcionalidad	Interfaz	Correspondencia	Resultados
1ra Iteración					
2da Iteración					
3ra Iteración					
Iteración n					

2.7 Conclusiones parciales

Los estándares de codificación, el tratamiento de errores y los elementos de seguridad informática propiciaron la aplicación de buenas prácticas en el desarrollo de software y la obtención de un producto con calidad.

Las pruebas realizadas tanto al código como a las funcionalidades validan que el sistema funciona correctamente, determinado por los resultados obtenidos en su ejecución.

Conclusiones

Una vez concluida la presente investigación se puede afirmar que se desarrolló satisfactoriamente el Sistema para el seguimiento y control de la Planeación estratégica de la facultad 1. Durante su realización se arribaron a las siguientes conclusiones:

La elaboración del marco teórico metodológico permitió crear las bases del desarrollo de la investigación y constatar su novedad de la investigación.

Se determinan aspectos importantes que se consideran en el desarrollo de la aplicación a partir del estudio de herramientas que existen y cuyo objetivo es el control, seguimiento y planeación estratégica.

Las herramientas, tecnologías y lenguajes de programación seleccionados permitieron obtener una solución robusta, con estándares de codificación y buenas prácticas de desarrollo, las cuales propician la extensibilidad de la aplicación.

El análisis y diseño aplicando la metodología definida permitieron identificar los requisitos funcionales y no funcionales, así como los artefactos ingenieriles que respaldan la solución propuesta, lo cual permitió cumplir con las expectativas del usuario.

El sistema obtenido propicia el control y seguimiento de la planeación estratégica de la facultad 1, lo cual aporta una solución que mejora los procedimientos de ejecución actual para efectuar estos procesos dentro de la facultad.

La solución es válida dictaminado por los resultados de la aplicación de pruebas desarrolladas, la solución a las no conformidades detectadas y las recomendaciones realizadas durante el proceso.

Recomendaciones

Para futuras investigaciones se recomienda:

Incorporar un componente para el análisis de los resultados del cumplimiento de los indicadores en el año.

Desarrollar un módulo de análisis para comparar el desempeño de la facultad en el año en curso con respecto a períodos anteriores, que propicie la toma de decisiones para el futuro.

Referencias Bibliográficas

Acosta, L. A., Becerra, F. A., & Jaramillo, D. (2017). Sistema de Información Estratégica para la Gestión Universitaria en la Universidad de Otavalo (Ecuador). *Formación universitaria*, 10(2), 103-112.

Adaptive Insights. (2019). Solución de planificación empresarial para elaboración de presupuestos, pronósticos e informes. Acceso [04/03/2020] Disponible en: <https://www.getapp.com.mx/software/90888/adaptive-planning>

Araya L. A. (2017). Modelos de planeación estratégica en las empresas familiares. *Tec Empresarial*, 11(1), 23-34.

Canós, José H., and M^a Carmen Penadés Patricio Letelier. "Metodologías ágiles en el desarrollo de software." (2012).

Chiavenato, I., & Sapiro, A. (2017). *Planeación estratégica*. McGraw-Hill Interamericana.

Fernández, D., Juvinao, D., and Solano E. (2016). "Planificación estratégica de recursos humanos: efectiva forma de identificar necesidades de personal." *Económicas CUC* 37, no. 1: 63-80.

Garay, L. W. P., & Soto, D. M. (2016). Guía para el desarrollo de almacenes de datos. *Iberoamerican Journal of Project Management*, 7(2), 18-42.

Gómez, O. T., López, P. P. R., & Bacalla, J. S. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial data*, 13(2), 70-74.

Ojeda, J. C., & Fuentes, M. D. C. G. (2012). Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. *Universidades*, (52), 37-47.

Hernández, L. A. Á., Quintero, D. N., Montoya, A. V., & Berrio, A. G. (2018). Modelos de gestión organizacional y planeación estratégica de las clínicas odontológicas especializadas ODONTOSTUDIO limitada. *Revista GEON (Gestión, Organizaciones y Negocios)*, 5(2), 38-49.

Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right*. Apress.

ISOTools (2018). Herramienta online para la creación de Cuadros de Mando Integral.

Acceso [04/03/2020] Disponible en: <https://www.isotoools.org/software/estrategia-balanced-scorecard/>

Jones, C., Motta, J., & Alderete, M. V. (2016). Gestión estratégica de tecnologías de información y comunicación y adopción del comercio electrónico en Mipymes de Córdoba, Argentina. *Estudios gerenciales*, 32(138), 4-13.

KPI Fire. (2019). software integral de planificación y ejecución de estrategias para gestión de proyectos, creación de tableros de KPI. Acceso [04/03/2020] Disponible en: <https://www.getapp.com.mx/software/113385/kpi-fire>.

Lutz, M. (2001). Programación de Python . "O'Reilly Media, Inc."

Ministerio de Educación Superior (2017). Planificación Estratégica 2017-2021. Acceso [04/03/2020] Disponible en: [https://www.mes.gob.cu/sites/default/files/documentos/PLANIFICACION%20ESTRATEGICA%20\(Final\)%202017-2021_4.pdf](https://www.mes.gob.cu/sites/default/files/documentos/PLANIFICACION%20ESTRATEGICA%20(Final)%202017-2021_4.pdf)

Ministerio de Educación Superior (2018). Objetivos de Trabajo del Ministerio de Educación Superior para el año 2019. Acceso [09/03/2020] Disponible en: [https://www.mes.gob.cu/sites/default/files/documentos/Objetivos%20MES%202019%20%20Revisado%20\(07-12-18\)_4.pdf](https://www.mes.gob.cu/sites/default/files/documentos/Objetivos%20MES%202019%20%20Revisado%20(07-12-18)_4.pdf)

Novoa, V. D., & Iglesias, V. G. (2016). La competitividad en las instituciones de Educación Superior: aplicación de filosofías de gestión empresarial. *Gestión del proceso de negocio. Economía y Desarrollo*, 157(2), 166-181.

Ortiz, J. B. R. (2018). Capital Humano: Un desglose teórico para su operatividad en organizaciones inteligentes. *CICAG: Revista del Centro de Investigación de Ciencias Administrativas y Gerenciales*, 16(1), 43-54.

Ortiz Pérez, A. (2014). Tecnología para la gestión integrada de los procesos en universidades. Aplicación en la universidad de Holguín. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Universidad de Holguín. 2-4 julio.

Pacheco Rodríguez, R., Alonso Rodríguez, S. H., & Mena Lorenzo, J. A. (2018). Génesis y evolución del Sistema de Trabajo Metodológico como parte del Sistema de Trabajo en Cuba. *Mendive. Revista de Educación*, 16(2), 314-329.

Peña, Ayala A. 2016. Ingeniería de Software: Una Guía para Crear Sistemas de Información. México : s.n., 2016.

Parra Nieto, C. F., & Davila Lancheros, M. L. (2016). Técnicas utilizadas en la gestión de integración en la planeación y seguimiento de proyectos en las organizaciones de Cali-Colombia.

PRESSMAN, R.S., 2010. Ingeniería del software, un enfoque práctico. 7ma edición. Madrid, España: s.n.

QuickScore (2019). Solución de gestión de rendimiento estratégico. Acceso [09/03/2020] Disponible en: <https://www.getapp.com.mx/software/129567/quickscore-balanced-scorecards>

Ramírez, J. L. (2017). Procedimiento para la elaboración de un análisis FODA como una herramienta de planeación estratégica en las empresas.

Romero Fernández, Y., & González Díaz, Y. (2014). Patrón modelo-vista-controlador. Revista Telem@tica, 47–57.

Santos, A. "Tecnología de gestión de recursos humanos." Anales de la Academia de Ciencias de Cuba 9, no. 3 (2019).

SELLARÈS, T., 2011. The Model View Controller: a Composed Pattern. Universitat de Girona [en línea], Disponible en: <http://ima.udg.edu/~sellares/EINF-ES1/MVC-Toni.pdf>.

Sommerville, Ian. 2005. Ingeniería de Software Séptima Edición. Madrid: Pearson Educación S.A, 2005. ISBN: 84-7829-074-5.

Tabares, R. 2014. Patrones Grasp y Anti-Patrones: un Enfoque Orientado a Objetos desde Lógica de Programación. s.l. : Entre Ciencia e Ingeniería, 2014.

Valdés, A., Cabral, A., Alvarado, T, & Rubio, R.D. (2017). "Una nueva metodología en planeación estratégica como un recurso básico en la ganadería lechera. Versión 2017." Revista Mexicana de Agronegocios 40, no. 1345-2017-1406 pp. 541-550.

Vargas, Adela Ascencio, Jhon Campo Ramírez, Andrés Ramírez Roldán, and Liliana Zapata Cardona. (2016) "Importancia de la planeación estratégica en las áreas de gestión humana de las organizaciones." Revista Fundación Universitaria Luis Amigó (histórico) 3, no. 1: 116-122.

VISCONTI, M. y ASTUDILLO, 2015. Fundamentos de Ingeniería de Software: Patrones de Diseño. Universidad Técnica Federico Santa María,

Zaldívar, R. V., Campaña, M. P., & Pérez, A. O. (2017). Planificación estratégica y gestión de la calidad: caso de una institución de educación superior cubana. Revista Cubana de Administración Pública y Empresarial, 1(1).

Anexos

Anexo 1. Entrevista realizada para el levantamiento de requisitos.

Persona Entrevistado(a): Yanet Espinal Martín.

Cargo que ocupa: Jefa de Dpto. (Dpto. de Informática)

Objetivo de a entrevista: Describir el proceso de negocio y las actividades asociadas a la Planeación Estratégica para su posterior análisis e implementación.

Preguntas:

1. ¿Qué son los objetivos estratégicos?
2. ¿Cómo se evalúa el cumplimiento de los objetivos estratégicos?
3. ¿Por qué están compuestos estos objetivos estratégicos?
4. ¿A quiénes le son asignadas las tareas?
5. ¿Quién asigna las tareas?
6. ¿Qué es un plan de resultado?
7. ¿Cómo se elabora un plan de resultados?
8. ¿Qué son las Áreas de Resultados Clave (ARC)?
9. ¿Qué son los indicadores para los planes de resultados clave?
10. ¿Qué son los indicadores?