

HERRAMIENTA ASISTIDA POR COMPUTADORA PARA LA MODELACIÓN FUNCIONAL DEL DISEÑO MECÁNICO

COMPUTER-AIDED TOOL FOR FUNCTIONAL MODELING OF MECHANICAL DESIGN

Dr.C. Roberto Pérez-Rodríguez
roberto.perez@uho.edu.cu
<https://orcid.org/0000-0001-5741-5168>
Universidad de Holguín, Cuba

Ing. Alexis Cordovés-Rodríguez
cordoves@uho.edu.cu
<https://orcid.org/0000-0002-5710-6194>
Universidad de Holguín, Cuba

Dr.C. Yunior Portilla-Rodríguez
portilla@uho.edu.cu
<https://orcid.org/0000-0002-9147-2470>
Universidad de Holguín, Cuba

Dr.C. Alexis Alvarez-Cabrales
aalvarezc@udg.co.cu
<https://orcid.org/0000-0003-4222-6449>
Universidad de Granma, Cuba

Dr.C. Enrique Ernesto Zayas-Figueras
enrique.zayas@upc.edu
<https://orcid.org/0000-0002-8241-130X>
Universitat Politècnica de Catalunya, Barcelona, España

Tipo de contribución: Artículo de investigación científica

Recibido: 25-02-2022

Aceptado para su publicación: 04-04-2022

Resumen: La etapa de diseño conceptual en el desarrollo de un producto determina de manera significativa su alcance, durabilidad, mantenibilidad y costos asociados. Esta etapa se caracteriza por un alto nivel de subjetividad y creatividad. Dentro de las subetapas del diseño conceptual, la modelación funcional resulta esencial para visualizar las futuras estructuras funcionales del producto. El desarrollo de herramientas que permitan mostrar de manera gráfica las funciones e interrelaciones en las etapas primarias del proceso de diseño, facilitarán de manera efectiva la toma de decisiones. Este artículo muestra el desarrollo de un sistema informático orientado a la modelación funcional del diseño de productos, caracterizado por la disponibilidad de datos y una interfaz amigable, lo que favorece la toma de decisiones en este proceso. La función principal de esta herramienta es asistir al diseñador en el establecimiento de las funciones predeterminadas para la creación de los esquemas o diagramas funcionales. Se utilizó el lenguaje de programación Java y para los diseños de reportes, el sistema informático iReport, unido a la librería JasperReport, así como el SQLite para la Gestión de la Base de Datos. El sistema es multiplataforma y los resultados obtenidos son satisfactorios.

Palabras clave: diseño asistido por computadora; diseño conceptual; modelación funcional; desarrollo de productos

Abstract: The conceptual design stage in the product development process significantly determines its scope, durability, maintainability, and associated costs. This stage is characterized by a high level of subjectivity and creativity. Within the sub-stages of conceptual design, functional modeling is essential to visualize the future functional structures of the product. The development of tools that allow the functions and interrelationships to be graphically displayed in the primary stages of the design process will effectively facilitate decision-making. This article shows the development of a computer system oriented to the functional modeling of product design, characterized by the availability of data and a friendly interface, which favors decision-making in this process. The main function of this tool is to assist the designer in establishing the functions for the creation of functional diagrams or schematics. The Java programming language was used and for the report designs, the iReport computer system, together with the JasperReport library, as well as SQLite for Database Management. The system is multiplatform and the results obtained were satisfactory.

Keywords: computer aided design; conceptual design; functional modeling; product development

1. INTRODUCCIÓN

En el proceso de desarrollo de productos, la etapa de diseño es una de las actividades humanas que demandan un alto grado de análisis y creatividad. La función principal de esta etapa es crear soluciones eficaces y definir un producto, de forma tal que satisfaga los requisitos del cliente y las restricciones establecidas por el entorno (Engelmann, Breining, & Gutowski, 2021).

El diseño en la ingeniería mecánica se divide en cuatro fases o etapas específicas (clarificación de la tarea, diseño conceptual, diseño básico y diseño de detalle). Cada una de las fases posee conexiones con las restantes, por lo que se hace necesario la iteración y la recursión, según el modelo prescriptivo de los investigadores alemanes Pahl y Beitz (Feldhusen & Karl-Heinrich, 2013).

El diseño conceptual es la etapa más compleja dentro del proceso de diseño de un producto. Se caracteriza por un alto grado de subjetividad y creatividad. Generalmente la experiencia previa tiene un rol fundamental en la generación de conceptos, según los modelos descriptivos.

La modelación de la estructura funcional se lleva a cabo mediante diagramas de flujo, los que se representan a través de algoritmos o procesos. Estos diagramas utilizan símbolos con significados bien definidos, que representan los pasos del algoritmo y el flujo de ejecución, mediante líneas que conectan los puntos de inicio y término.

La estructura funcional puede representarse a nivel de la función global del producto, o bien, puede subdividirse en partes que contienen subfunciones de menor complejidad. La subdivisión en subfunciones presenta una gran importancia en el proceso de diseño conceptual y se orienta a tres objetivos: (a) proporcionar una estructura funcional más detallada; (b) facilitar la búsqueda de principios de solución para las subfunciones; y (c) facilitar la creación de la estructura modular del producto.

En el proceso de descomposición funcional, la ausencia de una definición precisa de las subfunciones, es uno de los aspectos que influye en su efectividad (Feldhusen & Karl-Heinrich, 2013). El desarrollo de un conjunto estándar de funciones y flujos, ofrece una vía para minimizar el efecto de la incertidumbre.

El desarrollo de modelos conceptuales ayuda a los estudiantes e ingenieros a comprender y comunicar los sistemas complejos, de manera que puedan ser analizados de una manera funcional y holística.

El pensamiento sistémico (*systems thinking*) constituye una habilidad importante a desarrollar en los estudiantes de ingeniería (Huang, Muci-Kuchler, & Bedillion, 2015). Este tipo de pensamiento, sobre todo en las ingenierías, se basa en recursos gráficos que facilitan mostrar el conocimiento transdisciplinar.

Este artículo muestra el desarrollo de un sistema informático orientado a la modelación funcional del diseño de productos, caracterizado por la disponibilidad de datos y una interfaz amigable, lo que favorece la toma de decisiones en este proceso. La función principal de esta herramienta es asistir al diseñador en el establecimiento de las funciones predeterminadas para la creación de los esquemas o diagramas funcionales.

2. MATERIALES Y MÉTODOS

2.1. Modelos o enfoques del diseño mecánico

Concebir un producto consiste en transformar una idea, en la definición de las características que satisfagan los requerimientos del cliente y en la determinación de sus modalidades de fabricación. El producto pasa por una serie de etapas que van desde su concepción hasta su reciclaje, una vez concluida su vida útil.

Existen numerosas definiciones de diseño mecánico, sin embargo, se considera que las que se relacionan en la Tabla 1 contribuyen a visualizar más el significado moderno del término.

El modelo de Pahl y Beitz divide el proceso de diseño en cuatro fases o etapas (Feldhusen & Karl-Heinrich, 2013), expuestas en el apartado anterior. El nivel de imprecisión del conocimiento de diseño disminuye durante el proceso. Estos niveles de son mayores al inicio, debido a la incertidumbre que se tiene. Una vez que se va avanzando en el mismo, se progresa en el conocimiento y en el grado de madurez de la información.

Cualesquiera que sean las representaciones (semánticas, gráficas, analíticas, físicas), todas son utilizadas en el proceso de diseño del producto. Esta investigación se centra en la etapa de diseño conceptual, donde la información es preferentemente de tipo semántica y con un alto grado de incertidumbre.

Tabla 1. Definiciones de Diseño Mecánico

Mistree, <i>et al.</i>	El diseño mecánico comienza por un proceso de conversión de la información relacionada con las ideas y exigencias para un producto, y termina en un conocimiento más exacto sobre
------------------------	---

	el mismo (Mistree, Smith, Bras, Allen, & Muster, 1990)
Suh, N.P.	El diseño mecánico es la interacción entre lo que se quiere realizar y cómo se va a realizar; es decir, la transformación de las exigencias funcionales en parámetros de diseño (Gyung-Jin, 2007)
Dym <i>et al.</i>	El diseño mecánico en ingeniería es la generación y evaluación sistemática e inteligente de especificaciones para artefactos, cuya forma y función alcanzan los objetivos establecidos y satisfacen las restricciones especificadas (Dym, Little, & Orwin's, 2013)
Ullman, D.G.	El diseño mecánico es la evolución de la formulación de un problema de diseño, asistida por la toma de decisiones en el diseño preliminar y que parte de información tecnológica y social. Las representaciones abstractas evolucionan hacia un producto físico (Ullman, 2017)
Pahl y Beitz	El diseño mecánico es una actividad que afecta a casi todas las áreas de la vida humana, utiliza leyes de la ciencia, se basa en una experiencia especial y define los requisitos para la realización física de la solución (Engelmann et al., 2021)

2.2. Etapa de diseño conceptual y modelación funcional

La etapa conceptual es innovadora, y sus soluciones suelen llevar el germen de todo el desarrollo posterior. Por lo tanto, debe promoverse un ambiente propicio para la creatividad, pero al mismo tiempo, debe fomentarse un sentido crítico y riguroso en la evaluación de las soluciones (poco definidas en esta etapa del diseño). Cualquier omisión, olvido o error de concepto, ocasiona más adelante dificultades importantes en el proyecto (Dym *et al.*, 2013).

En el modelo de Pahl y Beitz (Feldhusen & Karl-Heinrich, 2013) se sugieren dos pasos para el diseño conceptual:

1. El primero constituye una recomendación, pues los diseñadores en el proceso de la búsqueda de soluciones, deberán centrarse en el proceso de abstracción, con el objetivo de identificar la esencia del problema y no tomar soluciones influenciadas por ideas convencionales.
2. El segundo es el establecimiento de la estructura funcional (también denominado modelación funcional). Es lo más importante dentro del diseño conceptual. Tiene por objetivo

la generación de las soluciones que deberán satisfacer los requerimientos funcionales del problema (especificaciones) y luego constituirán la base para el diseño básico y de detalle.

La estructura funcional puede representarse a nivel de la función global del producto o sistema, o bien, dependiendo de su complejidad, puede subdividirse en partes que contienen subfunciones de menor complejidad. La subdivisión en subfunciones presenta una gran importancia en el proceso de diseño conceptual y se orienta a tres objetivos:

1. Proporcionar una estructura funcional más detallada y comprensible, a la vez que menos ambigua.
2. Facilitar la búsqueda de principios de solución para las subfunciones que, por combinación, deben generar principios de solución para la función global.
3. Facilitar la creación de la estructura modular del producto.

En el proceso de descomposición funcional, la ausencia de una definición precisa de las subfunciones, es uno de los aspectos que influye en su efectividad (Ullman, 2017). El desarrollo de un conjunto estándar de funciones y flujos, ofrece una vía para minimizar el efecto de la incertidumbre.

Los dos elementos básicos que representan los diferentes enfoques de la modelación funcional lo constituyen: los flujos y las funciones. La transformación de los requerimientos funcionales en funciones y la definición de los flujos asociados, constituye un proceso complejo de síntesis con una elevada dosis de subjetividad y heurística. Debido a esta característica, el estudio de la trazabilidad de los requerimientos, desde su definición hasta la evaluación conceptual, debe ser abordado en la modelación funcional.

2.3. Herramientas informáticas para la modelación de flujos

2.3.1. Microsoft Visio

Microsoft Visio es el software propietario (Wikipedia, 2021c) de dibujo vectorial para Microsoft Windows. Comenzó a formar parte de los productos de Microsoft en el año 2000. Las herramientas que lo componen permiten realizar diagramas de oficinas, de bases de datos, de flujo de programas, UML (*Unified Modeling Language*), y otros.

Aunque apuntaba a ser una aplicación para dibujo técnico en el campo de la Ingeniería y la Arquitectura, su adquisición por Microsoft implicó

drásticos cambios de directrices. A partir de la versión de Visio para Microsoft Office 2003, el desarrollo de diagramas para negocios pasó de añadido a ser el núcleo central, minimizando las funciones para desarrollo de planos de Ingeniería y Arquitectura.

2.3.2. Dia

Dia es una aplicación informática de propósito general para la creación de diagramas (Wikipedia, 2021b). Está concebida de forma modular, con diferentes paquetes de formas y necesidades. Está diseñada como un programa sustituto de la aplicación comercial Visio de Microsoft. Se puede utilizar para dibujar diferentes tipos de diagramas.

En la actualidad se incluyen diagramas entidad-relación, UML, de flujo, de redes, de circuitos eléctricos, etc. Nuevas formas pueden ser fácil de agregar, dibujándolas con un subconjunto de SVG (*Scalable Vector Graphics*) e incluyéndolas en un archivo XML (*Extensible Markup Language*). El formato para leer y almacenar gráficos es XML. Puede producir salida en los formatos EPS (*Encapsulated PostScript*), SVG y PNG (*Portable Network Graphics*). Gracias al paquete dia2code, puede generar el esqueleto del código a escribir, si se utilizara con tal fin un UML.

2.3.3 OpenOffice.org Draw

OpenOffice.org Draw es un editor de gráficos vectoriales, forma parte de la suite ofimática OpenOffice.org (Wikipedia, 2021a). Ofrece como característica "conectores" versátiles entre figuras, que están disponibles en un rango de estilos de línea y facilitan hacer dibujos como organigramas (y otros tipos de representaciones).

Los usuarios de OpenOffice.org también pueden instalar la Biblioteca Abierta de Clip Art, que agrega una galería de logotipos, íconos y pancartas para presentaciones generales y dibujos. En particular, los sistemas operativos Linux Debian y Ubuntu han proporcionado un paquete de *openclipart* listo para usar y fácil de bajar e instalar desde sus repositorios en línea.

Después de analizar las características de los diferentes softwares, se pueden destacar algunas limitaciones asociadas a su uso:

- Los softwares propietarios impiden realizarles cambios, que lo adapten a las necesidades particulares de la institución que requiera su uso.
- Ninguno de los estudiados realiza las matrices de los diagramas de flujo de forma automática.

- Son muy adecuados para el mundo corporativo de los negocios y no en las mesas de dibujo de Arquitectos e Ingenieros.

Para la modelación funcional de productos se requiere de un software que realice diagramas de flujo con las siguientes características:

- Los diagramas de flujo deben estar representados por funciones, entradas y salidas.
- Los sistemas deben crear la modelación funcional.
- Permitir crear varias conexiones desde un punto a otro, sin limitaciones.
- Cuando se mueva alguna función, se debe mover todo lo asociado a ella.
- Cuando se realicen los diagramas de flujo funcional, se debe generar de forma automática una matriz cualitativa relacionada con las funciones, las entradas y las salidas.

3. RESULTADOS Y DISCUSIÓN

3.1. Herramienta informática para la modelación funcional. Modelo del Dominio

El modelo del dominio captura los tipos de objetos y conceptos fundamentales relacionados con el problema que el sistema pretende resolver. Estos objetos representan eventos que ocurren en el entorno en el que trabaja el sistema. Muchos de estos objetos del dominio o clases se obtienen a partir de la especificación de los requerimientos (Jacobson, Booch, & Rumbaugh, 2000).

En el proceso de ICONIX (proceso de modelado mínimo, basado en casos de uso), el modelo del dominio constituye una parte esencial, pues es refinado y actualizado durante todo el proyecto, de esta forma siempre refleja la comprensión actual del ámbito del problema. El modelo del dominio es en esencia un glosario de términos empleados en el proyecto. Para elaborar este modelo, ICONIX propone una serie de actividades (Rosenberg & Scott, 2007).

Para el desarrollo del sistema informático se aprovechó una de las principales ventajas de ICONIX, o sea, su flexibilidad y adaptabilidad ante requisitos cambiantes. En la Tabla 2 se muestran los requerimientos funcionales de este sistema informático.

Tabla 2. Requerimientos funcionales de este sistema informático (fragmento)

No.	Requerimiento funcional
1	El sistema estará basado en una aplicación Desktop
2	Las entradas y salidas estarán representados por una flecha, las cuales tendrá las siguientes propiedades: nombre, tipo y descripción. Estas pueden ser material, señal y energía
3	Las funciones estarán representadas por un rectángulo, las cuales tendrán las siguientes propiedades: tipo de función, nombre, color y en qué punto del diagrama de flujo funcional será insertada
4	El usuario debe tener la posibilidad de gestionar los diagramas de flujo funcional
5	El usuario debe tener la posibilidad de visualizar el diagrama de flujo funcional
6	El usuario debe tener la posibilidad de guardar el diagrama de flujo funcional en cualquier ubicación
7	El usuario debe tener la posibilidad de modificar los

diagramas de flujo funcional

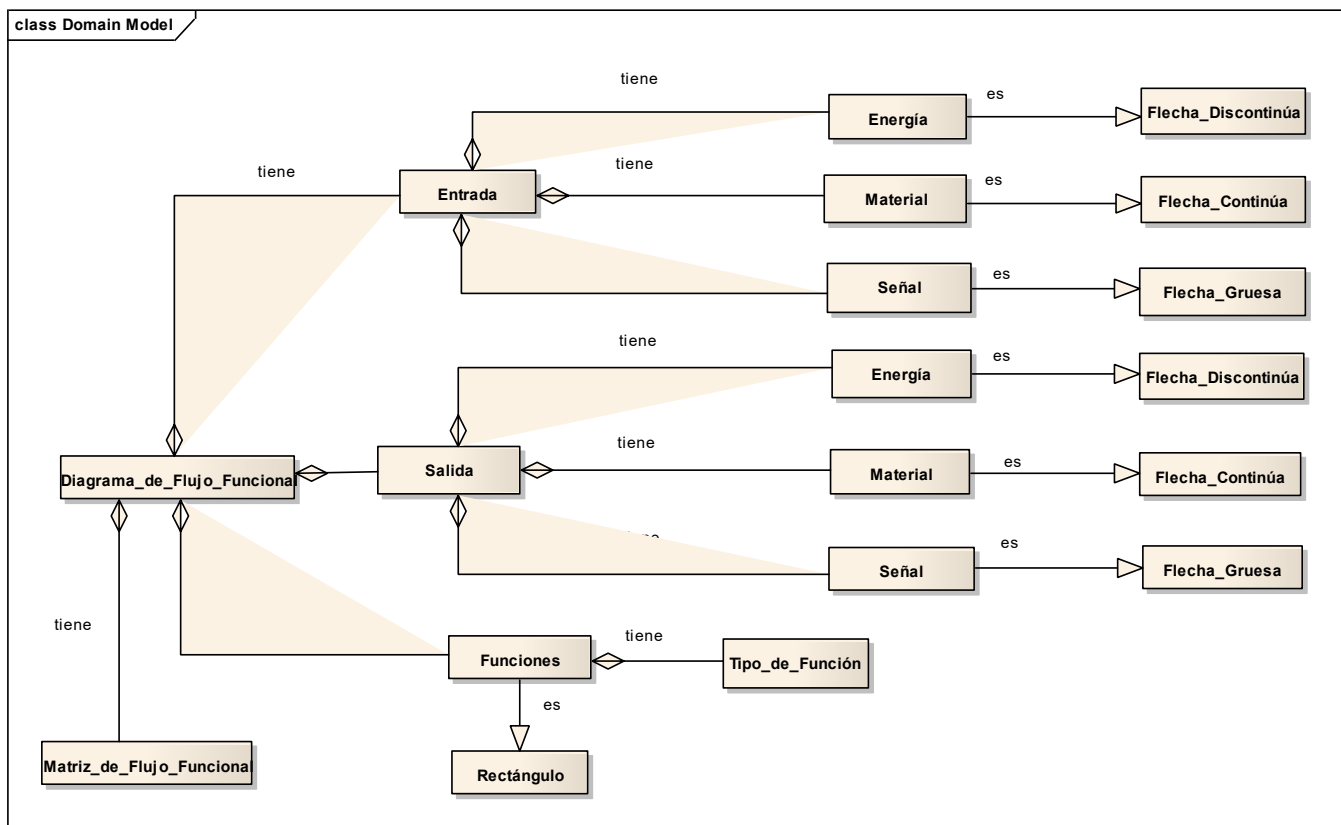
Fuente: Elaboración propia.

A partir de la lista de requerimientos (Tabla 2), se obtienen todos los términos y se elabora una lista ordenada alfabéticamente, eliminando los elementos repetidos. De esta manera se determinan los principales objetos que componen el modelo del dominio con el fin de comprender la problemática en cuestión (Figura 1).

3.2. Análisis y diseño preliminar

El diseño preliminar constituye el paso intermedio entre el análisis y el diseño, y es el que posibilita que se puedan entender por completo los requerimientos, refinando y eliminando la ambigüedad de los mismos, a través del vínculo existente entre los casos de uso y los objetos del modelo del dominio (Rosenberg, Stephens, & Cope, 2005).

Figura 1. Modelo del dominio



Fuente: Elaboración propia.

Dentro del proceso de ICONIX, uno de los primeros pasos, involucra la construcción del modelo de casos de uso. La representación de cada caso de uso

facilita especificar la secuencia de acciones que el sistema puede llevar a cabo, interactuando con sus actores, e incluyendo alternativas dentro de la

secuencia.

Conociendo que UML provee el mecanismo de agrupar los casos de uso en grupos, se organizó en un diagrama de paquetes.

Con el fin de detallar las relaciones entre los casos de usos y los actores, se realizaron las descripciones textuales de los casos de uso como un flujo principal de acciones, pudiendo contener flujos básicos y alternativos. Se escogieron tres casos de uso pertenecientes a los paquetes ilustrados con anterioridad, para mostrar los pasos que se siguieron según la metodología ICONIX (Tabla 3).

Las interacciones entre los objetos participantes de un caso de uso se ilustran en gráficos a través del diagrama de robustez, que permite analizar el texto narrativo de cada caso de uso e identificar un conjunto inicial de objetos involucrados en cada uno de ellos. En la Figura 2 se muestra un diagrama de robustez del sistema para la creación de diagramas de flujo funcional.

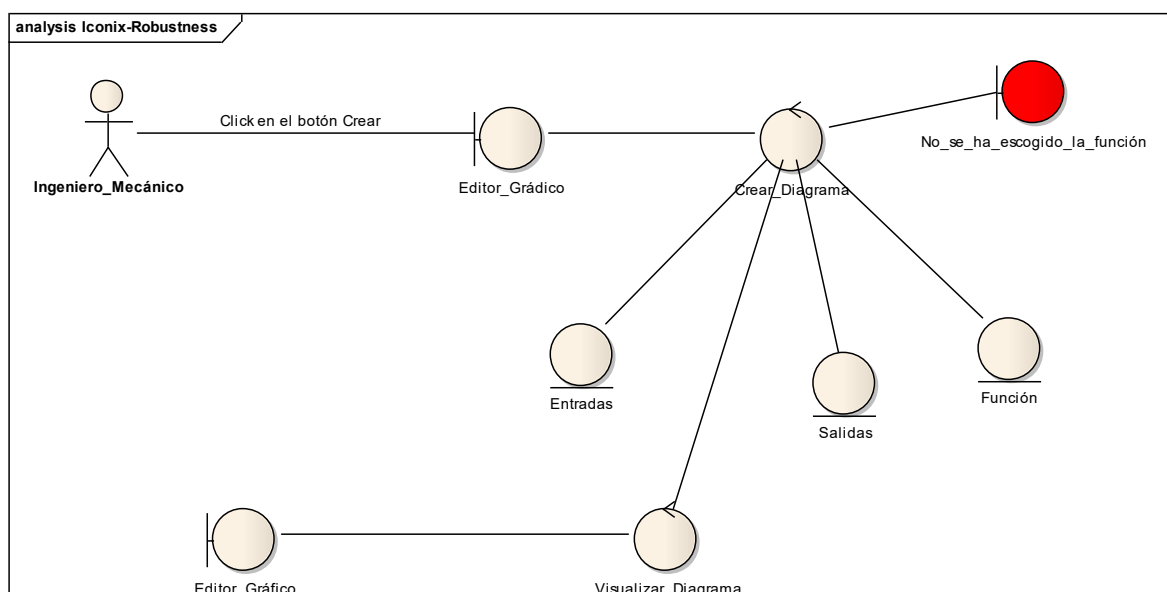
Tabla 3. Descripción de los tres casos de usos

Nombre del Caso de Uso	Descripción
Crear Diagrama Funcional	Curso básico: El usuario da click en el menú nuevo y selecciona en la barra de tareas la opción función. El usuario arrastra esa función hacia el área de trabajo o da doble click en la función creándose una nueva, posiciona el cursor sobre ella y la arrastra

	hasta donde uno desee ubicarla. El usuario selecciona de la barra de tarea la opción material, señal o energía, da click sobre una función inicial y da click sobre una segunda función, y el sistema dibuja la conexión seleccionada entre las dos funciones, creándose el diagrama de flujo funcional. Curso alterno: Verificar conexión: si el usuario intenta crear una conexión de una función inicial a otra, pero por detrás de la segunda función, no lo conecta.
Modificar diagrama funcional	Curso básico: El usuario selecciona el diagrama que desea modificar dando click sobre él. Para modificar cualquier componente del diagrama da doble click en el componente y lo modifica. Curso Alterno: Diagrama sin crear: el usuario no puede modificar ningún diagrama porque no existe.
Visualizar matriz funcional	Curso básico: El usuario da click en el diagrama que desea visualizar la matriz, va a la barra de herramientas y selecciona visualizar matriz, visualizándose de forma automática. Curso Alterno: Diagrama sin seleccionar: el usuario hace clic en el botón visualizar matriz. El sistema no muestra ninguna matriz porque el usuario no ha seleccionado ningún diagrama. Diagrama sin crear: El usuario hace clic en el botón visualizar matriz. El sistema no muestra ninguna matriz porque no hay diagramas creados.

Fuente: Elaboración propia.

Figura 2. Diagrama de Robustez del caso de uso



Fuente: Elaboración propia.

3.3. Arquitectura Técnica

La Arquitectura Técnica se conoce como arquitectura del sistema o arquitectura del software, y se realiza con el objetivo de satisfacer los requerimientos del usuario. Los requerimientos no funcionales definidos para el sistema informático propuesto en la presente investigación son:

- Apariencia o Interfaz externa

El sistema debe ser fácil de manipular. Está concebido para desarrollar un mejor desempeño en la creación de diagramas de flujo funcional.

La interfaz debe ser agradable para conseguir la confianza de los usuarios en la utilización del sistema.

Los colores deben encontrarse en tonalidades suaves y relajantes.

- Portabilidad

Las herramientas utilizadas para el desarrollo del sistema son de software libre y a su vez multiplataforma.

Facilidad de Mantenimiento

El software será capaz de adaptarse a las nuevas características que se les incorporen, garantizando el cambio progresivo.

Ayuda y documentación en línea.

Debe contar con un Manual de ayuda.

- Software

El sistema se desarrollará con el uso del lenguaje de programación Java y el gestor de Bases de Datos SQLite.

La computadora debe tener instalado cualquier versión del NetBeans (entorno de desarrollo integrado libre), Java (JRETM).

- Hardware

Microprocesador Intel Pentium III a 1 GHz de velocidad, con 256 MB de memoria RAM y memoria de video *on board* de 128 MB.

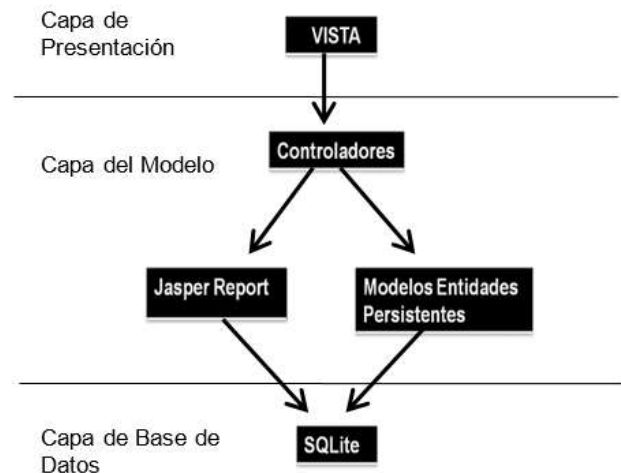
En la Figura 3 se muestran los *softwares* que compondrán el sistema dividido en capas, a través del diagrama de Arquitectura del sistema.

3.4. Diseño detallado

Uno de los pasos fundamentales que propone la metodología ICONIX es el diseño de la aplicación. Este sucede luego de terminado el análisis y captura de los requisitos. En el diseño, se crean los

diagramas de secuencia en los cuales se detalla cómo y a través de qué mensajes interactúan las clases.

Figura 3. Arquitectura del sistema



Fuente: Elaboración propia.

A cada diagrama de robustez le corresponde uno de secuencia. En la Figura 4 se muestra un diagrama de secuencia del sistema informático para la creación de diagramas de flujo funcional.

Crear Diagrama de Flujo Funcional

A partir de la evolución que ha ido sufriendo el modelo de dominio a lo largo de las fases anteriores, se construye el diagrama de clases. Este diagrama permite encontrar errores en los diagramas de secuencia.

El modelo de datos define las estructuras de datos que se utilizan en el sistema y describe la estructura lógica de la información que queda almacenada en la base de datos. Para determinar el modelo lógico de datos se hace necesario primero analizar cuáles son las clases persistentes del sistema informático a desarrollar, por lo que se realiza el diagrama de clases persistentes (Figura 5).

3.5. Modelo de implementación

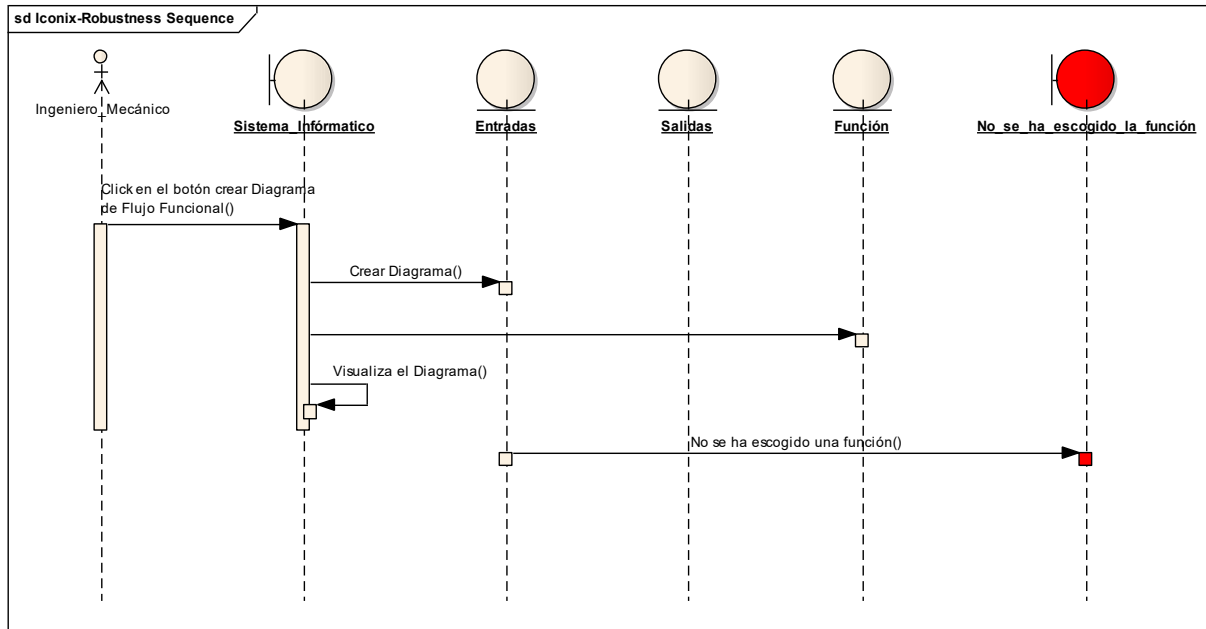
Una vez concluido el diseño detallado del sistema informático propuesto, a través de los diagramas de secuencia de cada caso de uso, se está en condiciones de traducir todo este diseño a código fuente. Esta fase posibilita definir la organización y estructura del código, e implementar las clases que se identificaron durante el diseño.

Un usuario (Ingeniero Mecánico o estudiante) comienza a interactuar con el sistema, una vez que

abre la aplicación (Figura 6). En la Figura 7 se muestra la interfaz Crear Función, donde el usuario inserta las funciones que va a utilizar. En esta interfaz se puede: insertar, modificar, cambiar color, mover y eliminar cualquier función.

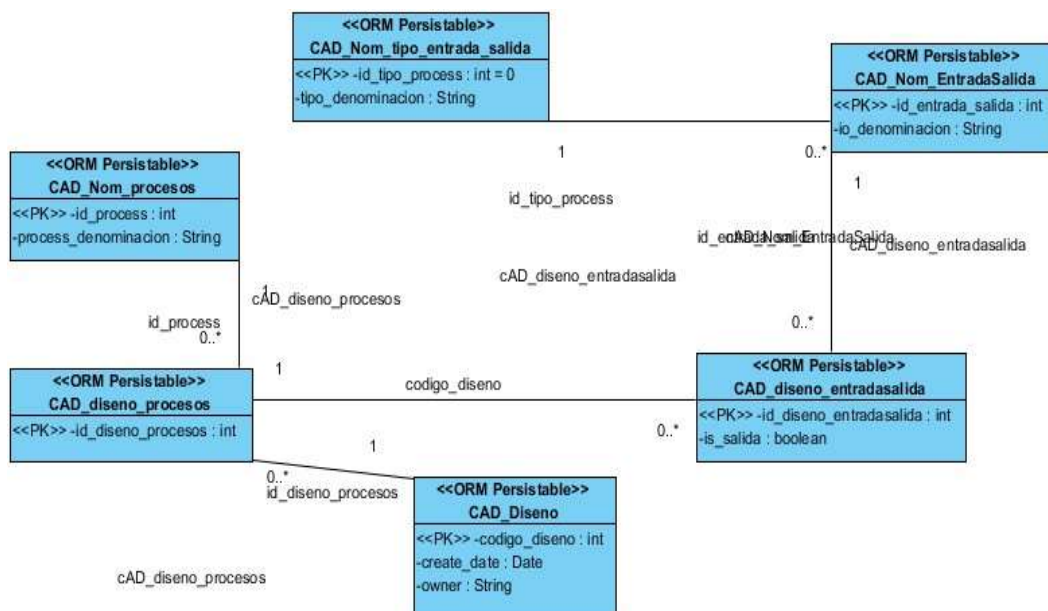
En la Figura 8 se muestra la interfaz Insertar conexiones, donde el usuario inserta las conexiones que va a utilizar. En esta interfaz se puede insertar, modificar, cambiar color y eliminar cualquier conexión seleccionada.

Figura 4. Diagrama de secuencia del caso de uso Crear Diagrama de Flujo Funcional



Fuente: Elaboración propia.

Figura 5. Diagrama de clases persistentes



Fuente: Elaboración propia.

Figura 6. Interfaz de Inicio



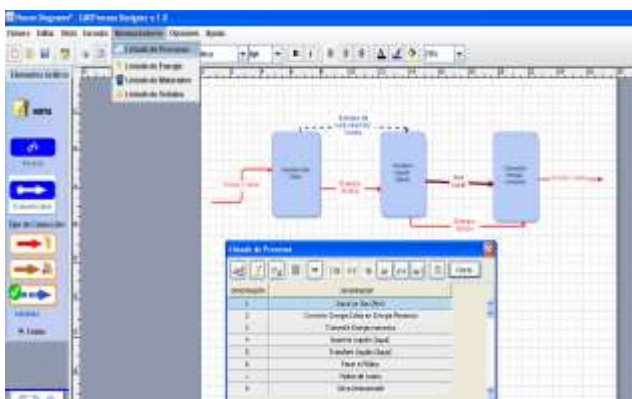
Fuente: Elaboración propia.

Figura 7. Interfaz Insertar Función



Fuente: Elaboración propia.

Figura 8. Interfaz Insertar Conexiones



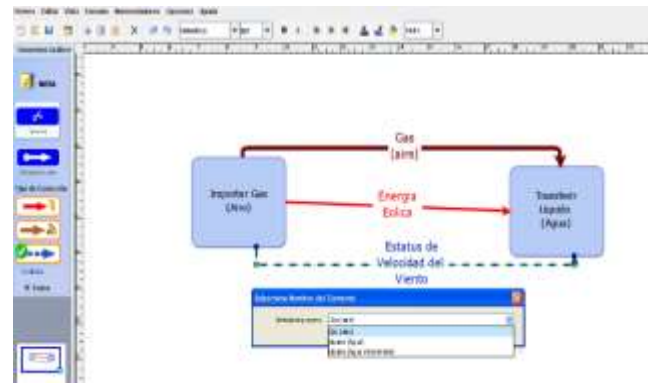
Fuente: Elaboración propia.

En la Figura 9 se ilustra la interfaz de las conexiones, donde el usuario tendrá la posibilidad de escoger la conexión que desee. Se brinda la opción de insertar, modificar y eliminar la o las conexiones.

A lo largo del sistema, se puede percibir el uso homogéneo de todos los elementos de diseño utilizados y los colores. La navegación se destaca

por medio de menús y el procesamiento de información mediante el uso de botones.

Figura 9. Modificar Conexión



Fuente: Elaboración propia.

En la implementación del sistema informático se tuvo en cuenta uno de los factores fundamentales que le proporciona calidad a un software, la facilidad de mantenimiento, el cual puede resultar complejo y costoso.

Se establecieron estándares de codificación que aseguran la legibilidad del código, tanto para los programadores, como para los futuros responsables del mantenimiento del sistema.

4. CONCLUSIONES

La informatización de la modelación funcional favoreció la creación de diagramas de flujo funcional dando cumplimiento al objetivo trazado en la investigación.

El uso de Java como lenguaje de programación y el SQLite como sistema gestor de base de datos, constituyeron una excelente opción para el desarrollo del sistema.

La metodología de desarrollo de software empleada resultó fundamental para maximizar los índices de calidad en los procesos de ingeniería de software implicados.

5. REFERENCIAS BIBLIOGRÁFICAS

Se agradece el apoyo del Centro de Estudios CAD/CAM, a la Facultad de Informática Matemática de la Universidad de Holguín; y al Departamento de Ingeniería Mecánica de la Escola Tècnica Superior d'Enginyers Industrials (ETSEIB) de la Universitat Politècnica de Catalunya en Barcelona (España), en la realización de la investigación conjunta.

6. REFERENCIAS BIBLIOGRÁFICAS

- Dym, C., Little, P., & Orwin's, E. J. (2013). *Engineering Design: A Project-Based Introduction* (4th ed.): Wiley.
- Engelmann, F., Breiing, A., & Gutowski, T. (2021). Engineering Design. In K. H. Grote & H. Hefazi (Eds.), *Handbook of Mechanical Engineering*: Springer, Cham.
- Feldhusen, J., & Karl-Heinrich, G. (2013). *Pahl/Beitz Konstruktionslehre*. Berlin, Heidelberg: Springer Vieweg.
- Gyung-Jin, P. (2007). *Analytic Methods for Design Practice*. London, England: Springer.
- Huang, S., Muci-Kuchler, K. H., & Bedillion, M. D. (2015). Systems thinking skills of undergraduate engineering students. IEEE, 1-5.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo del Software*: Addison-Wesley.
- Mistree, F., Smith, W. F., Bras, B., Allen, J. K., & Muster, d. (1990). *Decision-Based design: a contemporary paradigm for Ship Design*. Paper presented at the THE 98th Annual meeting of SNAME, San Francisco, California.
- Rosenberg, D., Stephens, M., & Cope, M. C. (2005). *Agile Development with ICONIX Process—People, Process, and Pragmatism*: Apress.
- Rosenberg, D., & Scott, K. (2007). *Use Case Driven Object Modeling with UML. Theory and Practice*: Apress.
- Ullman, D. G. (2017). *The mechanical design process* (6th ed.). New York: Wiley.
- Wikipedia. (2021a). Apache OpenOffice Draw. Recuperado el 06-09-2021, 2021, desde https://es.wikipedia.org/w/index.php?title=Apache_OpenOffice_Draw&oldid=119487447
- Wikipedia. (2021b). Dia. Recuperado el 06-09-2021, 2021, desde [https://es.wikipedia.org/w/index.php?title=Dia_\(programa\)&oldid=134267768](https://es.wikipedia.org/w/index.php?title=Dia_(programa)&oldid=134267768)
- Wikipedia. (2021c). Microsoft Visio. Recuperado el 06-09-2021, 2021, desde https://es.wikipedia.org/w/index.php?title=Microsoft_Visio&oldid=132763968