

DOI: https://doi.org/10.37811/cl_rcm.v7i1.4979

Diseño del sistema web SIMEFF para la gestión de presupuestos sanitarios de acuicultura

Karina De La Cruz Silva

MM21320006@acapulco.tecnm.mx
<https://orcid.org/0000-0001-8556-0880>

José Francisco Gazga Portillo

Jose.gp@acapulco.tecnm.mx
<https://orcid.org/0000-0002-5629-3020>

Alma Delia De Jesús Islao

Alma.di@acapulco.tecnm.mx
<https://orcid.org/0000-0002-1725-9145>

Francisco Javier Gutiérrez Mata

Francisco.gm@acapulco.tecnm.mx
<https://orcid.org/0000-0002-1753-5701>

Departamento de Posgrado e Investigación del TecNM campus Acapulco
Acapulco de Juárez, Guerrero, México.

RESUMEN

El presente trabajo tiene por objetivo mostrar el diseño del sistema web SIMEFF para automatizar la gestión de los presupuestos anuales en materia de sanidad e inocuidad acuícola del Comité de Sanidad Acuícola del Estado de Guerrero (COSAEG, A.C.). El diseño del sistema se realiza durante la segunda etapa de la metodología de desarrollo de software en cascada. Se hace mención, que el presente trabajo da continuidad al artículo titulado Modelo de datos para gestionar presupuestos sanitarios de acuicultura (De La Cruz Silva et al., 2022). El diseño se muestra mediante la notación UML y extensiones para modelado web WAE (Web Application Extension), en el cual se representa la arquitectura del sistema de forma global mediante diagramas de alto nivel de abstracción, mostrando los objetos fundamentales y su interacción de acuerdo con los requerimientos que describen la funcionalidad a desarrollar de acuerdo con los procesos de gestión de los presupuestos anuales que realiza el COSAEG, A.C.

Palabras clave: *diseño web; metodología cascada; sanidad acuícola; WAE; COSAEG*

Correspondencia: MM21320006@acapulco.tecnm.mx

Artículo recibido 25 enero 2023 Aceptado para publicación: 25 febrero 2023

Conflictos de Interés: Ninguna que declarar

Todo el contenido de **Ciencia Latina Revista Científica Multidisciplinar**, publicados en este sitio están disponibles bajo

Licencia [Creative Commons](https://creativecommons.org/licenses/by/4.0/) 

Cómo citar: De La Cruz Silva, K., Gazga Portillo, J. F., De Jesús Islao, A. D., & Gutiérrez Mata, F. J. (2023). Diseño del sistema web SIMEFF para la gestión de presupuestos sanitarios de acuicultura. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1), 7463-7479. https://doi.org/10.37811/cl_rcm.v7i1.4979

Design of the SIMEFF web system for the management of aquaculture sanitary budgets

ABSTRACT

The present work aims to show the design of the SIMEFF web system to automate the management of annual budgets in the field of aquaculture health and safety of the Aquaculture Health Committee of the State of Guerrero (COSAEG, A.C.). System design is done during the second stage of the waterfall software development methodology. It is mentioned that this work gives continuity to the article entitled Data model to manage aquaculture health budgets [5]. The design is shown by UML notation and extensions for web modeling WAE (Web Application Extension), in which the architecture of the system is represented globally through diagrams of high level of abstraction, showing the fundamental objects and their interaction according to the requirements that describe the functionality to be developed according to the management processes of the annual budgets carried out by the COSAEG, A.C.

Keywords: *web design; waterfall methodology; aquaculture health; WAE; COSAEG*

INTRODUCCIÓN

De acuerdo con (Pressman, 2002), el diseño es una representación significativa de ingeniería de algo que se va a construir. Se realiza basándose en los requisitos que otorga el cliente. Para realizar el diseño del sistema SIMEFF, se consideran los requisitos definidos en el primer artículo realizado, titulado “Modelo de datos para gestionar presupuestos sanitarios de acuicultura” (De La Cruz Silva et al., 2022). A partir de lo anterior, se definen los requisitos funcionales, dentro de los cuales, los principales son: el soporte y gestión de la información de los presupuestos anuales, soporte y gestión de los proyectos de trabajo anuales, soporte y gestión de las metas físicas y financieras anuales de cada uno de los proyectos anuales y los desagregados mensuales de metas físicas y financieras, asignación de suficiencia presupuestal, así como de las acciones y conceptos de gastos específicos, los registros de avances y el informe de avance mensual; que en conjunto con los requisitos no funcionales forman la base para dar continuidad con el desarrollo de la funcionalidad de los procesos asociados a la gestión de presupuestos sanitarios de acuicultura que realiza el COSAEG, A.C., así mismo, para el desarrollo del diseño arquitectónico del sistema para el cual se utilizan tecnologías web que permiten el cumplimiento de los requisitos específicos del sistema SIMEFF.

Por otra parte, de acuerdo con (Polo García, 2017), el presupuesto como mecanismo de control, permite realizar correctivos para mejorar la gestión de los presupuestos públicos, dicha gestión implica dirigir, planificar, organizar, suministrar, evaluar y administrar cada uno de los ingresos y egresos que se realizan diariamente, en este caso, el organismo registra cada movimiento de manera contable utilizando un sistema de contabilidad y simples hojas de cálculo. Tal como se describe en el primer trabajo sobre el Modelo de datos del sistema SIMEFF (De La Cruz Silva et al., 2022), utilizar hojas de cálculo para generar la información sobre las metas programadas en los proyectos de trabajo y la ejecución de las mismas, así como, el registro de la información de gastos previamente contabilizados en los informes de avances de metas mensuales que realiza el organismo, estos muestran la información de las metas físicas y financieras programadas y ejecutadas de forma inconsistente, no unificada, archivos duplicados y no se permite compartir los archivos de forma simultánea entre las personas interesadas, donde la ubicuidad es una característica no presente al utilizar estos métodos no automatizados, siendo este, un mecanismo importante para realizar la gestión de los programas de trabajo y tomar

decisiones oportunas y asertivas en el manejo del presupuesto. Por lo tanto, para el desarrollo del sistema web SIMEFF y así mismo automatizar los procesos de gestión del presupuesto público asignados a los proyectos de trabajo, se requiere la aplicación de una metodología de ciclo de vida de software, en este caso, se aplica la metodología en cascada, por lo que, el desarrollo del presente trabajo está enfocado en la segunda etapa de esta metodología, la cual consiste en el diseño del sistema (Pressman, 2002), esta se aborda en la sección de metodología del presente trabajo.

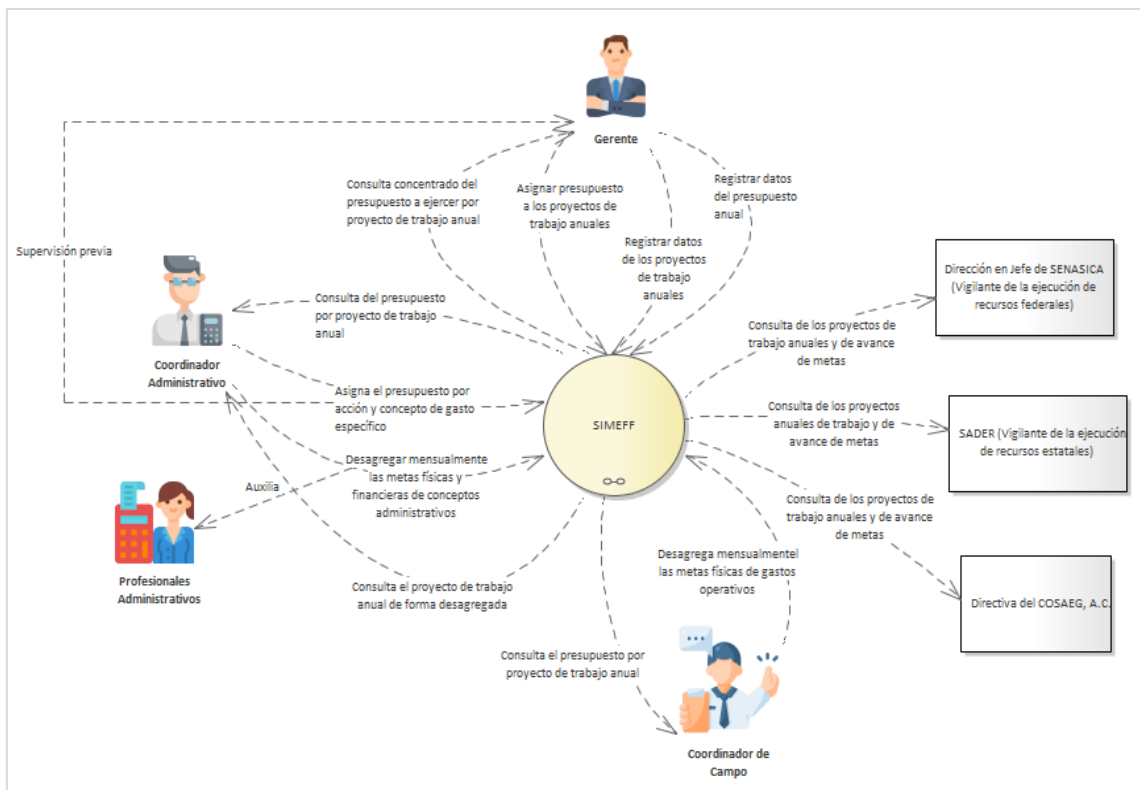
Ante la problemática descrita, el desarrollo de un sistema implementando tecnologías web para el COSAEG, A.C., contribuye a que se reduzca la cantidad de formatos y la multiplicidad de datos, así mismo, que sea más sencillo y eficiente al automatizar la generación y procesamiento de la información financiera relacionada a los proyectos de trabajo. Aunado a esto, facilita el acceso a la información financiera desde cualquier lugar donde se requiera consultar, permite el acceso recurrente de los usuarios, y la consulta de la información, de tal forma que siempre se encuentre disponible.

Marco Referencial

De acuerdo con (Alonso Amo et al., 2005) el diseño es el primer paso en la fase de desarrollo de cualquier producto o sistema de ingeniería. Desde el contexto de la ingeniería del software (Pressman, 2002), el diseño se centra en cuatro áreas de interés importantes: datos, arquitectura, interfaces y componentes. El área de interés concerniente a los datos se representó en el modelo de datos del trabajo anterior (De La Cruz Silva et al., 2022), en el cual se muestra la estructura de los datos mediante el diseño de la base de datos relacional del sistema SIMEFF.

Para proporcionar una visión general de los procesos que se llevan a cabo para la gestión de los proyectos de trabajo, se muestra en la Figura 1, el diagrama de contexto del sistema web SIMEFF, en el cual se identifican 7 entidades que interactúan entorno al sistema web, los cuales son: Gerente, Coordinador Administrativo, Profesionales Administrativos, Coordinador de Campo, la directiva del organismo y las dos entidades gubernamentales encargadas de la supervisión de la correcta ejecución de los proyectos de trabajo anuales.

Figura 1: Diagrama de contexto del sistema web SIMEFF



Fuente: Elaboración propia. Realizado con Enterprise Architect, versión: 15.0.1509, versión UML 2.1.

El diagrama de contexto anterior, además de mostrar las entidades y usuarios que interactúan con el sistema, describe en que consiste la interacción que se lleva a cabo con cada uno de ellos.

Para el diseño del sistema SIMEFF se considera el diagrama de contexto anterior, en conjunto con los requisitos del sistema, como lo son los requisitos no funcionales, los cuales especifican los atributos de calidad del producto software.

El atributo calidad

El concepto calidad, es uno de los más importantes a tomar en cuenta durante el diseño del sistema y de cualquier producto, este concepto se debe tener presente durante todo el ciclo de vida de software, por lo que es relevante durante el diseño del software.

El diseño del software traduce los requisitos a una forma gráfica, mediante la notación UML, extensiones WAE u otro tipo de diagramas que reflejen los atributos de diseño del software. Por lo que, se sugieren tres características que se utilizan como una guía para evaluar la calidad del diseño (Pressman, 2002), estas son:

1. El diseño debe implementar absolutamente todos los requisitos que se hayan definido en el modelo de análisis.

2. El diseño debe ser una guía para quienes generan código y para quienes se encargan de comprobar y dar soporte al software.

3. El diseño debe proporcionar la imagen global y completa del software.

Además de lo anterior, se establecen criterios técnicos para evaluar la calidad del diseño.

Algunas de las directrices más importantes para tomar en cuenta son las siguientes:

1. Un diseño debe presentar una estructura arquitectónica mediante patrones de diseño reconocibles.

2. Un diseño debe ser modular, es decir, el software deberá dividirse de forma lógica en elementos funcionales y subfuncionales específicos.

3. Un diseño deberá contener distintas representaciones de datos, arquitectura, interfaces y componentes (módulos).

Según la primer directriz, para presentar una estructura arquitectónica se eligen patrones de diseño, que de acuerdo con Sommerville (Sommerville, 2011), los patrones de diseño arquitectónicos se consideran como la descripción abstracta estilizada de buena práctica.

Entre los patrones de diseño arquitectónico vigentes y más utilizados, se encuentran los siguientes:

- Arquitectura MVC
- Arquitectura en capas
- Arquitectura cliente – servidor

Con el propósito de establecer una arquitectura de diseño utilizando los patrones de diseño antes mencionados, se muestra a continuación, en la Tabla 1, la relación entre los requerimientos no funcionales más comunes (RNFC) con los requerimientos no funcionales (RNF) del sistema web SIMEFF y el patrón de diseño arquitectónico adecuado para estos, siendo el RNFC2, el cual está asociado con el RNF4, y el RNFC6 que está asociado con el RNF7 y el RNF9, no tienen relación con un patrón arquitectónico adecuado.

Los requisitos no funcionales son los que muestran las restricciones con las que debe contar el sistema web SIMEFF y, por lo tanto, también aportan el atributo de calidad al sistema, ocupándose de aquellos atributos a través de los cuales pueden surgir problemas durante el funcionamiento de este.

Tabla 1. *Patrones de arquitectura contra los requisitos no funcionales comunes y los requisitos no funcionales del sistema web SIMEFF.*

ID	Requisitos no funcionales COMUNES	ID	Requisitos no funcionales SIMEFF	Patrón
RNFC1	Disponibilidad	RNF3, RNF5	Disponibilidad del sistema, Servicio de hosting	cliente-servidor
RNFC2	Tiempos de respuesta rápidos (alto rendimiento)	RNF4	Tiempo de respuesta ante actualizaciones	---
RNFC3	Integridad de la información	RNF1, RNF6, RNF8	Registro de usuarios, usuario con acceso a la base de datos relacional	cliente-servidor, MVC
RNFC4	Seguridad	RNF1, RNF6, RNF8	Registro de usuarios, usuario con acceso al servidor, utilizar base de datos relacional	cliente-servidor, capas, MVC
RNFC5	Mantenibilidad	RNF6	usuario con acceso al servidor	capas, MVC
RNFC6	Interoperabilidad	RNF7, RNF9	Manuales de usuario, compatibilidad con navegadores comunes	---

Fuente: *Elaboración propia.*

Por lo anterior, en la Tabla 2, se muestra una breve descripción de los patrones de diseño arquitectónico a ser considerados.

El patrón de diseño MVC, proporciona diversas formas de visualizar e interactuar con los datos, por lo que para el diseño del sistema web SIMEFF se toma como referencia principal este patrón de diseño, sin embargo, es posible aplicar las características de los patrones cliente-servidor y el de capas.

Tabla 2. *Principales patrones de diseño arquitectónico.*

Arquitectura	Descripción breve
MVC (Modelo-Vista-Controlador)	Se encarga de separar la presentación e interacción de los datos del sistema. Utiliza tres componentes lógicos: modelo, vista y controlador.
Cliente-Servidor	La funcionalidad se organiza en servicios y los clientes son usuarios de estos servicios. El servidor se encarga de responder a sus peticiones.
Capas	Es un modelo que se encarga de organizar el sistema en forma de capas de tal forma que su funcionalidad se relaciona con cada una de ellas.

Fuente: *Elaboración propia.*

Del patrón de diseño MVC se considera la lógica que implementa, al utilizar los componentes lógicos: presentación, controlador y modelo, lo que aporta seguridad al ser atendidas las peticiones del usuario por medio del controlador y no directamente de la base de datos. Del patrón cliente-servidor se considera la disponibilidad de la funcionalidad del sistema, la cual estaría disponible por todos los clientes ya que la base de datos estaría disponible de forma compartida, alojada en el servidor de BD. El patrón de diseño por capas permite aplicar el principio de separación de preocupaciones (SoC) (Blancarte Iturralde, 2020), que consiste que cada capa tenga un rol definido, como el front-end y back-end.

Además de lo anterior, se considera, por ser este sistema de tipo web, tecnologías de desarrollo primordiales, como son: HTML, CSS y JavaScript, las cuales son tecnologías que intervienen directamente en el desarrollo web (Rubiales Gómez, 2021).

Para realizar el diseño del sistema SIMEFF, se toman como referencia, herramientas comerciales que se encargan de gestionar presupuestos, los cuales, por ser de tipo comercial, cuentan con una amplia gama de módulos, abarcando gran parte de las áreas que maneja una empresa, estos productos, son realizados para la planificación de recursos empresariales, mejor conocidas como ERP. Entre estas se encuentran Dynamics 365 Project Operations (Microsoft, 2023), un software comercial realizado por Microsoft, el cual abarca diferentes áreas de operación de toda una empresa, como son: el área financiera, contable, ventas, servicios, recursos humanos, entre otros. Así mismo, la aplicación Odoo (ODOO, 2023), es un software comercial del mismo tipo, ERP, el cual se denomina así mismo como Open Source ERP, este cuenta con los mismos módulos que Dynamics 365, y además con el módulo de Productividad y MRP (Gestión de cambios).

De acuerdo con (Oracle, 2023), la planificación de recursos empresariales (ERP), es la solución líder en el sector, la cual se usa para gestionar de forma integral procesos y aplicaciones de negocio, el fin es, ganar resiliencia y agilidad en tiempo real y posicionarse para el crecimiento.

Sin embargo, este tipo de aplicaciones no se ajustan a los requisitos específicos que el COSAEG necesita para automatizar sus procesos de gestión.

Otras aplicaciones que sirven como investigaciones previas al desarrollo del sistema, y que se encargan del manejo de presupuestos, son los siguientes:

Tabla 3: Comparativa de proyectos de aplicaciones para manejo de presupuestos.

#	Proyecto	Tipo	Metodología	Herramientas tecnológicas	Arquitectura	Arquitectura lógica de software	Funcionalidades principales
1	Monitoreo y seguimiento presupuestal para control de gastos del programa Paz y Región de la Universidad de Ibagué, Colombia.		Cualitativa bajo un enfoque de estudio de caso	Microsoft Access	Uso local	Monolítica	Monitoreo y seguimiento presupuestal mediante la creación de una base de datos con módulos específicos.
2	Sistema web para la administración de eventos y control de gastos para la empresa "GM"	Web	Cascada	MySQL, PHP, HTML, CSS, JQuery, Ajax, JavaScript, XAMPP	Cliente - Servidor		Administra los presupuestos para eventos y el control de gastos.
3	Sistema de información basado en tecnología web mejorar el control de Gastos e Ingresos en la empresa Omega S.A.C	Web	Iconix (PU y XP)	PHP y MySQL	Cliente - Servidor	MVC (Modelo-Vista-Controlador)	Control de gastos e ingresos.

Fuente: Elaboración propia. Información extraída de: (Baron, 2019), (Mederos & Ylam, 2020) y (Torres Solis, 2021)

Dentro de las ventajas de las aplicaciones web, se destacan las siguientes: accesibilidad al sistema de manera local y remota, acceso de múltiples usuarios de forma recurrente, son dinámicos, portables, permiten compartir información en un formato homogéneo y unificado para los usuarios que registran, actualizan o consultan la información contenida en este tipo de aplicaciones. Estas ventajas muestran que utilizar tecnologías web para automatizar los procesos de gestión de proyectos que realiza el COSAEG, son fundamentales para el diseño del sistema SIMEFF.

A partir de lo anterior descrito, se describe el objetivo general del presente trabajo.

Objetivo general

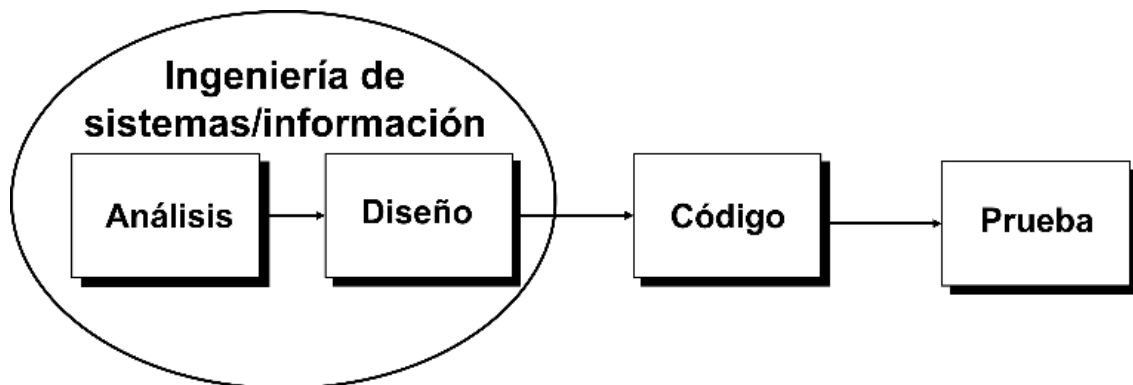
- Consiste en tomar como base los requerimientos establecidos por el COSAEG, A.C., para realizar el diseño del sistema web SIMEFF utilizando la notación UML y extensiones para modelado web (WAE), a fin de proporcionar el plano preliminar para el desarrollo del sistema encargado de la gestión de presupuestos asignados a los proyectos anuales de sanidad e inocuidad acuícola.

A continuación, se describe la metodología de desarrollo de software que se utiliza para el desarrollo del diseño del sistema web SIMEFF.

METODOLOGÍA

El desarrollo del sistema web SIMEFF, se realiza siguiendo la metodología en cascada. De acuerdo con (Pressman, 2002), se conforma de las siguientes etapas, las cuales se muestran en la Figura 2.

Figura 2: El modelo lineal secuencial



Fuente: (Pressman, 2002)

Se eligió esta metodología debido a que los requerimientos del sistema están completamente definidos y las partes interesadas están de acuerdo en evitar cualquier cambio radical durante el desarrollo del sistema.

De acuerdo con (Leach, 2016), es uno de los modelos más comunes, y esencialmente nunca se aplica en situaciones donde no hay un cliente conocido, como sería el caso del software diseñado para el mercado comercial para múltiples compradores minoristas.

Vinculado a lo anterior y como parte del diseño del sistema, es necesario aplicar una metodología de modelado web, como las extensiones de aplicaciones web (WAE) que proporciona UML y que representa el flujo de procesamiento del sistema en un ambiente web.

UML y extensiones de aplicaciones web (WAE)

El lenguaje de modelado unificado (UML, por sus siglas en inglés) (Casas Roma & Conesa i Coralt, 2013), es un lenguaje que comúnmente se utiliza para realizar el modelo de un sistema mediante la representación visual de los datos, entidades, componentes, conceptos, u objetos que permitan especificar, construir y documentar los artefactos de un sistema software (Rumbaugh et al., 1999). UML proporciona extensiones para modelado web, mejor conocidos como extensiones de aplicaciones web, por sus siglas en inglés WAE, (Web Application Extension) (Conallen, 2000). Dichas extensiones permiten el modelado web de manera formal mediante elementos que describen la interacción de componentes del sistema en un ambiente de tipo web. Estos elementos son: estereotipos, valores etiquetados y restricciones (Conallen, 2000). Entre los estereotipos más comunes se encuentran: server page, client page, form y frameset, ver Figura 3; estos se relacionan con las líneas: link: enlace puntero de una página cliente a

otra página, build: Une el vacío entre las páginas cliente y de servidor, redirect: asociación unidireccional con otra página web, dependency: indica dependencia entre una página y un frame o form, y submit: asociación entre un form y una page server (Conallen, 2000).

Figura 3: Representación en la aplicación Enterprise Architect de estereotipos comunes de las WAE.



Fuente: Obtenido de Enterprise Architect, UML version 2.1, Sparx Systems (Sparx Systems Pty Ltd., 2022).

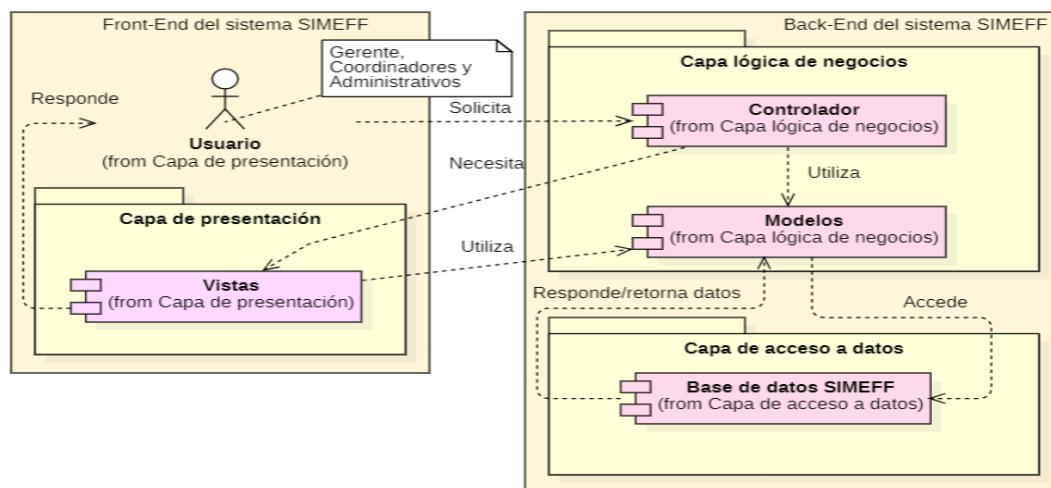
RESULTADOS Y DISCUSIÓN

Derivado de lo anterior, se muestra el resultado del diseño del sistema web SIMEFF, el cual se representa mediante los diagramas UML de la arquitectura del sistema y despliegue, además del modelo web utilizando las WAE.

Arquitectura del sistema web SIMEFF

De acuerdo con las Tablas 1 y 2, se determinó utilizar el patrón de diseño que cubra con los requerimientos funcionales y no funcionales de éste. El sistema web SIMEFF, está diseñado con base al patrón Modelo-Vista-Controlador o por sus siglas en inglés como Model-View-Controller, mejor conocido como MVC (Sommerville, 2011), ver Figura 4.

Figura 4: Arquitectura del sistema web SIMEFF utilizando el patrón de diseño arquitectónico MVC.



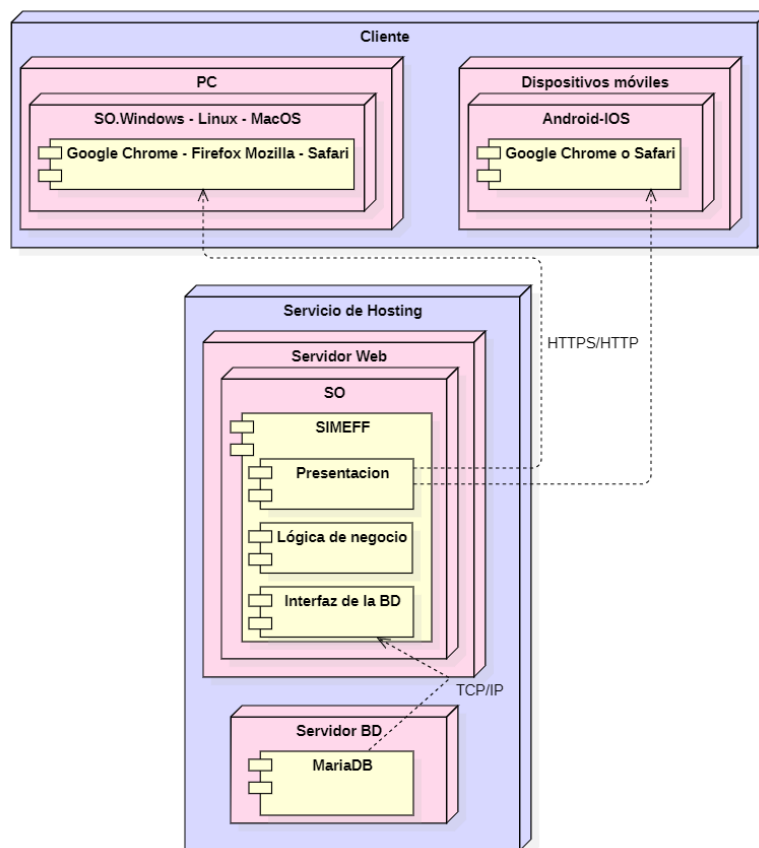
Fuente: Elaboración propia en la aplicación StarUML (MKLabs Co., Ltd, 2023).

En la Figura 3, se observa que en la parte del Front-End, es decir, del lado del cliente, la figura usuario representa los usuarios considerados en el sistema SIMEFF, los cuales son: Gerente, Coordinadores o personal administrativo del organismo, los cuales interactúan con las vistas que conforman la capa de presentación, a través de la cual se envía una solicitud por parte del usuario, esta solicitud se recibe en la parte del Back-End, es decir, del lado del servidor, ésta solicitud es atendida por el controlador en la capa correspondiente a la lógica de negocios; el controlador utiliza los modelos para acceder a los datos solicitados por el usuario, por lo que, se encarga de acceder a la base de datos que se encuentra en la capa de acceso a datos; de la capa de acceso a datos se envía la respuesta mediante un retorno de datos de regreso al modelo; de esta forma, las vistas utilizan los modelos para la presentación de los datos ante el usuario.

Diagrama de despliegue

Por otro lado, el despliegue del sistema web SIMEFF, representa la distribución estática del sistema, de cada uno de los componentes del software que se encuentran en los distintos nodos que conforman la red física se puede visualizar en la Figura 5.

Figura 5: Diagrama de despliegue del sistema SIMEFF.



Fuente: Elaboración propia. Realizado con (MKLabs Co., Ltd, 2023).

PC: Representa la computadora de escritorio a través de la cual, por medio de un navegador de internet (Google Chrome, Mozilla Firefox, Safari), el sistema se podrá utilizar por los usuarios, también conocido como cliente.

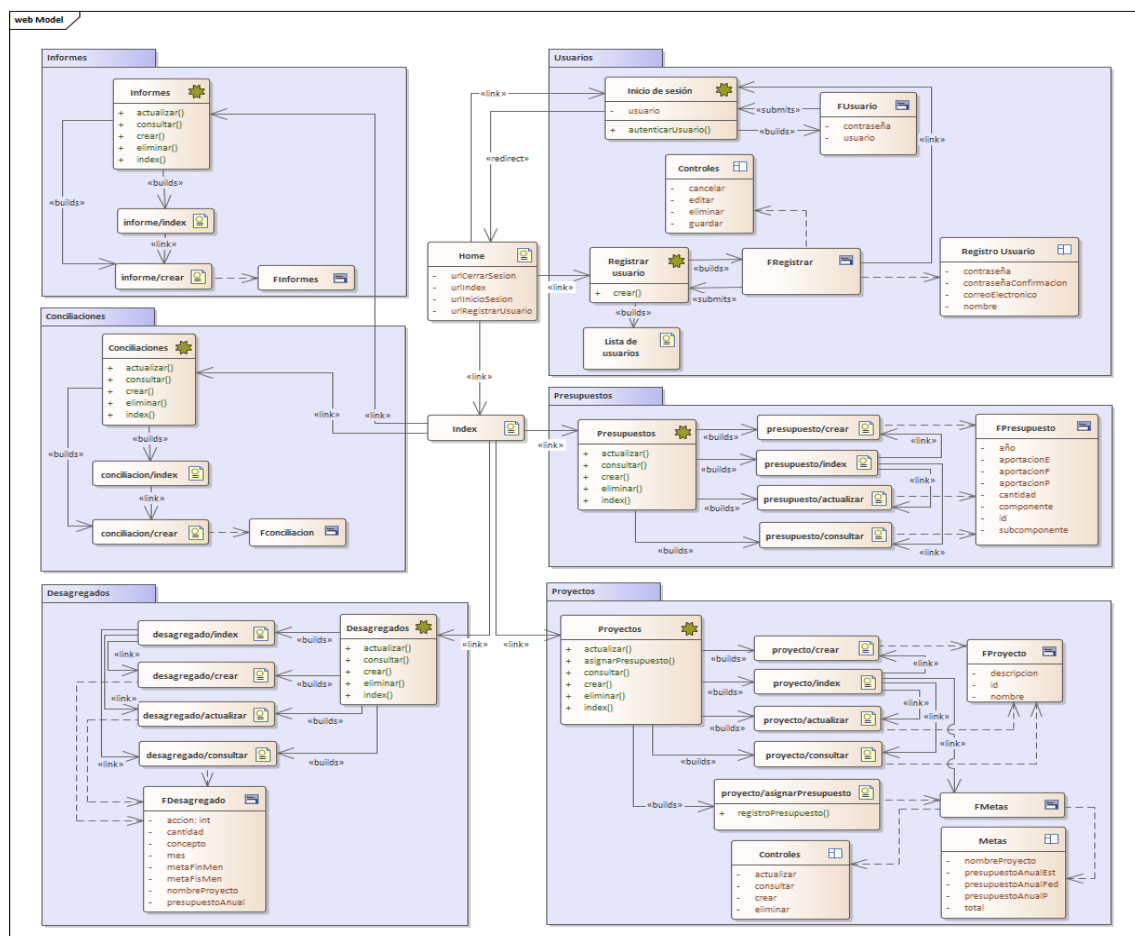
Dispositivos móviles: Representa dispositivos de tipo portátil a través del cual, por medio de un navegador de internet (Google Chrome o Safari), el sistema se podrá utilizar por los usuarios.

Servicio de Hosting: Proveedor del servicio de alojamiento del sistema. El cual consta del Servidor Web, en el cual se encontrarán la aplicación y sus componentes; y el Servidor de BD, donde se proveerá el servicio de la base de datos con MariaDB, para su administración.

Modelo de desarrollo web utilizando extensiones WAE.

El modelo web del sistema SIMEFF, muestra la estructura navegacional utilizando las extensiones de aplicaciones web WAE, realizado con la aplicación Enterprise Architect, ver Figura 6.

Figura 6: Modelo web utilizando extensiones web (WAE).



Fuente: Elaboración propia realizada con la aplicación Enterprise Architect, versión: 15.0.1509, versión UML: 2.1

Se observan las extensiones web mediante los estereotipos server pages, éstas actúan como el componente lógico controlador del patrón MVC. De la misma forma, las client pages actúan como el componente lógico de vistas y las extensiones form y frameset actúan como parte del componente lógico de modelo. Así mismo, se divide en secciones que indican de manera general los tipos de datos que se manejan. Estas secciones son:

Usuarios

Se muestra el flujo navegacional para el acceso de usuarios al sistema, en el cual solo el Gerente podrá realizar el registro de los usuarios.

Presupuestos

Se muestra el flujo navegacional para el registro de los datos concernientes a metas financieras anuales, tipos de aportaciones, año del ejercicio, así como de los componentes y subcomponentes a cuál pertenece el presupuesto.

Proyectos

Se muestra el flujo navegacional para el registro de los datos de los proyectos trabajo incluyendo la asignación del presupuesto, es decir, se debe registrar previamente la suficiencia presupuestaria para poder asignar el presupuesto a cada uno de los proyectos de trabajo.

Desagregados

Se muestra el flujo navegacional para registrar la distribución por mes de las metas físicas y financieras de los proyectos de trabajo, comprendido anualmente.

Conciliaciones

Muestra el flujo navegacional para crear el comparativo de lo programado en los proyectos de trabajo contra los movimientos bancarios realizados.

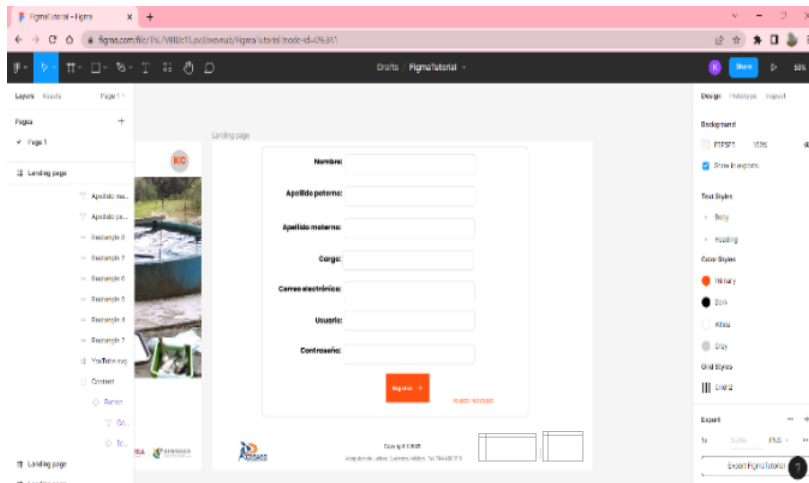
Informes

Muestra el flujo navegacional para crear los registros de avances de las metas físicas y financieras de los proyectos de trabajo, y generar los informes mensuales de cada uno de estos con el fin de verificar que se estén ejecutando las metas conforme a lo programado y obtener indicadores de avance.

Maquetación prototipo

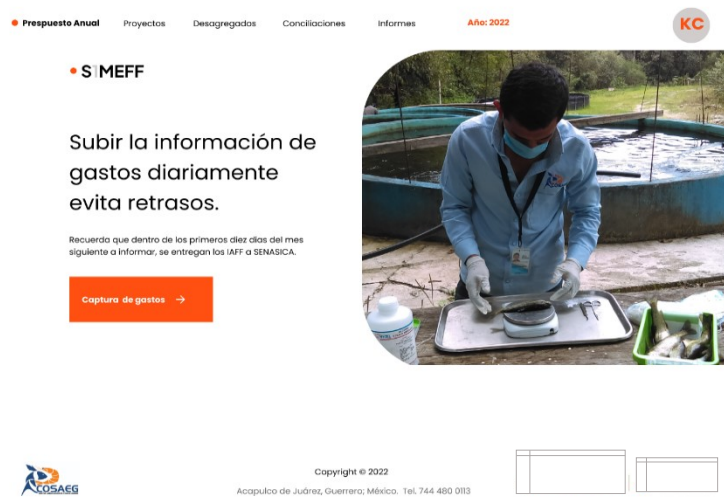
Finalmente, se muestran algunas de las vistas realizadas con la aplicación Figma (Cirujano & Moratinos, 2020), que permite realizar el prototipado de sistemas web, Figura 7, 8 y 9.

Figura 7: Diseño del formulario de registro de usuarios para el sistema SIMEFF.



Fuente: Elaboración propia.

Figura 8: Diseño de la vista principal del sistema SIMEFF.



Fuente: Elaboración propia

Figura 9: Diseño de la vista del presupuesto anual del sistema SIMEFF.



Fuente: Elaboración propia

El prototipo de la interfaz gráfica, representa la presentación del sistema a los usuarios, siendo esta la última parte del diseño del sistema SIMEFF.

CONCLUSIONES

Tomando como punto de partida los requerimientos del organismo, el diagrama de contexto y los antecedentes de desarrollo de proyectos similares, se obtiene el diseño del sistema SIMEFF el cual provee de forma sintetizada los datos relacionados con los presupuestos y proyectos de trabajo del organismo, permitiendo obtener una visión global de cómo es que está conformado el sistema, su estructura, componentes e interfaces, así como la secuencia de procesamiento mediante el modelo web.

Así mismo, aplicar en el diseño el patrón de diseño MVC y como complemento el patrón cliente-servidor y de capas, permiten que el diseño provea el desarrollo de un sistema con la característica de escalabilidad, es decir, que sea capaz de evolucionar en trabajos futuros.

Otra característica fundamental es, que la metodología WAE, permite que el diseño del sistema SIMEFF muestre claramente el flujo de navegación, siendo este metodologías un apoyo en la etapa de diseño de la metodología cascada.

Finalmente, el diseño realizado provee la base para la etapa de codificación del sistema, lo que permite ahorro de tiempo y aumento de consistencia en la codificación. Esto, además, beneficia a las siguientes etapas de desarrollo del sistema, como son: el mantenimiento del sistema y su evolución.

LISTA DE REFERENCIAS

- Alonso Amo, F., Martínez Normand, L., & Segovia Pérez, F. J. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones.
- Baron, L. C. (2019). *Sistema de monitoreo y seguimiento presupuestal para el Programa Paz y Región de la Universidad de Ibagué*. <https://repositorio.unibague.edu.co/>
<https://repositorio.unibague.edu.co/handle/20.500.12313/1951>
- Blancarte Iturralde, O. J. (2020). *Introducción a la arquitectura de software. Un enfoque práctico*. Oscar Javier Blancarte Iturralde.
- Casas Roma, J., & Conesa i Coralt, J. (2013). *Diseño conceptual de base de datos UML*. Barcelona, España: Editorial UOC.
- Cirujano, A., & Moratinos, P. (19 de Octubre de 2020). *Qué es figma*. 3ymedia.school:
<https://3ymedia.school/que-es-figma/>

- Conallen, J. (2000). *Building Web Applications with UML*. Addison Wesley.
- De La Cruz Silva, K., Gazga Portillo, J., De Jesús Islao, A., & Gutiérrez Mata, F. (septiembre de 2022). Modelo de datos para gestionar presupuestos sanitarios de acuicultura. *Robótica y Computación una nueva perspectiva*, 51-60/206.
- Leach, R. J. (2016). *Introduction to Software Engineering*. Washington, DC, USA: CRC Press.
- Mederos, M., & Ylam, A. (2020). *Renati*. Renati: <https://renati.sunedu.gob.pe/handle/sunedu/2955924>
- Microsoft. (2023). *dynamics.microsoft.com*. dynamics.microsoft.com: <https://dynamics.microsoft.com/es-mx/erp/erp-software>
- MKLABS Co., Ltd. (2023). *StarUML*. staruml.io: <https://staruml.io/>
- ODOO. (2023). https://www.odoo.com/es_ES. https://www.odoo.com/es_ES.
- Oracle. (2023). *oracle.com*. oracle.com: <https://www.oracle.com/mx/erp/what-is-erp/>
- Polo García, B. (2017). *Manual de Presupuestos Teórico y Práctico*. Bogotá, Colombia: Nueva Legislación SAS.
- Pressman, R. S. (2002). *Ingeniería del Software. Un enfoque práctico* (Quinta ed.). Aravaca, Madrid: McGraw-Hill/Interamericana de España, S.A.U.
- Rubiales Gómez, M. (2021). *Curso de Desarrollo Web. HTML, CSS y JavaScript*. Anaya Multimedia.
- Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *El lenguaje unificado de modelado. Manual de referencia*. Pearson Educación, S.A.
- Sommerville, I. (2011). *Ingeniería de software*. México: Pearson Education, Inc.
- Sparx Systems Pty Ltd. (2022). *sparxsystems*. sparxsystems: <https://sparxsystems.com/products/ea/>
- Torres Solis, E. V. (2021). *181.198.35.98*. http://181.198.35.98/Archivos/TORRES%20SOLIS%20EVELYN%20VANESSA_compressed.pdf