



FACULTAD 1

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

Título:

**SISTEMA PARA EL DIAGNÓSTICO ESTRATÉGICO EN EL
DESARROLLO DE PROYECTOS PRODUCTIVOS EN LA
UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

Autor:

Raydel Carmenate Legrá

Tutores

MSc. Yanisel Guillard Vara

Ing. Aneyty Martín García

La Habana, junio 2016

Declaración de Autoría

Declaro ser el único 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 _____.

Firma del Autor

Raydel Carmentate Legrá

Firma del Tutor

MSc. Yanisel Guillard Vara

Firma del Tutor

Ing. Aneyty Martín García

AGRADECIMIENTOS

Le agradezco a mis padres por todo el amor y dedicación en todo momento.

A mi tutoras por su comprensión y por el apoyo brindado

A mi hermana por su ayuda en los momentos más difíciles

A mi amigo Andy y a mis compañeros de apartamento.

DEDICATORIA

A mi madre por su apoyo, comprensión, y resistencia en los momentos más difíciles.

A mi padre por todo su apoyo y dedicación.

RESUMEN

La presente investigación consistió en desarrollar una aplicación de escritorio para el desarrollo de diagnóstico estratégico para los proyectos productivos de la Universidad de las Ciencias Informáticas. La herramienta propuesta posee un conjunto de características de herramientas similares analizadas en el estudio de homólogos y varias funcionalidades que no se encuentran en herramientas similares estudiadas. La propuesta de solución estuvo guiada por la metodología XP, se seleccionó para la implementación NetBeans IDE 8.0 como entorno integrado, con un lenguaje Java. Se eligió como sistema gestor de base de datos PostgreSQL. Las pruebas funcionales arrojaron como resultado que la aplicación es funcional, permitiendo validar que la propuesta de solución muestra la matriz DAFO con el cuadrante de mayor impacto en el proyecto productivo donde se desarrolle el diagnóstico estratégico.

Palabras clave: Diagnóstico estratégico, factores del entorno, proyectos productivos.

ÍNDICE

INTRODUCCIÓN	6
CAPÍTULO I. DISEÑO TEÓRICO-METODOLÓGICO PARA REALIZAR DIAGNÓSTICO ESTRATÉGICO EN PROYECTOS PRODUCTIVOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.	12
1.1 Introducción	12
1.2 Conceptos fundamentales:	12
1.3 Proceso de diagnóstico estratégico	18
1.4 Formulación de estrategias.....	24
1.5 Modelos de planeación estratégica	26
3.5 Pruebas.	62
3.5.1 Pruebas de Unidad.....	63
3.5.2 Pruebas de aceptación.....	65
3.6 Conclusiones del Capítulo.	68
CONCLUSIONES	69
RECOMENDACIONES	70
Bibliografía	71
GLOSARIO DE TÉRMINOS	73
ANEXOS	74

INTRODUCCIÓN

En agosto del año 2009, el Presidente del Consejo de Estado y de Ministros, Raúl Castro Ruz subrayó la necesidad de actualizar el modelo económico cubano. A raíz de esto, en abril del 2010 se realizó el VI Congreso del Partido, donde se expusieron las principales directrices a seguir para el desarrollo de la actualización del modelo económico cubano, los “Lineamientos de la Política Económica y Social del Partido y la Revolución”. Desde esa fecha, el país se encuentra inmerso en el proceso de actualización de estas políticas, cuyo propósito fundamental es ajustar el modo de organización y funcionamiento de la economía nacional a las circunstancias actuales como fundamento de la sociedad a la que aspira Cuba.

El proceso de actualización del modelo económico cubano, en relación al programa de informatización, subraya la necesidad de desarrollar las Tecnologías de la Información y las Comunicaciones (TIC), en beneficio de la economía y la sociedad, y convertirlas en un sector de desarrollo estratégico para Cuba. Este programa demuestra la disposición del país de lograr un mayor acercamiento de la población al desarrollo informático, mediante el aprovechamiento de las TIC como herramientas del conocimiento, la economía y la actividad político-ideológica. En este sentido, se han desarrollado algunas acciones con el objetivo de alcanzar un mayor desarrollo de la industria cubana del software, la cual exhibe algunas insuficiencias y debilidades, y exige que se enfrenten mayores desafíos.

En la actual situación, el desarrollo de la industria cubana de software presenta una serie de riesgos y restricciones importantes que dificultan su progreso. Existe una creciente preocupación acerca de las insuficiencias en la contextualización del entorno para el progreso de dicha industria, así como en la determinación en forma correcta de la posición estratégica en la que se encuentra. Por otra parte, se reflejan limitaciones en el grado de participación de los diferentes actores en el proceso de informatización de la sociedad. De igual modo, se ha manifestado incertidumbre en la valoración de los criterios que contribuyen al proceso de toma de decisiones, para responder de manera efectiva a las necesidades expuestas en la actualización del modelo económico.

Para lograr el despegue definitivo de la industria cubana del software, se han señalado entre otros aspectos, la necesidad del papel protagónico y coordinación de los principales actores en el proceso, la definición de un sistema de indicadores que permita conducir y evaluar los avances e impactos del proceso de informatización de la sociedad, etc. Constituyen otras de las exigencias actuales, establecer programas para la introducción del desarrollo informático en los diferentes sectores económicos estratégicos del país, alcanzar óptimos niveles de eficiencia en la exportación de servicios y productos informáticos de alto contenido tecnológico (ALTEC), y el establecimiento de modelos de negocio entre operadores y proveedores de telecomunicaciones. Se destaca, además, la necesidad de la integración de la industria nacional de software al desarrollo en América Latina y el Caribe, y la participación en alianzas y contratos a nivel regional. Alcanzar esos retos favorecerá el fortalecimiento de la industria nacional de software, que tiene un potencial inmenso en el capital humano formado por la Revolución.

El proceso de formulación de la estrategia determina el rumbo y la posición competitiva de una organización durante mucho tiempo. Su importancia radica en permitir solucionar correctamente las distintas situaciones que se puedan presentar y establecer planes de acción alternativos, lo que conducirá a un proceso de toma de decisiones oportuno y eficaz. La realización del proceso de diagnóstico estratégico permite conocer cuáles son las fuerzas actuantes del entorno en los proyectos productivos de la Universidad de las Ciencias Informáticas, lo cual posibilita que se tomen decisiones correctas. El diagnóstico estratégico facilita la determinación de las ventajas competitivas de la organización y la estrategia genérica a emplear por la misma. De lo anterior que se señale, el proceso de formulación de estrategias posibilita a la industria cubana de software mayor capacidad de respuesta ante la influencia del entorno, y la obtención de ventajas competitivas basadas en la reducción de los precios y el margen de beneficios.

Actualmente, a nivel internacional existen algunas herramientas informáticas que permiten realizar el proceso de diagnóstico estratégico en forma semi-automatizada. Estas herramientas presentan funcionalidades que permiten la formulación y evaluación de estrategias. En contraste con lo anterior, las mismas presentan algunas limitaciones, relacionadas con la adaptación de las variables y criterios de medida en el desarrollo de la industria nacional de software. Esas limitaciones se evidencian a partir de lo siguiente:

1. No permiten la comparación entre los diferentes indicadores empresariales con los indicadores a nivel de sector o de región.
2. No posibilitan adecuar los indicadores y políticas de desarrollo a las condiciones propias de la región en que se encuentra la organización que se evalúa.
3. Se encuentran exclusivamente disponibles online por lo que no se pueden guardar en una base de datos local.
4. Presentan dificultad para adaptarse a las políticas de migración a software libre en el país.

Por lo antes mencionado se plantea como **problema de investigación**: ¿Cómo realizar una herramienta informática para automatizar el proceso de diagnóstico estratégico adaptado a las actuales necesidades de los proyectos productivos de la Universidad de las Ciencias Informáticas?

Enmarcado en el **objeto de estudio** el proceso de diagnóstico estratégico y como **campo de acción** el proceso de diagnóstico estratégico en los proyectos productivos de la Universidad de las Ciencias Informáticas.

Se plantea como **objetivo general** del presente trabajo desarrollar una aplicación informática que permita automatizar el proceso de diagnóstico estratégico de los proyectos productivos de la Universidad de las Ciencias Informáticas.

A partir del objetivo general planteado se desglosaron varios **objetivos específicos**:

- Fundamentar el marco teórico referencial relativo al proceso de diagnóstico estratégico.
- Implementar las funcionalidades de la aplicación de la herramienta de diagnóstico estratégico adaptado a las necesidades actuales de los proyectos productivos de la Universidad de las Ciencias Informáticas.
- Validar las funcionalidades de la aplicación para realizar diagnóstico estratégico mediante la realización de pruebas de software.

Como **idea a defender** se plantea que el desarrollo de una herramienta informática podría contribuir a la automatización del proceso de diagnóstico estratégico en los proyectos productivos de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento a los objetivos específicos se establecen las siguientes **tareas de investigación**:

- Análisis del marco teórico referencial asociado al proceso de diagnóstico estratégico.
- Análisis y selección de las metodologías para la realización del diagnóstico estratégico.
- Definición de la metodología a utilizar en el desarrollo de la aplicación para realizar diagnóstico estratégico en los proyectos productivos de la Universidad de las Ciencias Informáticas.
- Selección de las herramientas y tecnologías a utilizar en el desarrollo de la aplicación para realizar diagnóstico estratégico en los proyectos productivos de la Universidad de las Ciencias Informáticas
- Descripción del modelo del negocio a utilizar para la realización de la aplicación.
- Obtención y especificación de los requisitos funcionales y no funcionales de la aplicación.
- Modelado y diseño de los artefactos de ingeniería de software propuestos por la metodología de desarrollo.
- Implementación de las funcionalidades de la aplicación para realizar diagnóstico estratégico.
- Definición de las estrategias y técnicas de prueba a realizar al software para garantizar su calidad.
- Validación de la propuesta de solución mediante la realización de pruebas de software a la aplicación.

Métodos científicos de investigación:

Métodos teóricos

Analítico – Sintético: Se utilizó para analizar el marco teórico referente al diagnóstico estratégico de organizaciones para obtener los conocimientos necesarios sobre este tema.

Histórico – Lógico: Se utilizó para describir el proceso de diagnóstico estratégico en la investigación asumida.

Métodos empíricos:

Observación: Se utilizó en el estudio de los sistemas homólogos, para determinar las características de estos sistemas que puedan emplearse en la solución a desarrollar.

Entrevistas: Se desarrolló para entrevistar al cliente y conocer los elementos fundamentales sobre el diagnóstico estratégico y así determinar las funcionalidades a implementar.

Estructura del documento

- **CAPÍTULO 1. DISEÑO TEÓRICO-METODOLÓGICO PARA REALIZAR DIAGNÓSTICO ESTRATÉGICO EN PROYECTOS PRODUCTIVOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.**

Contiene los fundamentos teóricos del tema que se trata, se exponen los conceptos relacionados con el proceso de diagnóstico estratégico en las organizaciones, se analizan las principales metodologías existentes para la realización de dicho proceso y se realiza el estudio de los diferentes sistemas homólogos que existen. Se argumenta el uso de las herramientas y tecnologías escogidas para la realización de la investigación.

- **CAPÍTULO 2. CARACTERIZACIÓN DEL SISTEMA DE DIAGNÓSTICO ESTRATÉGICO.**

Se identifican los requisitos funcionales y no funcionales del sistema, se define el diseño de la aplicación utilizando los distintos artefactos según la metodología de

desarrollo. Se exponen los patrones de diseños y estilos arquitectónicos utilizados para la realización del sistema y se realiza la implementación de las funcionalidades del sistema.

- **CAPITULO 3. VALIDACIÓN DEL SISTEMA DE DIAGNÓSTICO ESTRATÉGICO.**

Se exponen los resultados de la evaluación de la solución mediante las diferentes pruebas realizadas y se realiza la validación de la solución.

CAPÍTULO I. DISEÑO TEÓRICO-METODOLÓGICO PARA REALIZAR DIAGNÓSTICO ESTRATÉGICO EN PROYECTOS PRODUCTIVOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.

1.1 Introducción

En el presente capítulo se define el marco conceptual que sustenta la investigación estrechamente vinculada a la herramienta de diagnóstico estratégico para los proyectos productivos de la Universidad de las Ciencias Informáticas. Se exponen los principales conceptos relacionados con el estado del arte de las herramientas que permiten realizar diagnóstico estratégico online. A partir de los resultados se analizan diferentes lenguajes y herramientas que sean factibles para desarrollar una solución a los inconvenientes planteados, así como la metodología para guiar el proceso de desarrollo de software.

1.2 Conceptos fundamentales:

Planeación Estratégica:

La planeación es el proceso de establecer las metas, los métodos a utilizar y las acciones a seguir para el logro de un determinado resultado. Su propósito fundamental consiste en coordinar los esfuerzos y los recursos dentro de las organizaciones, para garantizar el logro de los objetivos empresariales, reduciendo la incertidumbre y definiendo las consecuencias de una acción administrativa determinada.

“Dentro del proceso administrativo, la planeación constituye el punto inicial de la acción directiva, ya que supone un establecimiento de objetivos y los cursos de acción para lograrlos. Este logro depende en gran medida del grado de conocimiento de las variables del entorno organizacional” (Scott, 2011).

La planificación permite enfocar los esfuerzos y provee un marco en el cual los integrantes de la empresa podrán desarrollar sus actividades más eficazmente. La misma comprende el análisis de la situación, el establecimiento de objetivos, la formulación de

estrategias que permitan alcanzar dichos objetivos, y el desarrollo de planes de acción que señalen cómo implementar dichas estrategias.

“Planificar es, por lo tanto, un proceso mediante el cual los actores involucrados estudian el entorno interno y externo de la organización para establecer cuál es su aquí y ahora, y acto seguido reflexionan acerca de a dónde desean llegar y cuando debe ser ese arribo. Establecidos estos dos importantes referentes, se trazan entonces las estrategias que deben llevar a la organización de un punto hasta el otro, de la forma más eficiente y eficaz posible, siempre bajo los principios filosóficos establecidos y en concordancia con las especificaciones del sistema social, tecnológico, administrativo, financiero, productivo y cultural” (Scott, 2011).

Como característica de la planificación, Cámara et al. (2005) afirman que la planificación debe responder a la idea de sistema, estableciendo relaciones de interdependencia entre los distintos instrumentos, áreas de trabajo de la organización, resultados y objetivos, y contribuyendo a la coherencia y coordinación de todo el conjunto. Desde esta perspectiva los autores señalan que planificar también consiste en pensar acerca de estrategias en forma racionalizada, descompuesta, articulada en una serie de pasos. Además, expresan que la planificación debe concebirse como un medio para la consecución de un fin, objetivo o propósito, cualquiera que éste sea. Lo cual obliga a clarificar desde el inicio cuál es el propósito que se pretende lograr; ya que si no se conoce y definen con precisión adonde se quiere llegar, cualquier ejercicio de planificación carece de sentido (Cámara et al., 2005).

La planificación requiere de un proceso de toma de decisiones cuyos resultados implicarán una adecuación reguladora de las acciones operativas a aplicar para la consecución de las metas previstas. Esta visión integradora hace que la planificación sea entendida como un sistema compuesto por procesos capaces de diseñar, regular y evaluar la eficacia del mismo (Scott, 2011).

En ese sentido, se destaca (Cámara et al., 2005) la planificación no debe quedarse en un mero ejercicio analítico, intelectual o de diagnóstico de la realidad, sino que debe tener un carácter eminentemente práctico, orientado a la acción y, en última instancia, a la transformación de la realidad en un futuro más o menos inmediato. Considerando lo

anterior, diferentes autores asumen que la planificación también debe responder a criterios de racionalidad y transparencia, permitiendo así fundamentar las acciones y explicar a los demás cómo y por qué se toman las decisiones, sometiendo a éstas y a los motivos que las impulsan al escrutinio y crítica de externos (Cámara et al., 2005).

Por otro lado, las modernas concepciones de la planificación, aun reconociendo la necesidad insoslayable de que este proceso sea impulsado y apoyado desde los niveles directivos, destacan la importancia de la participación e implicación de los distintos miembros y niveles de la organización, como un requisito esencial del proceso de planificación, para que la implantación de los cambios sea posible y resulte exitosa (Cámara et al., 2005). Si bien es cierto, que en toda organización existe una división de tareas y funciones, y que probablemente habrá equipos o grupos de trabajo especializados en las labores de planificación, ésta, para ser eficaz, debe involucrar a toda la organización (Cámara et al., 2005).

Como otra de las características de la planificación, se afirma (Cámara et al., 2005) la misma responde al intento de reducir la incertidumbre, tratando de hacer un ejercicio de prospectiva para anticipar los cambios del entorno en el que las organizaciones operan, y adaptar permanentemente los planes de acción a esos cambios. En relación con esta idea, destacan que mediante la planificación se trata de reducir o acotar la incertidumbre, pero en ningún modo se puede eliminar por completo.

La planificación no finaliza con el diseño de un plan concreto, sino que es un proceso dinámico, y por tanto, debe estar sujeto a un continuo reajuste y revisión de lo planificado para adaptarlo constantemente a las circunstancias, así como de evaluación de sus resultados, con el fin de extraer lecciones de la experiencia, y generar procesos de aprendizaje para la organización (Cámara et al., 2005).

La planeación estratégica es definida como el análisis racional de las oportunidades y amenazas que presenta el entorno para la empresa, de los puntos fuertes y débiles de la empresa frente a su entorno, y la selección de un compromiso estratégico que mejor satisfaga las aspiraciones de los directivos en relación con la empresa (Menguzzato, 1991). Otros autores afirman, consiste en el esfuerzo sistemático y más o menos formal de una compañía para establecer sus propósitos, objetivos, políticas y estrategias

básicas, para desarrollar planes detallados con el fin de poner en práctica las políticas y estrategias, y así lograr los objetivos y propósitos básicos de la compañía (Steiner, 1996; en Armijo, 2009).

Según Armijo (2009), la planeación estratégica es una herramienta de gestión que permite apoyar la toma de decisiones de las organizaciones, en torno al quehacer actual y las metas futuras, para adecuarse a los cambios y demandas que impone el entorno y lograr mayor eficiencia, eficacia y calidad en los bienes y servicios que se proveen. Por su parte, Ronda (2006) afirma, es el proceso iterativo y holístico de formulación, implantación, ejecución y control de un conjunto de maniobras que garantizan una interacción proactiva de la organización con su entorno, para coadyugar la eficiencia y eficacia del cumplimiento de su objeto social.

Por otra parte, la autora Eyzaguirre (2006) afirma, la planificación estratégica establece los objetivos de la organización y define los procedimientos adecuados para alcanzarlos, constituye la orientación o guía para que la organización obtenga y aplique los recursos para lograr los objetivos; y que sus miembros desempeñen determinadas actividades y tomen decisiones congruentes con los objetivos y procedimientos escogidos. Salazar (2006), por su parte, señala la planeación estratégica permite hacer un análisis de la misión, propósitos, y estrategias para el cumplimiento de los objetivos y la obtención de resultados.

De igual modo, Eyzaguirre (2006) expresa, el proceso de planificación comprende diversas fases: definir la misión y visión de la organización, establecer objetivos y metas, desarrollar supuestos acerca del entorno en que se desarrolla la organización, tomar decisiones respecto a las acciones a seguir, emprender las acciones elegidas, y finalmente evaluar la retroalimentación del desempeño para volver a planificar. La autora declara, la planificación estratégica es un proceso dinámico que desarrolla la capacidad de las organizaciones para fijarse un objetivo en forma conjunta y observar, analizar y anticiparse a los desafíos y oportunidades que se presentan, tanto con relación a la realidad interna como a las condiciones externas de la organización, para lograr dicho objetivo (Eyzaguirre, 2006).

En relación a lo anterior, la autora expone que para un adecuado proceso de planificación estratégica, es importante considerar el marco de referencia general y el contexto global en el que se desenvuelve la organización. Asimismo, plantea, el proceso de planificación estratégica supone además, el análisis de las ventajas competitivas y debilidades de la institución, así como de su entorno, la formulación y puesta en marcha de estrategias que le permita alcanzar su propósito planteado en el largo plazo (Eyzaguirre, 2006).

Por otro lado, el autor Contreras (2013) afirma, la planeación estratégica requiere que las personas encargadas de tomar las decisiones en una empresa tengan claro qué clase de estrategias van a utilizar, y cómo van a adecuar las distintas alternativas que se van a presentar en la medida en que van creciendo o posicionándose en el mercado.

Por otra parte, Rodríguez (1998) expone las diferencias entre la planificación estratégica con respecto a la planificación tradicional, conocida también como planificación a largo plazo. Este autor afirma que la planeación tradicional da por hecho que el sistema organizacional es un sistema cerrado, en el cual se puede elaborar un plan articulado y definido, mientras que la planificación estratégica considera el sistema abierto, en el que la organización está llamada a cambiar, en la medida en que integra las diferentes informaciones provenientes del medio externo e interno (Rodríguez, 1998; en Ronda, 2005).

Del mismo modo, señala que la planeación tradicional, basada en fórmulas y en modelos, trata de cambiar la realidad haciendo poco caso de los valores y las situaciones cambiantes, en contraste con la planificación estratégica, que incorpora la realidad cambiante y es considerada como una actividad lógica y realista que pone énfasis en el análisis externo, sin descuidar el interno (Rodríguez, 1998; en Ronda, 2005). Por último, apunta que la documentación que produce la planeación estratégica se debe modificar periódicamente, a medida que las informaciones evolucionan; mientras que la planeación convencional establece proyectos estáticos, que representan un gran esfuerzo intelectual, pero que en la práctica tienen muy poca utilidad para la organización (Rodríguez, 1998; en Ronda, 2005).

En relación a los beneficios de la planificación estratégica, Pimentel (1999) afirma la planificación estratégica permite pensar en el futuro, visualizar nuevas oportunidades y

amenazas, enfocar la misión de la organización y orientar de manera efectiva el rumbo de una organización, facilitando la acción innovativa de dirección y liderazgo. Además, expresa que la misma es una manera intencional y coordinada de enfrentar la mayoría de los problemas críticos, intentando resolverlos en su conjunto y proporcionando un marco útil para afrontar decisiones, anticipando e identificando nuevas demandas. Finalmente, enuncia, la planificación estratégica genera fuerzas de cambio que evitan que las organizaciones se dejen llevar por los cambios, las ayuda a tomar el control sobre sí mismas y no sólo a reaccionar frente a reglas y estímulos externos (Pimentel, 1999).

En la revisión del marco teórico referencial existe la tendencia a emplear indistintamente los términos formulación de estrategias (Ronda, 2006), planificación estratégica (Menguzzato, 1991), planeación estratégica (Steiner, 1996; Armijo, 2009) y dirección estratégica (Menguzzato, 1991; Ronda, 2006), entre otros términos, para referirse al proceso de planeación estratégica. Algunos autores (Menguzzato, 1991; Ronda, 2006; Armijo, 2009) coinciden en abordar el proceso de planeación estratégica desde una perspectiva de gestión, en torno al proceso de toma de decisiones en las organizaciones, en los diferentes niveles decisionales. De forma general, en los diferentes conceptos se expone que el proceso de planeación estratégica está basado en un análisis de las características actuales de la empresa y su relación con el entorno, permite la definición de metas y objetivos, así como de métodos y herramientas en las organizaciones, para alcanzar dichos objetivos con base en el análisis sistemático y riguroso de las relaciones que establece con el entorno.

Con respecto a la planificación estratégica, Cámara et al. (2005) señalan, la misma pone un énfasis especial, aunque no único, en el análisis del contexto y su evolución pasada y futura, así como en las estrategias generales que permitan a la organización diferenciarse del resto de las organizaciones y posicionarse con éxito en ese entorno, adaptándose permanentemente a sus cambios y exigencias. En relación a lo anterior, Menguzzato (1991) expresa, el proceso de planificación estratégica potencia el desarrollo de estrategias en relación y frente al entorno, y básicamente estrategias competitivas, es decir, que permite a la empresa mantener y mejorar su posición ante los competidores.

1.3 Proceso de diagnóstico estratégico

En relación al proceso clásico de planificación estratégica, Menguzzato (1991) señala que la necesidad de una metodología que guíe la formulación de la estrategia empresarial se hace patente a partir de la concepción de la estrategia como un conjunto de objetivos y de líneas de acción orientadas hacia el futuro, como expresión de la voluntad de la empresa frente a los muchos factores que condiciona su evolución.

Por otra parte, respecto al diagnóstico estratégico, Menguzzato (1991; ver figura 1) afirma que el mismo consiste en dos análisis paralelos que van a permitir conjuntamente una evaluación de la situación actual y potencial de la empresa en relación con su entorno. Estos dos análisis son: el análisis externo y el análisis interno. De igual modo, la autora aclara que este doble análisis encaminado a explicitar cuál es la situación actual de la empresa y su potencial de evolución nos informa acerca de lo que es, lo que hace y lo que puede hacer la empresa.

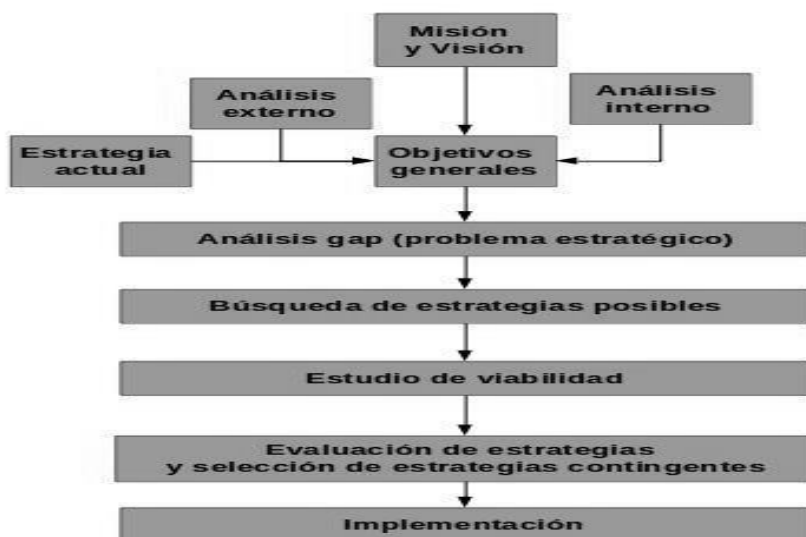


Figura 1: Proceso clásico de planeación estratégica

El análisis externo es definido por Menguzzato (1991) como un estudio de los impactos de los llamados factores estratégicos del entorno tanto general como específico y la evolución futura de estos, a fin de detectar las posibles amenazas y oportunidades para la empresa.

Con respecto al entorno empresarial, Hunger (2007) señala que el mismo está integrado por variables (oportunidades y amenazas) que se encuentran fuera de la organización y que no están comúnmente bajo el control a corto plazo de la administración de alto nivel.

Por otro lado, como se menciona anteriormente, Ronda (2005) divide el entorno en macroentorno y microentorno. En relación al macroentorno, dicho autor señala, el mismo está constituido por las fuerzas político-legales, económicas, tecnológicas y sociales y que, en menor o mayor grado afecta a todas las organizaciones. Del mismo modo, afirma, el microentorno está constituido por un grupo de fuerzas más específico, que influyen directamente y afectan poderosamente la planeación estratégica de las actividades de la organización y para su análisis se emplean las cinco fuerzas de Porter (Ronda, 2005).

Menguzzato (1991) define el análisis interno como un estudio para identificar y valorar la estrategia actual y la posición de la empresa frente a la competencia, así como evaluar los recursos y habilidades de la empresa a fin de conocer cuáles son los puntos fuertes que se esforzará de explotar en una nueva estrategia, así como los puntos débiles que se intentará remediar o eliminar. Además, afirma que este análisis debe ir orientado a una evaluación del potencial de la empresa, de su capacidad global, como un sistema que incluye todas las capacidades específicas desarrolladas en cada función básica de la empresa Menguzzato (1991). Del mismo modo, Ronda (2005) considera, el análisis interno tiene como objetivo identificar y evaluar las capacidades internas de la organización, es decir, las principales fortalezas y debilidades de la misma.

Como ha señalado Hunger (2007), el ambiente interno de una organización está integrado por variables (fortalezas y debilidades) presentes dentro de la organización misma. Estas variables forman el contexto en el que se realiza el trabajo e incluyen la estructura, la cultura y los recursos de la organización.

Considerando lo anterior, se destaca el planteamiento de Ronda (2005) enunciando que las fuerzas tanto del macroentorno y microentorno, y los factores internos se manifiestan de manera diferente para las distintas organizaciones, por ello se define como primera variable la forma de manifestación de esa fuerza externa o factor interno, luego se evalúa el impacto que tiene para la organización dicha forma de manifestación, y luego la capacidad de respuesta para aprovechar o mitigar el impacto. El conocimiento de estos elementos propicia las condiciones para determinar si se está en presencia de amenaza, oportunidad, fortaleza o debilidad (Ronda, 2005). En relación a lo anterior, se resume, la clasificación de un factor externo como amenaza u oportunidad depende de las siguientes variables: impacto del factor externo o interno sobre la organización, capacidad de respuesta de la organización para responder ante dicho impacto e intensidad del efecto del impacto (Ronda, 2005).

Como apunta Scott (2011), la clave del proceso de diagnóstico estratégico está en el estudio, análisis y comprensión para establecer la posición estratégica y en consecuencia trazar las estrategias adecuadas para sacar el máximo de provecho de las oportunidades, reducir los impactos de las amenazas y atenuar las debilidades que permita mantener las ventajas competitivas.

Ronda (2005) señala, que una vez concluido el análisis del entorno y el análisis interno se conocen los factores que influyen positiva y negativamente sobre la empresa y su forma de manifestación, en forma de amenaza, oportunidades, fortaleza o debilidad, pero aún no se conoce el grado de intensidad del impacto en que se manifiesta cada uno, que permita definir la posición estratégica externa e interna de la empresa, lo cual significa conocer el predominio de algunos de los factores señalados. Las herramientas empleadas para determinar la intensidad del impacto con que se manifiesta los factores en la organización es la matriz de evaluación de factores internos y la matriz de evaluación de factores externos (véanse Figuras 2 y 3).

Factor externo	Formas de manifestación del factor externo	Magnitud de Impacto	Capacidad de respuesta	Intensidad del efecto del impacto
Proviene del análisis de los factores externos	Oportunidad(0) Amenaza (1)	0.00	0.00	0.00

Figura 2: Matriz de evaluación de factores externo, Adaptado de Ronda (2005)

Factor interno	Formas de manifestación del factor interno	Magnitud de Impacto	Capacidad de respuesta	Intensidad del efecto del impacto
Proviene del análisis de los factores internos	Fortaleza (0) Debilidad (1)	0.00	0.00	0.00

Figura 3: Matriz de evaluación de factores interno, Adaptado de Ronda (2005)

El autor Scott (2011) emplea el término “resumen de factores externos” para referirse a la matriz de evaluación de los factores externos y el término “resumen de factores internos”

para referirse a la matriz de evaluación de factores internos, define las mismas como una manera de organizar los factores externos e internos en las categorías generalmente aceptadas; y analizar cómo responde la administración (calificación) de una empresa en particular a estos factores específicos en vista de la importancia percibida (valor) de estos para ella.

El procedimiento expuesto por Ronda (2005) para la evaluación del impacto de los factores externos e internos utiliza las matrices de evaluación de impacto de factores. Los pasos para la evaluación de los factores externos e internos son: listado de factores externos e internos que intervienen en la organización, determinación de la forma de manifestación de cada factor en relación a la organización como amenaza u oportunidad, fortaleza o debilidad, evaluación del grado de impacto de cada fuerza o factor sobre la organización asignándole una ponderación, evaluación del nivel de capacidad de respuesta que posee la organización, y la determinación de la intensidad del efecto del impacto de cada factor evaluado sobre la organización (Ronda, 2005). El autor declara, una vez conocida la posición estratégica interna y externa de la empresa, a través de las matrices de evaluación de impactos, se proceda a establecer la posición estratégica general de la empresa. Para el establecimiento de la posición estratégica general, una de las herramientas que se emplea es la matriz DAFO. En la revisión de la bibliografía consultada, la matriz DAFO también es conocida como matriz FODA (Ronda, Hunger), matriz SWOT por sus siglas en inglés, análisis FODA (Hunger) y matriz de resumen de factores estratégicos (Hunger). Al respecto, Hunger (2007) afirma, el término FODA es un acrónimo que se usa para describir las fortalezas, oportunidades, debilidades y amenazas particulares que son factores estratégicos para una empresa específica. Además, señala que la matriz de resumen de análisis de factores estratégicos de una organización resume estos factores combinando los factores externos con los factores internos. Para la realización de ésta se requiere un decisor que resuma las fortalezas, debilidades, oportunidades y amenazas en menos de diez factores estratégicos, lo cual se realiza mediante la revisión del valor dado a cada factor. El uso de esta matriz reduce la lista de factores estratégicos a un número manejable.

Con respecto a la elaboración de la matriz DAFO, Ronda (2005; véase figura 4) propone un conjunto de pasos para la elaboración de dicha matriz. Se debe comenzar

introduciendo los factores claves evaluados en los diagnósticos interno y externo de la organización, es decir, las fortalezas, debilidades, amenazas y oportunidades, escogiendo aquellos factores que se hayan determinado a través de la ponderación en las matrices de evaluación como predominantes, es decir, las amenazas, debilidades, fortalezas y oportunidades con mayor impacto en la organización. Posteriormente, se procede a evaluar cada impacto cruzado. El procedimiento que se emplea es evaluar cada impacto asignándole una ponderación que oscila entre 0 y 3 asumiendo que: 0 el impacto que no tiene ninguna intensidad o valor, 1 al impacto que posee poca intensidad, mayor intensidad 2 y el impacto con una máxima intensidad se le asignan 3 puntos, luego se define el valor a cada impacto cruzado. Esta valoración ponderada permite centrar la atención en aquellos impactos que poseen un máximo de intensidad y tienen un carácter estratégico para la organización. Luego de realizar la ponderación de cada impacto cruzado, se define la posición mediante la suma de la ponderación en cada cuadrante, definiéndose la posición estratégica de la empresa a partir del cuadrante con la votación mayor (Ronda, 2005).

	Oportunidades					Amenazas				
FORTALEZAS										
DEBILIDADES										

Figura 4: Matriz DAFO

Como conclusión final, Scott (2011) destaca que el paso del diagnóstico estratégico es esencial por cuanto propicia el conocimiento de las amenazas, oportunidades, fortalezas y debilidades y para ello se ha estandarizado el empleo de las matrices de evaluación de

factores internos y externos (matriz DOFA), no obstante, las variables y procedimientos de estas matrices no son suficientes para definir una posición estratégica fiable. Por esta razón el diagnóstico estratégico no se limita a la construcción y análisis de esta matriz (Scott, 2011). Por su parte, Hunger (2007) afirma, el análisis DAFO no sólo debe permitir la identificación de las competencias distintivas de una corporación, sino también identificar las oportunidades que la empresa no es capaz de aprovechar actualmente, debido a la falta de recursos adecuados y en la actualidad, se considera una de las técnicas analíticas más usadas en el diagnóstico estratégico.

1.4 Formulación de estrategias

Hunger (2007) define la formulación de estrategia como el proceso que consiste en encontrar una concordancia estratégica entre las oportunidades y fortalezas, y al mismo tiempo, trabajar con las amenazas y debilidades. La formulación de la estrategia general se ocupa del desarrollo de las estrategias genéricas y las políticas de una corporación. En general, se ha extendido el uso de la matriz DAFO como herramienta para la formulación de la estrategia. Dicha matriz ilustra cómo las oportunidades y amenazas se relacionan e impactan con las fortalezas y debilidades en la organización, para generar series de posibles alternativas estratégicas (Hunger, 2007).

En la implementación de la matriz DAFO (véase figura 5) se identifican cuatro alternativas estratégicas, que conducen según sus elementos a la definición de la estrategia general, las cuales pueden ser: la estrategia FO, estrategia FA, estrategia DO, y la estrategia DA, o usualmente clasificadas, como estrategias ofensivas, defensivas, de supervivencia, o de adaptación, según el cuadrante de mayor impacto entre los factores.

	Oportunidades	Amenazas
FORTALEZAS	<p>Estrategia FO: Crear aquí estrategias que usen fortalezas para aprovechar oportunidades</p>	<p>Estrategia FA: Crear estrategias que usen fortalezas para evitar amenazas</p>
DEBILIDADES	<p>Estrategia DO: Crear estrategias que aprovechen oportunidades superando las debilidades</p>	<p>Estrategia DA: Crear estrategias que minimicen las debilidades y eviten las amenazas</p>

Figura 5: Formulación de estrategias, Adaptado de: Hunger, 2007

En relación con las estrategias mencionadas, Hunger (2007) afirma, las estrategias DAFO se crean con base en las maneras en que la empresa o unidad de negocio podría usar sus fortalezas para aprovechar las oportunidades. Esa estrategia constituye el cuadrante más ventajoso, donde todas las empresas querrían estar ubicadas para utilizar sus fortalezas en el aprovechamiento de las oportunidades (Díaz y Hernández, 2011).

Por otra parte, Díaz y Hernández (2011) plantean, la estrategia FA se basa en que las fortalezas de la institución pueden enfrentar las amenazas del entorno. Su objetivo es maximizar las primeras mientras se minimizan las segundas. Esta situación no significa necesariamente que la organización deba dedicarse a buscar amenazas para enfrentarlas, sino todo lo contrario, las fortalezas de la empresa deben estar aplicadas a discreción y oportunamente. La posición correcta para una empresa en tal situación es la defensiva (Díaz y Hernández, 2011). Asimismo, expresan, la estrategia DO requiere minimizar las debilidades y maximizar las oportunidades. Una empresa en esta situación identifica las oportunidades que ofrece el entorno, pero reconoce que sus debilidades organizacionales no permiten aprovecharlas. En esta situación la empresa debe asumir una posición adaptativa (Díaz y Hernández, 2011).

Finalmente, señalan que la estrategia DA tiene como objetivo minimizar tanto las debilidades como las amenazas. Una organización que resulte ubicada en este cuadrante

estaría enfrentando su peor situación respecto al logro de sus objetivos, sus esfuerzos principales tendría que dedicarlos a luchar por su supervivencia o llegaría irremisiblemente hasta su liquidación definitiva. Como alternativa de estrategia se puede asumir la reducción de las operaciones en busca de minimizar las debilidades, o esperar un cambio del entorno donde desaparezcan las amenazas, esta última a un elevado riesgo de no resultar exitosa. Cualquiera sea la estrategia seleccionada, la posición DA será la más peligrosa y se sugiere asumir una actitud de supervivencia (Díaz y Hernández, 2011).

1.5 Modelos de planeación estratégica

Para el proceso de planeación estratégica se han concebido diversos modelos. Un modelo es una representación abstracta de una idea o realidad. Los modelos de planeación estratégica presentan las etapas para la realización de dicho proceso.

En el estudio de los modelos de planeación estratégica, el autor toma como base el análisis realizado por Ronda (2005). En este análisis se revisaron 32 modelos de planeación estratégica diseñados por diversos autores, entre los años 1984 y 2002. El autor del presente trabajo realizó una selección de los modelos de planeación estratégica analizados por Ronda, tomando aquellos centrados en la formulación de las estrategias, clasificados por Ronda como pertenecientes al grupo III (ver tabla 1)

Tabla 1. Modelos de planeación estratégica. Adaptado de Ronda (2005)

Autor	Modelos de Planeación estratégica
Marina Menguzzatto (1984)	Misión Análisis del entorno Análisis interno Objetivos generales Estrategia actual Análisis del problema estratégico Búsqueda de estrategias posibles Estudio viabilidad Estudio de estrategias alternativas y elección de la estrategia Implementación
George A. Steiner (1996)	Expectativas de accionista Deseos y valores directivos Crecimiento y utilidades deseados Factores de utilidad/ crecimientos futuros

	<p>GAP Potencialidades de la compañía Debilidades de la compañía Oportunidades ambientales Peligros Problemas especiales Estrategias Implantación Resultados operacionales</p>
Joaquín Valencia Rodríguez (1999)	<p>Actual postura estratégica Análisis del entorno Análisis interno Valores, aspiraciones de los gerentes Presiones y responsabilidad social Rango limitado de alternativas y de estrategias Plan estratégico Plan mediano plazo Plan corto plazo Implementación/ Resultado</p>
J. Argenti.	<p>Fijar metas. (aclarar los objetivos de la empresa, fijar niveles de metas de los objetivos) Análisis de la brecha. (pronosticar resultados futuros con estrategias presentes, identificar brechas entre pronóstico y brecha) Evaluación de estrategias.(evaluación del entorno e interno, identificación de las ventajas competitivas y redefinir metas a partir de la información de la etapa 3) Formulación de estrategias. (generar opciones estratégicas, evaluar las opciones estratégicas y tomar decisiones) Aplicación de la estrategia</p>
Carlos Lazo Vento (1998)	<p>Misión Procesos estratégicos claves Implicados principales Diagnóstico estratégico Escenario tendencial Visión Objetivos generales Revisión de las direcciones estratégicas Objetivos estratégicos por ARC Programas de implantación Implementación municipal Diseño estrategia empresarial Diseño estrategia municipal Amplia divulgación, compromiso, negociación de los objetivos</p>
Rogelio Gárciga Marrero (1999)	<p>Análisis del entorno actual y futuro Análisis interno Visión</p>

	Misión (cultura, profesión, principios) Objetivos generales estratégicos Análisis de las estrategias actuales Determinación de la brecha estratégica Formulación de las estrategias, evaluación y selección Implementación
--	---

Los modelos de planeación estratégica presentados con anterioridad son conocidos como modelos tradicionales del proceso clásico de planeación estratégica. Dichos modelos centran su atención en los conceptos relacionados a la formulación de las estrategias. En el análisis de los modelos propuestos, Ronda (2005) afirma, no existe un modelo universal que pueda ser aplicado como recetas a la deficiencia en la dirección de las organizaciones, sino que cada uno, responde a las características de la organización para la que fueron creados. Además, plantea que la realización del análisis DAFO se mantiene como característica común en los modelos analizados (Ronda, 2005).

En relación a estos modelos tradicionales de dirección estratégica, Ronda (2005) señala algunas insuficiencias y limitaciones presentes. Entre las limitaciones, plantea que dichos modelos no explican suficientemente los procedimientos para lograr la implementación de la estrategia y presentan insuficiencias en la identificación de las variables que influye en la implementación (Ronda, 2005). Además, expresa que los modelos tradicionales de planeación evidencian una separación entre las fases de planeación y de implementación, y no observan la integración entre los niveles estratégicos, tácticos y operativos.

Estudio de sistemas homólogos.

A nivel internacional existen algunas herramientas para la realización del diagnóstico estratégico. Dichas herramientas presenta algunas características similares a las de la solución a desarrollar en cuanto a sus funcionalidades, aunque presentan ciertas limitaciones para su utilización en el entorno que requiere la solución a desarrollar. A continuación se muestra las principales características de las distintas soluciones existentes que fueron analizadas en el desarrollo de la investigación:

Herramientas	Características
DGIPYMES	Es de uso gratuito Permite salvar y cargar los análisis realizados como archivos en formato xml Permite priorizar las distintas estrategias genéricas de la empresa.

	Presenta una interfaz interactiva y de fácil uso Se encuentra exclusivamente disponible online
Creately	Presenta ejemplos de análisis DAFO de otras empresas Permite la personalización de los diagramas
Inghenia: SWOT	Agrupar los distintos factores por categorías Permite la importación y exportación de datos a archivos de texto Presenta generación automática de los cuadrantes DAFO. Generación de una gráfica que muestra el estado actual y el vector estratégico Es de uso gratuito Permite el manejo de plantillas
SmartDraw	Permite guardar, imprimir, modificar o borrar los análisis DAFO realizados. Permite la realización de diversos tipos de diagnósticos estratégicos para las diferentes áreas de la empresa Es de uso gratuito

En sentido general, las herramientas analizadas presentan funcionalidades similares a la solución que pueden ser utilizadas en la propuesta de solución a desarrollar. Las mismas tienen como desventaja principal que para su uso son accesibles exclusivamente en la web, y no brindan información relacionada sobre los factores luego de la actualización de sus bases de datos. Por otra parte, las herramientas antes mencionadas no permiten brindar una adecuada información a los diferentes niveles directivos de una organización, pues no considera los niveles jerárquicos en el proceso de toma de decisiones, y presenta dificultades para dar soporte a una adecuada comunicación interorganizacional. Además, no ofrece la información relacionada sobre la comparación entre los indicadores empresariales con indicadores a nivel de sector o de región, y su necesaria adecuación a las políticas de desarrollo y condiciones propias de la región, país, localidad o actividad económica en que se encuentra la organización que se evalúa.

1.6 Tecnologías a utilizar en la investigación

Metodologías de desarrollo de software

En la Ingeniería de software son usadas las metodologías de desarrollo para estructurar, planear y controlar el proceso de desarrollo del software. Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Tienen como objetivo guiar a los desarrolladores en la creación de un producto de software. La elección de la metodología juega un papel fundamental en el éxito o fracaso de un producto de software.

Metodologías ágiles:

Las metodologías ágiles se basan en el uso de métodos de Ingeniería de software para un desarrollo iterativo e incremental. Sus métodos de trabajo se basan fundamentalmente en la colaboración de equipos auto-organizados y multidisciplinarios. En cada iteración del ciclo de vida del producto el objetivo es obtener un software que funcione sin errores. Las metodologías ágiles hacen énfasis en la comunicación entre los desarrolladores y el cliente, así como en la obtención de un software funcional. Entre las metodologías ágiles más utilizadas se encuentran Scrum y Programación Extrema XP.

Programación Extrema (XP)

La Programación Extrema es una metodología ágil desarrollada por Kent Beck. Se basa en la simplicidad, la comunicación entre el equipo de desarrollo y el cliente, y la reutilización de código. Tiene como características fundamentales la simplicidad y claridad de las soluciones implementadas, así como la retroalimentación y comunicación constante entre el cliente y el equipo de desarrollo. Es una metodología útil para proyectos con corto plazo de entrega y pequeño equipo de desarrollo.

Scrum

La metodología Scrum define un marco de desarrollo ágil para la gestión de proyectos caracterizado por adoptar una estrategia de desarrollo de software incremental e iterativo, mediante iteraciones denominadas Sprint. Tiene como características principales la gestión regular de las expectativas del cliente, flexibilidad y adaptación, así como la inspección continua. Otro de los aspectos destacado en la metodología Scrum es la priorización del trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. La metodología Scrum está diseñada para adaptarse a proyectos con constantes cambios en los requerimientos.

Justificación del uso de la metodología Programación Extrema XP

La selección de la metodología XP se debe fundamentalmente a que el cliente se encuentra dentro del equipo de desarrollo y la necesidad de entregar un software en un corto período de tiempo. Otra de las causas de esa selección es que el equipo de

desarrollo es pequeño y la simplicidad de la solución a desarrollar, por lo que la metodología XP se hace más adecuada para este tipo de proyectos. Además, se tuvieron en cuenta, las características de la metodología en cuanto a:

Corrección de todos los errores antes de añadir nueva funcionalidad.

Pruebas unitarias continuas, frecuentemente repetidas y automatizadas.

Simplicidad en el código.

Lenguaje de modelado

Para el diseño de la solución fue utilizado el Lenguaje de Modelado Unificado (UML) El Lenguaje de Modelado Unificado es el lenguaje gráfico de modelado orientado a objetos estándar de la industria para especificar construir y documentar los elementos de los sistemas de software. UML permite el modelado de los procesos de negocio y de las estructuras y funciones del sistema, así como de los esquemas de datos. El objetivo del modelado en UML es simplificar el proceso complejo de análisis y diseño de software, facilitando un plano para la construcción. El uso de UML tiene como beneficios que se pueden automatizar algunos procesos y permite generar el código a partir de los modelos.

Otra de sus ventajas es permitir una comunicación sencilla y rápida entre desarrolladores y clientes.

Lenguaje de Programación

Un lenguaje de programación es un idioma artificial diseñado para expresar algoritmos que pueden ser llevadas a cabo por máquinas como las computadoras. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Un lenguaje de programación le da la capacidad al programador de especificarle a la computadora los tipos de datos y las acciones a ejecutar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano. Se utilizan para el desarrollo de software que controlen el comportamiento físico y lógico de una computadora (Zukowski, 2003).

Selección del lenguaje de programación a utilizar:

El lenguaje de programación utilizado en la implementación de las funcionalidades de la aplicación a desarrollar es Java. Su elección se debe principalmente a que es un lenguaje que se ajusta a los requisitos del desarrollo del software a desarrollar y el equipo de desarrollo posee dominio de este lenguaje. Java es un lenguaje multiplataforma que puede ser utilizado para el desarrollo de cualquier tipo de aplicaciones. Entre las características más destacadas de este lenguaje se encuentran su sencillez, portabilidad y estabilidad. A continuación se exponen las características de Java que fundamentan su elección:

Orientado a objetos: Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

Interpretado y compilado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (runtime).

Robusto: Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. El compilador y cargador de clases aseguran la corrección de todas las llamadas de métodos, lo que evita las diferencias implícitas entre tipos y las incompatibilidades entre versiones. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

Seguro: El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Java no permite el manejo de apuntadores, y evita que un programa malicioso corrompa los espacios de memoria. En cuanto a la ejecución de programas Java integra funciones de seguridad: el verificador de código de bit, el cargador de clases

y el gestor de seguridad. Esto garantiza que el código no seguro realice operaciones seguras, como leer el disco duro.

Portátil e independiente de la plataforma: Los programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo. La portabilidad viene dada por la capacidad que tiene Java para ejecutar un mismo programa en cualquier arquitectura.

Multitarea: Un programa que ejecuta varias tareas o subprocesos simultáneamente, se dice que es multitarea, en Java se incluyen múltiples recursos con el objetivo de facilitar la comunicación garantizando la seguridad de los subprocesos.

Dinámico: Las bibliotecas de Java se encuentran evolucionando constantemente, sin embargo eso no implica que los programas anteriores dejen de funcionar.

Herramientas para el modelado. Visual Paradigm

Como herramienta a utilizar para el modelado fue utilizada la herramienta Visual Paradigm, la misma es una herramienta de Ingeniería de Software Asistida por Computadora (CASE) que facilita el modelado de artefactos en un proceso de desarrollo de software, mediante el lenguaje de modelado UML. Es una herramienta diseñada para una amplia gama de usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocio, arquitectos y desarrolladores. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Permite el dibujo de todos los diagramas de clases y la generación del código a partir de los diagramas, así como exportar los diagramas en diversos formatos. Garantiza que el modelo y el código permanezcan sincronizados en todo el ciclo de desarrollo. Entre sus características principales se encuentran: es un software multiplataforma, posee licencia gratuita y comercial, soporta aplicaciones web y la generación de bases de datos. Además, Visual Paradigm brinda la posibilidad de integrarse con los principales Entornos de Desarrollo Integrado y está orientada a la creación de diseños utilizando el paradigma de programación orientada a objetos

Visual Paradigm es la herramienta usada en la universidad para modelar y documentar los productos de software que se desarrollan ya que se cuenta con la licencia que aprueba su utilización. Por tanto se decide utilizar el Visual Paradigm for UML Enterprise Edition en su versión 8.0 como herramienta CASE para el modelado.

Herramienta de desarrollo. NetBeans IDE 8.0

En el desarrollo de la aplicación se utilizará el Netbeans IDE 8.0 como entorno de desarrollo. El NetBeans IDE 8.0 es un entorno de desarrollo visual para aplicaciones programadas a partir del uso del lenguaje de programación Java. Es un producto de código abierto y gratuito, sin restricciones de uso. La plataforma NetBeans IDE 8.0 permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregando nuevos módulos. Las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente. NetBeans IDE 8.0 brinda un completamiento del código de forma inteligente y resaltada. Además, aporta muchas facilidades para el diseño de aplicaciones de escritorio.

Herramienta para pruebas de unidad automatizadas. JUnit 1.4

Para la realización de las pruebas unitarias se utilizó la herramienta JUnit, para automatizar pruebas unitarias. JUnit es un conjunto de bibliotecas utilizadas en programación para hacer pruebas unitarias de aplicaciones Java. Fue creada por Erich Gamma y Kent Beck.

JUnit que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Además, incluye formas de ver los resultados (runners) que pueden ser en modo texto o gráfico (AWT o Swing).

NetBeans 8.0 cuenta con plug-ins que permiten que la generación de las plantillas necesarias para la creación de las pruebas de una clase Java se realice de manera automática utilizando JUnit, facilitando al programador enfocarse en la prueba y el resultado esperado, y dejando a la herramienta la creación de las clases que permiten coordinar las pruebas.

Sistema gestor de base de datos PostgreSQL 9.4

En el desarrollo de la propuesta de solución se utiliza PostgreSQL 9.4 como sistema gestor de base de datos. PostgreSQL 9.4 es un sistema gestor de base de datos objeto-relacional basado en el modelo cliente/servidor. Utiliza multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Es libre, multiplataforma y su código fuente está completamente disponible. Cuenta con dos ventajas claras: un código fuente optimizado que puede ser modificado y adaptado, y una baja inversión por implementación, ya que no existen costos por licencia.

Entre sus principales características están (PostgreSQL, 2010):

- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de base de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee características significativas del motor de datos, entre las que se pueden incluir las subconsultas y los valores por defecto.
- Incorpora una estructura de datos array.
- Permite la declaración de funciones propias, así como la definición de disparadores. Soporta el uso de índices, reglas y vistas.

Sus características técnicas la hacen uno de los sistemas gestores de base de datos más potentes y robustas del mercado. Su estabilidad, facilidad de administración, implementación de estándares y su uso en diferentes plataformas han sido las características que más se han tenido en cuenta durante para su selección.

Conclusiones parciales

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

El análisis de las diferentes herramientas y tendencias sobre las herramientas de diagnóstico estratégico a nivel internacional permitió determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.

El estudio sobre las metodologías, herramientas y lenguajes permitió definir los componentes base para el desarrollo de la solución, donde se define a XP como metodología de desarrollo y como herramienta CASE Visual Paradigm 8.0 para el modelado de los artefactos del análisis y diseño de la solución. Como sistema gestor de base de datos PostgreSQL y como IDE de desarrollo NetBeans 8.0.

CAPÍTULO 2. CARACTERIZACIÓN DEL SISTEMA DE DIAGNÓSTICO ESTRATÉGICO.

2.1 Introducción.

En el presente capítulo se caracteriza la propuesta de solución a través de la definición de los requisitos funcionales y no funcionales, las tareas a realizar durante la implementación y la arquitectura y patrones utilizados en el desarrollo del sistema de diagnóstico estratégico.

2.2 Descripción de los procesos de negocio.

En el proceso de diagnóstico estratégico de la empresa, se realiza un análisis de los factores internos y externos. Los factores internos se clasifican como fortalezas y debilidades. Estos factores representan las características internas propias de la empresa. Por su parte, los factores externos se clasifican como oportunidades y amenazas y son las características del entorno que influyen sobre la empresa.

Para el análisis de los factores internos y externos, es necesario realizar una evaluación del impacto que tienen los mismos sobre la empresa mediante el uso de la matriz de evaluación de impacto de los factores internos y externos. Dicha matriz permite conocer el impacto que tienen los factores estratégicos en relación con la empresa, a partir del peso asignado a cada factor y la capacidad de respuesta de la empresa.

Posteriormente, se realiza la matriz DAFO. La matriz DAFO permite conocer el impacto cruzado de la relación entre los factores internos y externos con el propósito de identificar la estrategia genérica que se debe emplear por la empresa para el logro de los objetivos y metas empresariales. Esta matriz es elaborada mediante la asignación de los pesos del impacto cruzado de los factores internos y externos. Por último, a partir de la realización de la matriz DAFO, se define la estrategia genérica a emplear en la empresa.

2.3 Modelo Conceptual.

El modelo conceptual es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software. (Larman, 2004). Un modelo conceptual es un diagrama de clases en el que

se muestran conceptos u objetos del dominio del problema (clases conceptuales) y las asociaciones entre estas. No muestra comportamiento y las clases conceptuales no pueden tener métodos.

A continuación, se presenta el modelo conceptual con las correspondientes descripciones de los conceptos proceso asociados al negocio.

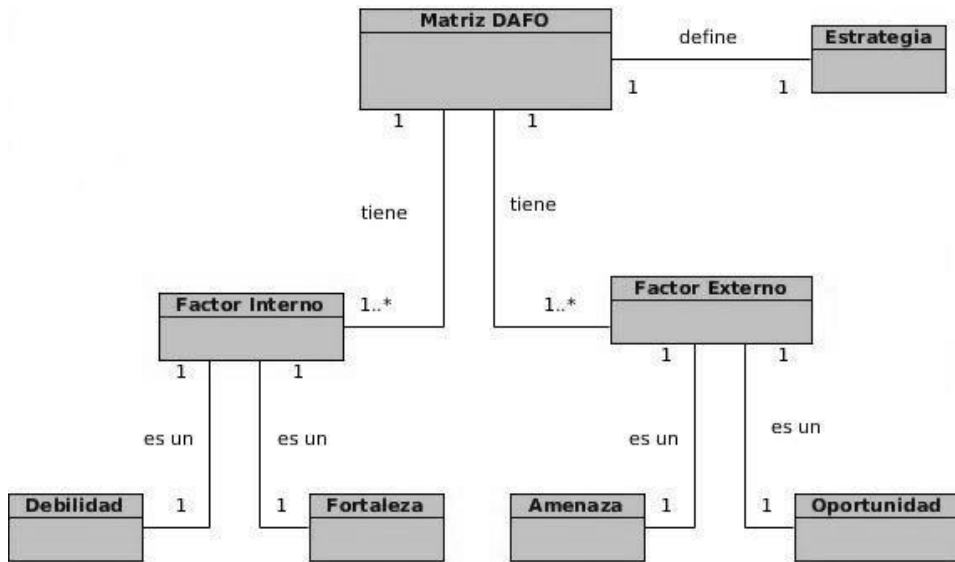


Figura 6. Modelo Conceptual.

Descripción de los elementos del modelo conceptual.

Matriz DAFO: Representa la entidad que se encarga de la construcción de la matriz DAFO y de agrupar los factores internos y externos.

Factor Externo: Es la entidad que representa los factores externos utilizados para la creación de la matriz DAFO. Los factores externos se clasifican en oportunidades y amenazas.

Factor Interno: Es la entidad que representa los factores internos utilizados para la creación de la matriz DAFO. Los factores internos se clasifican en debilidades y fortalezas.

Amenaza: Es la entidad que representa los factores externos clasificados como amenazas.

Oportunidad: Es la entidad que representa los factores externos clasificados como oportunidades.

Fortaleza: Es la entidad que representa los factores internos clasificados como fortalezas.

Debilidad: Es la entidad que representa los factores internos clasificados como debilidades.

Estrategia: Representa la entidad que se encarga de la definición de la estrategia a partir de la realización del gráfico de análisis DAFO.

2.4 Características del sistema.

EL sistema para la realización del diagnóstico estratégico en los proyectos productivos de la Universidad de las Ciencias Informáticas tiene como objetivo automatizar el proceso de diagnóstico estratégico de los proyectos productivos. El sistema permitirá la evaluación del impacto cruzado de los factores internos y externos para determinar la estrategia genérica que ofrezca mejores posibilidades para el logro de los objetivos empresariales de los proyectos de desarrollo vinculados a la producción de software.

El sistema permite que el administrador gestione los factores internos y externos predeterminados. El usuario puede agregar nuevos factores internos y externos. Para la agregación de los factores, el sistema brinda un formulario donde el usuario debe introducir los datos de los factores, además, debe asignar un peso específico y una clasificación a cada factor. Las clasificaciones de los factores se dividen en fortalezas, debilidades, amenazas y oportunidades.

Por otra parte, el sistema permite evaluar el impacto de cada factor interno y externo mediante la realización de las matrices de evaluación de impacto. Para la evaluación del impacto, el usuario debe insertar la magnitud del impacto de cada factor y la capacidad de respuesta con que cuenta la empresa para enfrentar o aprovechar ese factor.

Posteriormente, el sistema le brindará al usuario la evaluación de ese factor.

El sistema permite, además, que el usuario pueda realizar el análisis de los impactos cruzados de los factores estratégicos. El análisis de los impactos cruzados de los factores estratégicos permitirá la formulación de las estrategias genéricas a adoptar por la empresa. Para su realización, el usuario debe asignarle un peso a cada una de las relaciones cruzadas entre los impactos. A partir de este análisis, el sistema brindará la estrategia genérica que mejor se adapta a la empresa analizada.

2.5 Requisitos Funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Los requisitos se obtienen a través de entrevistas, la revisión bibliográfica relacionada con el proceso de formulación estratégica y el estudio de sistemas homólogos.

A continuación se presentan los requisitos funcionales del sistema:

RF Gestionar Factor Interno Predeterminado

RF1 Insertar factor interno predeterminado

RF2 Listar factor interno predeterminado

RF4 Eliminar factor interno predeterminado

RF5 Modificar factor interno predeterminado

RF Gestionar Factor Externo Predeterminado

RF6 Insertar factor externo predeterminado

RF7 Listar factor externo predeterminado

RF8 Eliminar factor externo predeterminado

RF9 Modificar factor externo predeterminado

RF Factores internos

RF10 Insertar factor interno

RF11 Validar factor interno

RF12 Listar factor Interno

RF Factores Externos

RF13 Insertar factor externo

RF14 Validar factor externo

RF15 Listar factor externo

RF Clasificar

RF16 Clasificar Factores Internos (Fortalezas y Debilidades)

RF17 Clasificar Factores Externos (Amenazas y Oportunidades)

RF Clasificar (Impacto y Probabilidad de ocurrencia) □

RF18 Clasificar Factores Internos y Externos

Impacto (alto, medio, bajo)

Probabilidad de ocurrencia (alto, medio, bajo)

RF Evaluación

RF19 Definir Escala de Evaluación

RF20 Evaluar Factores Internos y Externos (Evaluación y Peso)

RF21 Elaborar Matriz de Evaluación para los Factores Internos

RF22 Ponderar impacto de los Factores Internos

RF23 Elaborar Matriz de Evaluación para los Factores Externos

RF24 Ponderar impacto de los Factores Externos

RF Matriz DAFO

RF25 Definir escala de evaluación

RF26 Ponderar interrelación factores internos y externos

RF27 Mostrar matriz cruzada de factores externos e internos

RF28 Mostrar factores de mayor impacto cruzado

RF29 Mostrar cuadrante de mayor impacto cruzado

RF30 Definir estrategia según el cuadrante de mayor impacto cruzado

RF31 Mostrar estrategia

RF32 Elaborar Matriz DAFO

Requisitos no funcionales:

Los requisitos no funcionales son propiedades o cualidades que el software debe tener. Describen las características que hacen al producto de software atractivo, usable, rápido y confiable.

Requisito de apariencia o interfaz de usuario

□ **RNF1** La herramienta debe tener una interfaz sencilla, agradable, legible y de fácil uso para el usuario.

Requisito de usabilidad

RNF2 La herramienta debe garantizar un acceso fácil y rápido a las distintas opciones, y debe permitir ser usada por personas con poco o ningún conocimiento de informática.

RNF3 La herramienta debe contar con una ayuda que permita a los usuarios conocer cómo trabajar con la herramienta.

Requisito de portabilidad

□ **RNF4** La herramienta debe ser multiplataforma, capaz de ejecutarse en los sistemas operativos Windows y Linux.

Requisito de rendimiento

□ **RNF5** La herramienta deber ser capaz de ejecutarse en computadoras con bajas prestaciones de rendimiento y poca memoria.

Requisito de hardware

□ **RNF6** La herramienta se ejecutará en un ordenador que tenga como mínimo las siguientes características:

- Microprocesador: Pentium IV 1,8GHz o superior,
- Memoria RAM: 512 MB ○ Capacidad de almacenamiento: 40GB

Requisito de software

□ **RNF7** Para el uso de la herramienta se debe tener instalado el Java Runtime Enviroment (JRE) en su versión 1.7 o superior.

RNF7 Para el uso de la herramienta se debe tener instalado el sistema gestor de base de datos PostgreSQL en su versión 9.4 o superior

2.6 Historias de Usuarios.

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer. (Ver anexos)

Tabla 2 Historia de Usuario Gestionar Factor Interno predeterminado

No: 1	Nombre: Insertar factor interno predeterminado
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Media
Estimación: 2 días	Iteración: 1
Descripción:	El usuario inserta el nombre de los factores internos predeterminados. El sistema valida que los datos insertados sean correctos y si son correctos introduce los factores internos predeterminados en la base de datos.
Observaciones:	

Tabla 3 Historia de Usuario Gestionar Factor Externo predeterminado

No:2	Nombre: Insertar factor externo predeterminado
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Media
Estimación: 2 días	Iteración: 1
Descripción:	El usuario inserta el nombre de los factores externos predeterminados. El sistema valida que los datos insertados sean correctos y si son correctos introduce los factores internos predeterminados en la base de datos.
Observaciones:	

2.7 Plan de Iteraciones.

Iteración 1:

En la iteración 1 se implementan aquellas funcionalidades que tienen mayor relevancia para el negocio. Son desarrolladas las historias de usuario que tienen mayor prioridad

para el cliente: las principales funcionalidades de gestión y las funcionalidades que soportan la lógica del negocio.

Iteración 2:

La iteración 2 tiene como objetivo desarrollar las historias de usuario complementarias para el negocio. En esta iteración son desarrolladas las historias de usuario de las funcionalidades de prioridad media.

Iteración 3:

La iteración 3 tiene como objetivo desarrollar las historias de usuario de menor importancia para el usuario. Son desarrolladas las historias de usuario de las funcionalidades opcionales o auxiliares.

Tabla 4 Plan de Iteraciones

Iteración	Historias de Usuario	Duración
Iteración 1	Gestionar factor externo predeterminado	10 semanas
	Gestionar factor interno predeterminado	
	Insertar factor interno	
	Elaborar matriz de evaluación de impacto de factores interno	
	Ponderar impacto de factor interno	
	Insertar factor externo	
	Clasificar factor interno (fortalezas y debilidades)	
	Elaborar matriz de evaluación de impacto de factores externo	
	Ponderar impacto de factor externo	
	Elaborar matriz DAFO	
	Ponderar interrelación factores internos y externos	
	Mostrar matriz cruzada de factores externos e internos	

	Clasificar factor externo (oportunidades y amenazas)	
	Evaluar Factores Internos y Externos (Evaluación y Peso)	
Iteración 2	Mostrar factores de mayor impacto cruzado	5 semanas
	Mostrar cuadrante de mayor impacto	
	Definir estrategia según el cuadrante de mayor impacto cruzado	
	Mostrar estrategia	
	Clasificar Factores Internos y Externos	
Iteración 3	Definir escala de evaluación	4 semanas

2.7 Arquitectura de Software.

Estilo arquitectónico

La arquitectura de todo software es un elemento de vital importancia para su desarrollo por lo que debe tenerse en cuenta. Es una vista estructural de alto nivel, ocurre muy tempranamente en el ciclo de vida del software y define los estilos o grupos de estilos adecuados para cumplir con los requisitos no funcionales.

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones donde la vista es la interfaz de usuario y el código que provee de dinámica a dicha interfaz. El modelo es el responsable del acceso a los datos de contenidos en los sistemas gestores de base de datos y el controlador es el responsable de recibir los eventos de entrada desde la vista y de realizar las funcionalidades asociadas a la lógica del negocio.

Modelo Vista Controlador es un patrón de diseño que considera dividir una aplicación en tres módulos claramente identificables y con funcionalidad bien definida: El Modelo, las Vistas y el Controlador (Pantoja, 2004).

Modelo: Todas las aplicaciones de software dejan a los usuarios manipular ciertos datos que proceden de una realidad sobre la que se pretende actuar. A estos datos en estado puro, que representan el estado de la realidad se les llama modelo: modelan la parte de la realidad sobre la que se desea actuar. El modelo es el objeto que representa y trabaja directamente con los datos del programa: gestiona los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los diferentes controladores y/o vistas, ni siquiera contiene referencias a ellos.

Vista: La vista es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio Modelo. Las vistas son las encargadas de:

Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.

Controlador: El controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo. El controlador se encarga de:

Implementar la lógica del negocio, es decir la funcionalidad de la aplicación.

Permitir la validación de los datos.

A continuación, se muestra la arquitectura de MVC de la aplicación:

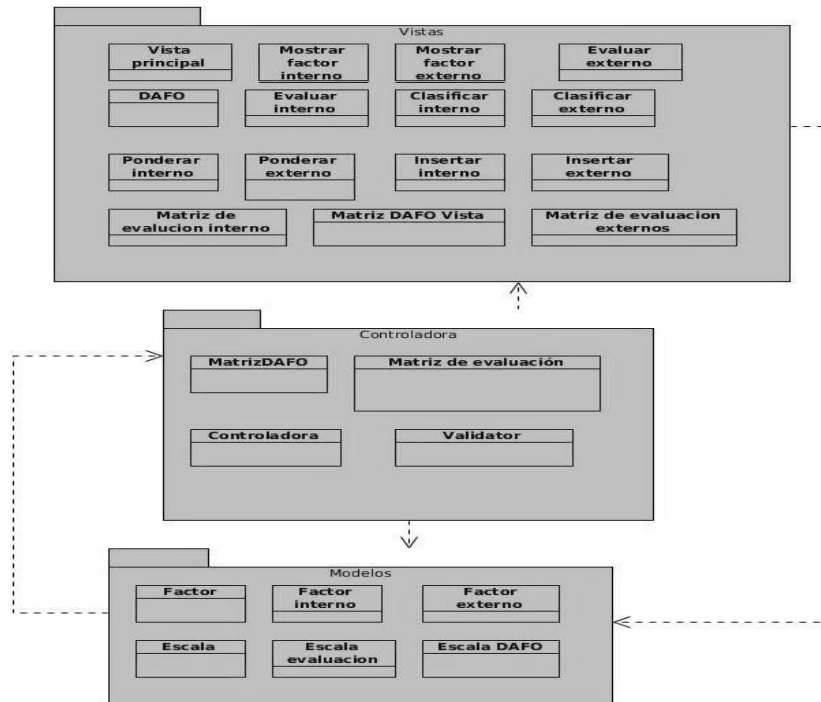


Figura 7. Arquitectura de MVC

2.8 Patrones de Diseño.

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. Muchos patrones proporcionan guías sobre el modo en el que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema (Larman, 2004).

Patrones GRASP

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados (atributos). Una clase se considera experta si contiene toda la información necesaria para realizar la labor que tiene encomendada.

Aplicación del patrón experto:

El patrón experto se puede observar en las clases que se encargan del acceso a los datos. Dichas clases contiene todas las funcionalidades comunes para asignarle las responsabilidades de ejecutar las acciones de las entidades que representan y de la cual poseen información. Un ejemplo concreto es la clase factor interno que contiene los datos y funcionalidades para acceso a datos.

Factor interno
#nombrefactor : string #clasificacion : string #probabilidad : string #impacto : string #evaluacion : int #importancia : int -attribute
+ObtenerFactores() : ArrayList<Factor> +insertarFactor() : void +UpdateClasificacion(nombre : string, clasificacion : string) : void +FactorInterno() +UpdateProbabilidadImpacto()(nombre : string, probabilidad : string, impacto : string) : void +UpdateEvaluacionImportancia(nombre : string, evaluacion : int, importancia : int) : void

Creador: Ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Tiene la información necesaria para realizar la creación del objeto que es una de las actividades más comunes en un sistema orientado a objetos.

Almacena o maneja varias instancias de la clase.

Una clase B tiene la responsabilidad para crear una instancia de la clase A cuando:

B agrega objetos de A,

B contiene objetos de A.

B almacena objetos de A.

B usa objetos de A.

B tiene los datos necesarios para inicializar a A cuando este es creado.(Larman, 2004)

Aplicación del patrón creador:

El uso del patrón de diseño Creador se puede observar en las clases de interfaz de usuario. La interfaz de usuario crea objetos de tipo Controladora que permite acceder a las funcionalidades que implementan la lógica del negocio.

```
public class EvaluarExterno extends javax.swing.JFrame {  
  
    private Controladora control;  
  
    /**  
     * Creates new form EvaluarExterno  
     */  
    public EvaluarExterno() {  
        initComponents();  
        this.setLocationRelativeTo(this);  
        control=new Controladora();  
        this.CargarDatos();  
    }  
  
    /**  
     * This method is called from within the constructor to initialize the form.  
     * WARNING: Do NOT modify this code. The content of this method is always  
     * regenerated by the Form Editor.  
     */  
}
```

Controlador: Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto es para aumentar la reutilización de código y a la vez tener un mayor control. Recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Aplicación del patrón controlador:

La aplicación del patrón de diseño Controlador se puede apreciar en la clase Controladora. Dicha clase actúa como intermediario entre el modelo y la interfaz de usuario y contiene la lógica de negocio. Su principal función es recibir los datos introducido mediante la interfaz de usuario y enviarlos a las distintas clases para la realización de las funcionalidades.

```

1 public class Controladora {
2
3     private ArrayList<Factor> externo;
4     private ArrayList<Factor> interno;
5
6     private ArrayList<Factor> internopredeterminado;
7     private ArrayList<Factor> externopredeterminado;
8
9     public ArrayList<Factor> getExterno() {...3 lines }
10
11    public void setExterno(ArrayList<Factor> externo) {...3 lines }
12
13    public ArrayList<Factor> getInterno() {...3 lines }
14
15    public void setInterno(ArrayList<Factor> interno) {...3 lines }
16
17    public Controladora() {...6 lines }
18
19    public ArrayList<Factor> getInternopredeterminado() {...3 lines }

```

Patrones GOF

Composite: Compone objetos en estructuras arbóreas para representar jerarquías entre el todo y las partes. Manipular todos los objetos del árbol de manera uniforme. Sirve para construir objetos complejos a partir de otros más simples y similares entre sí, gracias a la composición recursiva y a una estructura en forma de árbol.

Aplicación del patrón Composite:

La utilización del patrón Composite se puede observar en la interfaces de usuario creadas con la librería Swing de Java. La clase JFrame de la librería Swing permite crear un árbol de componentes contenidos dentro de un componente de tipo JFrame que puede contener otros objetos. Estos componentes pueden tratarse de manera uniforme dentro del árbol de componentes de JFrame.

2.9 Modelo de la base de datos.

El modelo de datos es una definición lógica, independiente y abstracta de objetos, operadores que en conjunto ayudan a diseñar aplicaciones cuyo objetivo es permitir describir y manipular información almacenada en la base de datos. El modelo de datos estudia los datos independientemente del procesamiento que los transforma. Se compone de tres piezas de información interrelacionadas: el objeto de datos, los atributos

que describen el objeto de datos y la relación que conecta objetos de datos entre sí (Date, 2001).

El modelo de datos es un lenguaje utilizado para la descripción de una base de datos. Permite describir las estructuras de datos de la base de datos (el tipo de los datos que incluye la base de datos y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos. Son abstracciones que permiten la implementación de un sistema eficiente de base de datos.

En la siguiente imagen se visualiza el modelo de datos perteneciente a la solución propuesta.

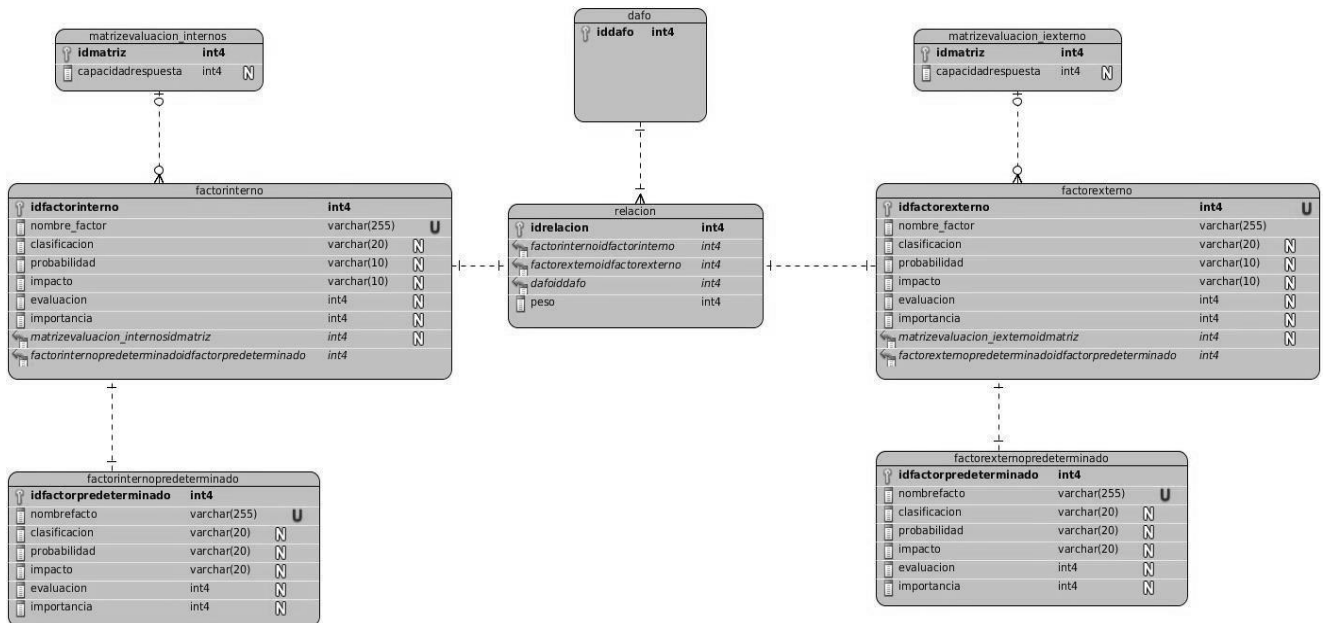


Figura 8. Modelo de la base de datos

2.10 Conclusiones.

Como parte del desarrollo del presente capítulo se determinan las siguientes conclusiones parciales:

La identificación de los requisitos, sirvieron de guía para desarrollar las distintas funcionalidades y de este modo satisfacer las necesidades detectadas.

La metodología de desarrollo utilizada y los patrones de arquitectura y diseño descritos, constituyeron una guía fundamental para la construcción de la propuesta de solución.

Capítulo 3. VALIDACIÓN DEL SISTEMA DE DIAGNÓSTICO ESTRATÉGICO.

3.1 Introducción.

En el presente capítulo se describen los elementos utilizados en la implementación de la herramienta de diagnóstico estratégico para los proyectos productivos de la UCI. Se describen los estándares de codificación empleados durante la implementación. Se define el plan de pruebas, se le realizan los casos de pruebas pertinentes para darle validez a los requisitos funcionales y garantizar el óptimo funcionamiento de la aplicación. Además, se realiza una valoración de los resultados obtenidos al aplicar las pruebas.

3.2 Diagrama de Componentes.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes.

Debido a que los diagramas de componentes son más parecidos a los diagramas de casos de usos, éstos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. En él se situarán librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema.

A continuación, se muestra el diagrama de componentes de la aplicación:

Diagrama de componentes

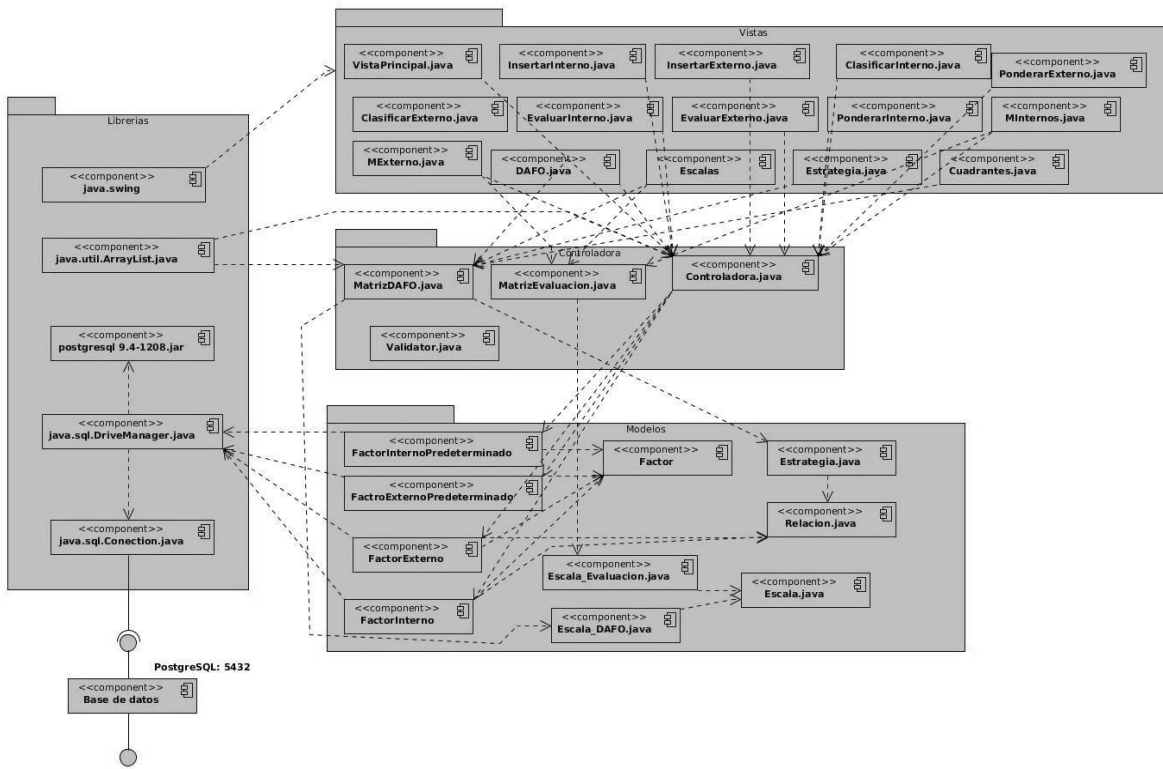


Figura 9. Diagrama de componentes

3.3 Estándares de Codificación.

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico (Hommel, 1999).

Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Al comenzar un proyecto de software, es necesario establecer un estándar de codificación para asegurarse de que todos los programadores del proyecto trabajen de forma coordinada. El mejor método para asegurarse de que un equipo de programadores mantenga un código de calidad es establecer un estándar de codificación sobre el que se efectuarán luego las revisiones del código.

Para la implementación de la propuesta de solución se utilizaron los siguientes estándares de codificación:

Ficheros fuentes

Un fichero consiste en secciones que deben estar separadas por líneas en blanco y comentarios opcionales que identifican cada sección. Se debe evitar tener más de 2000 líneas de código por cada fichero

Cada fichero fuente Java contiene una única clase o interfaz pública. La clase o interfaz pública debe ser la primera clase o interface del fichero. Si existen algunas clases o interfaces privadas asociadas a una clase pública, las mismas pueden ubicarse en el mismo fichero que la clase pública.

Ejemplo:

```
package Modelos;

/**...4 lines */
public class Escala {

    private Integer maxvalue;
    private Integer minvalue;
    private Integer value;
    private Integer semilla;

    public Escala(Integer minvalue, Integer maxvalue, Integer value) {...13 lines }

    public Escala(Integer minvalor, Integer maxvalor){
        this.maxvalue=maxvalor;
        this.minvalue=minvalor;
        this.value=minvalor;
        this.semilla=1;
    }

    public Integer getMaxvalue() {
        return maxvalue;
    }
}
```

Indentación

La indentación se utiliza para mejorar la legibilidad del código fuente por parte de los programadores, teniendo en cuenta que los compiladores o intérpretes raramente consideran los espacios en blanco entre las sentencias de un programa.

En el lenguaje de programación Java, la indentación se realiza utilizando el carácter espacio.

Ejemplo:

```
try(Connection nueva = DriverManager.getConnection(url, usuario, passwd);
    PreparedStatement pstmt =nueva.prepareStatement(sql)){

    Class.forName(driver);
    pstmt.setString(1,this.nombrefactor);
    pstmt.setString(2,this.clasificacion);
    pstmt.setString(3,this.probabilidad);
    pstmt.setString(4,this.impacto);
    pstmt.setInt(5,this.evaluacion);
    pstmt.setInt(6,this.importancia);
    respuesta=pstmt.executeUpdate();

}
catch (ClassNotFoundException exp){
    System.out.println("Fallo de la clase:" + exp.getMessage());
}
catch(SQLException e){
    System.out.print("Fallo SQL" + e.getMessage());
    System.out.print(respuesta);
    return 0;
}
return respuesta;
```

Longitud de la línea

Las líneas no deben tener más de 80 caracteres, ya que no son manejadas bien por muchas herramientas. En la documentación se recomienda mostrar líneas que tengan menos de 70 caracteres.

Ejemplo:

```
Connection nueva=null ;
String sql="Select nombre_factor, clasificacion, probabilidad, impacto,"
          + " evaluacion, importancia from factorinterno where"
          + " clasificacion='Debilidad'";
ResultSet rs=null ;
```

Líneas en blanco

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 entre las secciones de un fichero fuente y entre las definiciones de clase e interfaces, y una línea en blanco entre métodos y entre las variables locales de un método y su primera sentencia, así como entre las distintas secciones lógicas de un método para facilitar su lectura.

Los nombres de las variables deben ser cortos pero con significado. La elección del nombre de una variable debe ser designada para indicar a un observador casual su función.

Ejemplo

```
public abstract class Factor {  
  
    protected String nombrefactor;  
    protected String clasificacion;  
    protected String probabilidad;  
    protected String impacto;  
    protected Integer evaluacion; // Se da por la escala definida  
    protected Integer importancia; // se da por la escala definida  
}
```

Clases

Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Se utilizan nombres de las clases simples y descriptivos. Se debe evitar el uso de abreviaturas.

Ejemplo:

```
public class FactorExterno extends Factor {  
  
    public FactorExterno() { ...2 lines }  
  
    public FactorExterno(String nombre) { ...8 lines }  
}
```

Método

Los métodos que permitan acceder a variables estáticas o de instancia siempre deben llevar el prefijo set o get.

```

public ArrayList<Factor> getExterno() { ...3 lines }

public void setExterno(ArrayList<Factor> externo) { ...3 lines }

public ArrayList<Factor> getInterno() { ...3 lines }

public void setInterno(ArrayList<Factor> interno) { ...3 lines }

```

Declaraciones de variable

Se recomienda una declaración por línea para garantizar la legibilidad y comprensión del código y facilitar el uso de comentarios.

Las declaraciones de variables deben situarse al principio de los bloques. Se recomienda no realizar las declaraciones de las variables en primer uso de la misma; debido a que puede confundir al programador incauto y limitar la portabilidad del código dentro de su ámbito.

Ejemplo:

```

private Float ImportanciaTotal() {
    float suma=0;

    for(int i = 0; i < factores.size(); i++) {
        suma+=factores.get(i).getImportancia();
    }
    return suma;
}

```

Declaraciones de clases e interfaces

En la declaración de la clase la llave de apertura «{» aparece al final de la misma línea de la sentencia de declaración. De igual manera, la llave de cierre «}» empieza una nueva línea indentada ajustada a su sentencia de apertura correspondiente.

Ejemplo

```

public class EscalaEvaluacion extends Escala {

    public EscalaEvaluacion(Integer minvalue, Integer maxvalue) {
        super(minvalue, maxvalue);
    }

}

```

Bloques de sentencia

Los bloques de sentencia se utilizan para realizar un conjunto de sentencias en las estructuras del control del algoritmo (if, if...else, switch, for, while, do while). Deben ponerse encerrados entre llaves "{ }". Las sentencias encerradas deben indentarse un nivel más que la estructura de control. Las llaves se usan en todas las sentencias, incluso las simples, cuando forman parte de una estructura de control. Esto hace más sencillo añadir sentencias sin olvidar las llaves.

La llave de apertura se debe poner al final de la línea que comienza la sentencia compuesta; la llave de cierre debe empezar una nueva línea y ser indentada al mismo nivel que el principio de la sentencia compuesta.

Ejemplo

```

if(Mescalal.isSelected()){
    evaluacion_scala=new EscalaEvaluacion(1,3);
    JOptionPane.showMessageDialog(this,"Usted ha seleccionado la "
        + Mescalal.getText());
}

if(Mescalal2.isSelected()){
    evaluacion_scala=new EscalaEvaluacion(1,5);
    JOptionPane.showMessageDialog(this,"Usted ha seleccionado la "
        + Mescalal2.getText());
    //Code for Scala2 here
}

```

3.4 Modelo de Despliegue.

El diagrama de despliegue, representa de forma visual las relaciones físicas que existen entre los componentes de software y hardware en el sistema. El módulo se encuentra hospedado en un servidor local por ser una aplicación de escritorio y este mismo es la pc cliente. A continuación, se muestra el diagrama de despliegue propuesto para el sistema donde se muestra un solo nodo, que es la PC cliente, en donde van a estar alojados el servidor de base de datos y la aplicación.



Figura 10. Modelo de despliegue

3.5 Pruebas.

El aseguramiento de la calidad del software se ha convertido en una necesidad prioritaria y en una tarea vital en el desarrollo de cualquier sistema informático por la necesidad de garantizar que el producto cumpla con los requisitos especificados y que no presente errores. Por esta razón es necesario establecer un conjunto de pruebas para determinar el correcto funcionamiento del sistema. Existen varias estrategias de pruebas que suelen ser utilizadas, dentro de las que se pueden mencionar:

Pruebas de caja negra: se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.

- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Pruebas de Sistema: Tienen como objetivo verificar el sistema de software para comprobar si el mismo cumple con sus requisitos. Cuenta con distintos tipos de pruebas, como de usabilidad, de rendimiento, de seguridad, entre otras.

3.5.1 Pruebas de Unidad.

La prueba de unidad enfoca 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. Este tipo de pruebas puede realizarse en paralelo para múltiples componentes (Pressman, 1988).

Las pruebas de unidad permiten probar las funcionalidades de las clases de forma aislada para comprobar si las mismas presentan un correcto funcionamiento. La realización de este tipo de prueba ayuda a que cada clase sea evaluada de manera independiente. Lo que quiere decir los métodos de cada clase se pueden probar independientemente al resto de las clases del sistema.

La herramienta JUnit permitió realizar pruebas de unidad automatizadas a la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos se comporta como se espera. Para el desarrollo de las pruebas de unidad se definieron un conjunto de entradas y se evaluó el valor de retorno esperado; si la clase cumple con la especificación, entonces, JUnit devuelve que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

Las clases de la capa de Modelos presentan gran importancia para el correcto funcionamiento de la propuesta de solución, debido a que para el funcionamiento adecuado de la capa Controladora es necesario que las funcionalidades que brindan el acceso a datos se encuentren lo mejor depuradas posible, para que no se propaguen los

posibles errores hacia la capa que contiene las funcionalidades del negocio. Para la realización de las pruebas unitarias se definieron un conjunto de datos de entradas y de salida esperada, conocidos como casos de pruebas, que permitieron verificar que las funcionalidades de la capa de acceso a datos no presenta errores.

Tabla 5 Resultados de las Pruebas de Unidad

Funcionalidad	Datos de entrada	Respuesta esperada	Respuesta Real
InsertarFactor()	Mercado confiable, Fortaleza, Alto, Alto, 1,3	Correcto	Correcto
InsertarFactor()	Producto con elevada calidad, Fortaleza, Alto, Alto, 1, 3	Correcto	Correcto
InsertarFactor()	Altos costos de mano de obra, Amenaza, Medio, 3, 1, 4	Incorrecto	Correcto
InsertarFactor()	1, Amenaza, Medio, Alto, 1 ,4	Incorrecto	Correcto

```

@Test
public void testInsertarFactor() {
    System.out.println("InsertarFactor");
    FactorInterno instance = new FactorInterno("Mercado confiable", "Fortaleza", "Alto", "Alto", 1, 3);
    Integer result = instance.InsertarFactor();
}

```

Figura 11 Código para pruebas

Resultado de las pruebas de unidad.

La realización de las pruebas de unidad permitió detectar los errores en las funcionalidades asociadas a clases. Para el desarrollo de dichas pruebas, las mismas se realizaron en 3 iteraciones, que permitieron obtener un total de 12 no conformidades relacionadas principalmente con las funcionalidades de acceso a los datos de la base de datos.



Figura 12 Resultados de las pruebas

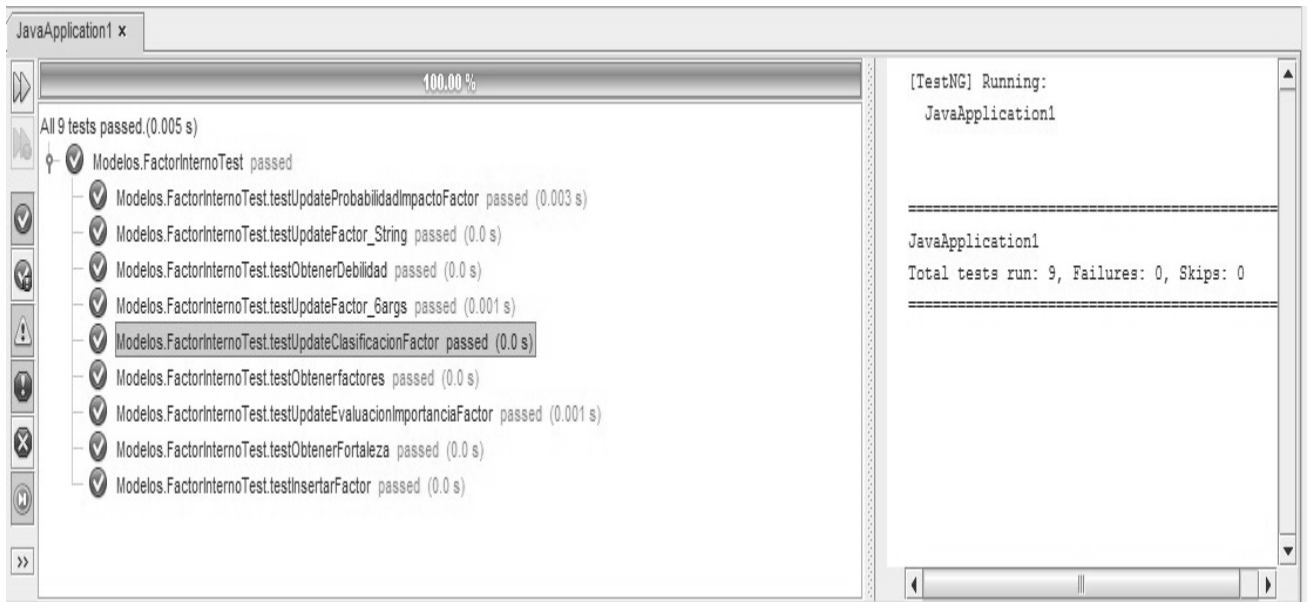


Figura 13 Resultados de las pruebas de unidad

3.5.2 Pruebas de aceptación.

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para

comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

A continuación se muestran los casos de prueba utilizados en las pruebas de aceptación realizadas a la solución (ver anexos):

Tabla 6 Caso de prueba de aceptación: Insertar factor externo

Caso de prueba de aceptación		
Código de caso de Prueba	Nombre de historia de usuario: Insertar factor externo	
Descripción de la prueba: El usuario inserta los factores externos al sistema		
Condiciones de ejecución: Ninguna		
Entrada / Pasos de ejecución: El usuario llena el campo con el nombre del factor externo a insertar y presiona el botón Insertar. El sistema valida que los datos de los factores sean correctos. El sistema inserta los factores en la base de datos		
Resultados de las pruebas	Escenario positivo: El sistema introduce correctamente el factor externo	Escenario negativo: El sistema no puede introducir correctamente el factor externo
	El sistema muestra un mensaje indicando que se insertaron correctamente los datos. El usuario termina de insertar el factor externo.	El sistema muestra un mensaje de error indicando que hubo un problema en la inserción del factor externo.
Evaluación de la Prueba: Satisfactoria		

Resultado de las pruebas de aceptación.

Registro de no conformidades.

Entre los aspectos más importantes al momento de realizar las pruebas se encuentran las no conformidades detectadas, las cuales se traducen en los errores encontrados y funcionalidades no deseados por el cliente. Al final de cada iteración se le muestra al cliente una versión funcional del software de forma que pueda detectar aquellas no conformidades que serán corregidas al inicio de la subsiguiente iteración. Para el desarrollo de la estrategia de prueba se realizaron 3 iteraciones. A continuación se listan las no conformidades detectadas en cada iteración.

Tabla 7 No conformidades encontradas

Iteración	No conformidades
Primera	<ol style="list-style-type: none">1. Inconsistencias en la homogeneidad de las interfaces de usuario de la inserción de los factores externos2. Campos no validados en la interfaz de usuario para insertar los factores internos predeterminados3. Errores ortográficos en el mensaje de error mostrado en la validación de los factores internos4. Error en la carga de datos en la interfaz para mostrar los factores internos predeterminados
Segunda	<ol style="list-style-type: none">5. Inconsistencias en la homogeneidad de la interfaz de usuario de elaboración de la matriz DAFO.6. Error de concordancia en el texto mostrado en el botón para la evaluación de los factores internos7. Errores ortográficos en los mensajes de error asociados a la inserción de los factores predeterminados
Tercera	<ol style="list-style-type: none">8. El botón Insertar no muestra un texto apropiado a la funcionalidad que realiza9. Uso excesivo de letra mayúscula en la interfaz de usuario

3.6 Conclusiones del Capítulo.

La confección del diagrama de componentes permitió observar la integración de los componentes de software.

La utilización de estándares de codificación de código permitió adoptar una estructura homogénea que facilita la comunicación y asegura la calidad, menos errores y fácil mantenimiento.

La aplicación de las pruebas de funcionalidad, y unidad permitieron identificar las principales deficiencias en el desarrollo de la herramienta para solucionar los errores detectados y obtener un producto con un alto valor, pertinencia y utilidad para el diagnóstico estratégico en los proyectos productivos de la UCI.

CONCLUSIONES

1. El estudio de los referentes teóricos y el análisis de las diferentes herramientas y tendencias para el diagnóstico estratégico permitieron determinar las características que constituyen la base para el diseño de las funcionalidades que se definen en la propuesta de solución.
2. La realización de las historias de usuarios facilitó una descripción detallada de las funcionalidades del sistema. A partir de esta implementación se obtuvo un sistema capaz de satisfacer las necesidades del cliente.
3. La validación de la propuesta con las pruebas de unidad y de funcionalidad demostró que el sistema de diagnóstico estratégico para los proyectos productivos de la UCI tiene un correcto funcionamiento y calidad requerida.

RECOMENDACIONES

1. Aplicar la propuesta de herramienta para el proceso de diagnóstico estratégico en los proyectos productivos de la UCI, a fin de contribuir al proceso de toma de decisiones en la organización.
2. Continuar el perfeccionamiento de la herramienta propuesta, a partir de su aplicación práctica en los proyectos productivos de la universidad.

Bibliografía

1. Armijo Marianela, 2009, *Manual de Planificación Estratégica e Indicadores de Desempeño en el Sector Público* [pdf]
2. Bustamante Dayana y Rodríguez Jean C , 2006, *Metodología XP* [pdf] [
3. Cámara Luis, Berzosa Beatriz, Corrêa Émerson. 2005 *PLANIFICACIÓN ESTRATÉGICA Guía para entidades sin ánimo de lucro que prestan servicios de inserción sociolaboral* [pdf]
4. Centro Coordinador de Estudios de Dirección. Ministerio de Educación Superior. 1998. *Dirección por objetivos y Dirección estratégica: la experiencia cubana: compendio de artículos*. Ciudad de La Habana. Cuba. ISBN 959-16-0032-1
5. CONTRERAS SIERRA, Emigdio Rafael. 2013 *El concepto de estrategia como fundamento de la planeación estratégica*. Pensamiento & Gestión, 2013, [pdf]
6. DATE, C. J, 2001 , *Introducción a los sistemas de bases de datos*. Edtion ed.: José Luis Vázquez, 960 p. ISBN 968-444-419-2.
7. Díaz Olivera y Matamoros Hernández: *El análisis DAFO y los objetivos estratégicos*, en Contribuciones a la Economía, marzo 2011, Disponible en <http://www.eumed.net/ce/2011a/>
8. Eyzaguirre, Norma E. Rojas 2006 *METODOLOGÍA INTEGRADA PARA LA PLANIFICACIÓN ESTRATÉGICA* [pdf]
9. Hommel. Scott , 1999 “Convenciones de Código para el lenguaje de programación JAVA TM” Sun Microsystems Inc
10. HUNGER, J. David, et al. 2007, *Administración Estratégica Y Política de Negocios* 10ed. Pearson Educación, [pdf]
11. Larman, Craig. 2004 UML y Patrones. *Introducción al análisis y diseño orientado a objetos* [pdf] Prentice Hall
12. Menguzzato Marina, Renau Piqueras Juan José .1991 *La dirección estratégica de la empresa un enfoque innovador del management*
13. PANTOJA, Ernesto Bascón. 2004 *El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing*. Acta Nova, vol. 2, no 4, p. 493.
14. PATRICIO, Letelier; CARMEN, Penadés.2003. *Metodologías ágiles para el*

- desarrollo de software: extreme Programming (XP)*. [pdf] Disponible en:
<ftp://uclv.edu.cu/ci/is.XP.pdf>
15. Pimentell Villalaz Luis, 1999. *INTRODUCCIÓN AL CONCEPTO DE PLANIFICACIÓN ESTRATÉGICA* [doc]
 16. Ponce Talancón, H. 2006 “*La matriz FODA: una alternativa para realizar diagnósticos y determinar estrategias de intervención en las organizaciones productivas y sociales*” [en línea] [consultado 10 de diciembre 2015] Disponible en: <http://www.eumed.net/ce/>
 17. PostgreSQL-es. [En línea] 2 de octubre de 2010. [Citado el: 12 de mayo 2016.]
http://www.postgresql.org.es/sobre_postgresql
 18. PRESSMAN, Roger S.; TROYA, Jose Maria. *Ingeniería del software*. McGraw Hill, 1988.
 19. Rodríguez González. Fermín O y Alemañy Ramos. S [sin fecha] *Enfoque, dirección y planificación estratégicos. Conceptos y metodología*. [en línea] [consultado: 15 de diciembre 2015] . Disponible en: www.sld.cu/galerias/doc/sitios/infodir/
 20. Ronda Pupo Guillermo. A, José Ángel Marcané Laserra 2005. *DIRECCIÓN ESTRATÉGICA PARA ORGANIZACIONES LATINOAMERICANAS* [pdf]:
 21. Ronda Pupo, Guillermo A., 2006, *Dirección estratégica, constructo y dimensiones*, [pdf] ISBN 978-959-286-003-2 Disponible en: <http://biblioteca.uci.cu>
 22. Salazar Flores , Luis Eduardo. 2006 *LA PLANEACION, EL BALANCED SCORECARD Y UNA PROPUESTA PARA EL BANCO DE LA NACION* [pdf]
 23. Scott Jervis David, 2011 *MANUAL DE PLANEACIÓN CORPORATIVA PARA EMPRESAS DE ASEGURAMIENTO EN SALUD EN COLOMBIA* [pdf] ISBN 978-84-694-5334-6 Disponible en: <http://www.eumed.net/libros-gratis/2011c/993/>
 24. Zukowski, Jhon, 2003, *Programacion Java 2*. [pdf] Disponible en: <http://sunshine.prod.uci.cu>

GLOSARIO DE TÉRMINOS

Amenazas: Son aquellas situaciones que presenta el entorno externo a la organización, que no puede controlar pero le pueden afectar desfavorablemente y en forma relevante.

Fortalezas: Son las características y capacidades internas de la organización que le han permitido llegar al nivel actual de éxito y lo que le distingue de la competencia (ventaja competitiva).

Debilidades: Son las características y capacidades internas de la organización que no están en el punto que debieran para contribuir al éxito y más bien provocan situaciones desfavorables.

Oportunidades: Son aquellos factores externos a la organización que esta puede aprovechar para obtener ventajas competitivas. La organización no los controla y no dependen de esta, pero puede obtener ventajas de tales hechos relevantes.

Ponderación: Acción de ponderar.

Ponderar: Según la Real Academia Española (RAE), determinar el peso de algo.

Ventajas competitivas: Una ventaja competitiva es una ventaja en algún aspecto que posee una empresa ante otras empresas del mismo sector o mercado, y que le permite tener un mejor desempeño que dichas empresas y, por tanto, una posición competitiva en dicho sector o mercado.

ANEXOS

Tabla 6 Historia de Usuario Clasificar Factores Internos

No:	Nombre: Clasificar Factores Internos
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Media
Estimación: 2 días	Iteración: 1
Descripción:	El sistema muestra un listado con los factores internos y permite clasificar los factores en fortalezas y debilidades.
Observaciones:	

Tabla 7 Historia de Usuario Clasificar Factores Externos

No:	Nombre: Clasificar Factores Externos
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Media
Estimación: 2 días	Iteración: 1
Descripción:	El sistema muestra un listado con los factores externos y permite clasificar los factores en amenazas y oportunidades.
Observaciones:	

Tabla 8 Historia de Usuario Clasificar Factores Internos y Externos

No:	Nombre: Clasificar Factores Internos y Externos
Usuarios:	Usuario
Prioridad: Media	Complejidad: Media
Estimación: 2 días	Iteración: 2
Descripción:	El sistema muestra un listado con los factores internos clasificados en fortalezas y debilidades y muestra los factores externos clasificados en amenazas y oportunidades. El usuario debe clasificar los factores según el impacto (alto, medio y bajo) y según la probabilidad de ocurrencia (alto, medio y bajo).
Observaciones:	

Tabla 9 Historia de Usuario Definir escala de evaluación

No:	Nombre: Definir escala de evaluación
Usuarios:	Usuario
Prioridad: Baja	Complejidad: Baja
Estimación: 2 días	Iteración: 3

Descripción:	El sistema muestra varias escalas de evaluación (1-3, 1-5, 1-7, 1-10, -3-3), el usuario debe escoger una de las escalas a utilizar para la evaluación de los factores de alto y medio impacto.
Observaciones:	

Tabla 10 Historia de Usuario Evaluar Factores Internos y Externos

No:	Nombre: Evaluar Factores Internos y Externos
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Alta
Estimación: 3 días	Iteración: 1
Descripción:	El sistema muestra los factores internos y externos clasificados anteriormente para evaluarlos según el peso (importancia) y evaluación según la escala escogida.
Observaciones:	

Tabla 11 Historia de usuario Elaborar matriz de evaluación de impacto de factor interno

No:	Nombre: Elaborar matriz de evaluación de impacto de factor interno
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Alta

Estimación: 3 días	Iteración: 1
Descripción:	El sistema muestra una matriz de factor interno con un listado de los factores internos en la parte izquierda y su clasificación.
Observaciones:	

Tabla 12 Historia de Usuario Ponderar impacto de factor interno

No:	Nombre: Ponderar impacto de factor interno
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Alta
Estimación: 3 días	Iteración: 1
Descripción:	El usuario asigna una evaluación a la magnitud del impacto y a la capacidad de respuesta de la empresa a cada factor interno seleccionado. El sistema valida que todos los campos estén correctamente. El sistema devuelve el peso ponderado del impacto de cada factor interno. El usuario presiona el botón aceptar. El sistema guarda en la base de datos el peso ponderado del impacto de cada factor interno y notifica que el peso ponderado ha sido guardado.
Observaciones:	

Tabla 13 Historia de usuario Elaborar matriz de evaluación de impacto de factor externo

No:	Nombre: Elaborar matriz de evaluación de impacto de factor externo
------------	---

Usuarios:	Usuario
Prioridad: Alta	Complejidad: Alta
Estimación: 3 días	Iteración: 1
Descripción:	El sistema muestra una matriz de factor externo con un listado de los factores externos en la parte izquierda y su clasificación.
Observaciones:	

Tabla 14 Historia de Usuario Ponderar impacto de factor externo

No:	Nombre: Ponderar impacto de factor externo
Usuarios:	Usuario
Prioridad: Alta	Complejidad: Alta
Estimación: 3 días	Iteración: 1
Descripción:	El usuario asigna una evaluación a la magnitud del impacto y a la capacidad de respuesta de la empresa a cada factor externo seleccionado. El sistema valida que todos los campos estén correctamente. El sistema devuelve el peso ponderado del impacto de cada factor externo. El usuario presiona el botón aceptar. El sistema guarda en la base de datos el peso ponderado del impacto de cada factor externo y notifica que el peso ponderado ha sido guardado.
Observaciones:	