



UNIVERSIDAD
**SAN IGNACIO
DE LOYOLA**

FACULTAD DE INGENIERÍA

Ingeniería de Sistemas y de Software

Curso : Programación Orientada a Objetos II

Proyecto: “Tres Marías Restaurant”

Proyecto Final

Bloque: FC-PREISF03A1M(H)

BRAVO OCAMPO, DAVID AQUILES

CHUMBIRAICO MARTINEZ, RICARDO ANDRE

DIOSES CALLE, YLEM NADIA

IBARRA SANCHEZ, KEVIN RODRIGO

RIVAS PERALES, ERNESTO SEBASTIAN

Profesor:

DIAZ ARENAS, DANIEL JESUS

Lima - Perú

2023 - 01

Índice

I. Título

II. Descripción de la Organización o Caso

A. Descripción

B. Objetivo General

C. Objetivos Específicos

III. Diagrama de procesos (Actividades del proceso de negocio elegido)

A. Atención Presencial

B. Atención Telefónica

IV. Requerimientos funcionales y no funcionales

V. Diagrama de casos de uso del sistema

VI. Diseño Web del aplicativo

VII. Modelo de la base de datos

A. Modelo Entidad - Relación

B. Modelo Relacional

VIII. Mantenimientos y procesos

IX. Revisión de bibliografía. Mínimo 10 papers

1. Título

“Tres Marías Restaurant”

2. Descripción de la Organización o Caso

a. Descripción

Tres Marías es un restaurante que brinda un servicio de alta calidad a sus clientes. Los componentes esenciales que conforman este restaurante son las mesas, los meseros, los clientes y las reservas.

Las mesas están organizadas de manera que se cree un ambiente agradable y cómodo para los clientes. Los meseros son altamente capacitados y están comprometidos con proporcionar un servicio excepcional, desde tomar los pedidos hasta entregar la comida y bebida.

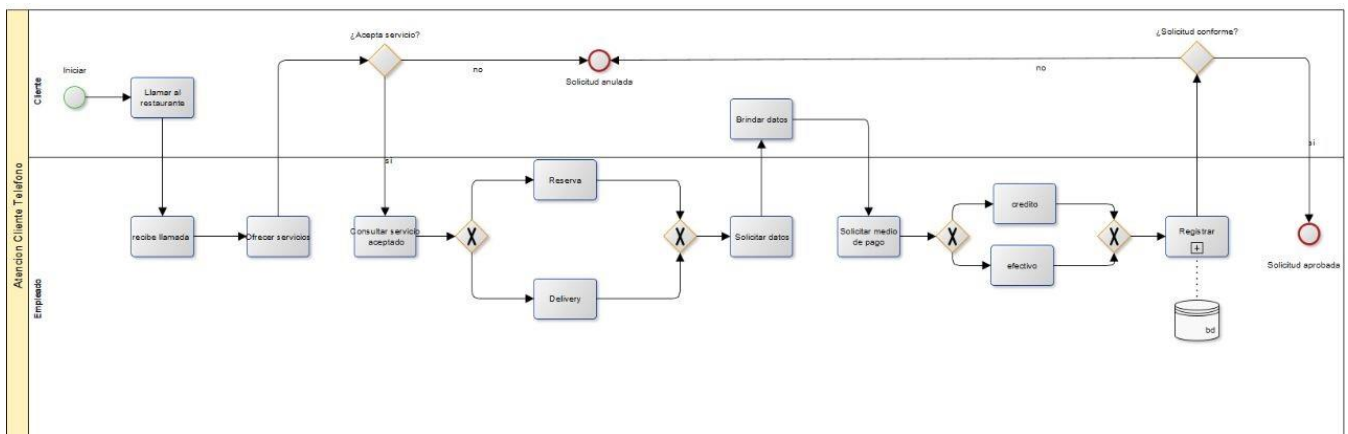
Los clientes son la prioridad en Tres Marías, y se garantiza que siempre se sientan bien atendidos y satisfechos con la comida y el servicio. Las reservas permiten a los clientes planificar su visita con anticipación y asegurar su lugar en el momento deseado.

Además de estos componentes, el sistema de gestión de Tres Marías también incluye los platillos, un seguimiento en tiempo real del estado de las mesas y los pedidos, horarios y rotación de meseros, edición de datos de usuarios e inicio de sesión.

En cuanto al servicio de entrega a domicilio, Tres Marías ofrece este servicio para mayor comodidad de sus clientes. El proceso de gestión de los pedidos es el siguiente: los clientes pueden llamar al restaurante para hacer su pedido, indicando si es para delivery o para llevar, en otras palabras, un "Pick Up". Un mesero/camarero atenderá la llamada y procederá a tomar el pedido, proporcionando un servicio amable y eficiente.

Una vez recibido el pedido, el restaurante lo preparará para el envío y se asegurará de que llegue al destino deseado en el menor tiempo posible. Los clientes pueden estar seguros de que sus pedidos serán entregados en perfectas condiciones y con el mismo nivel de calidad que se ofrece en el restaurante.

En definitiva, Tres Marías es un restaurante que se preocupa por brindar la mejor experiencia posible a sus clientes, tanto en el salón como en la entrega a domicilio. Su sistema de gestión integral y su



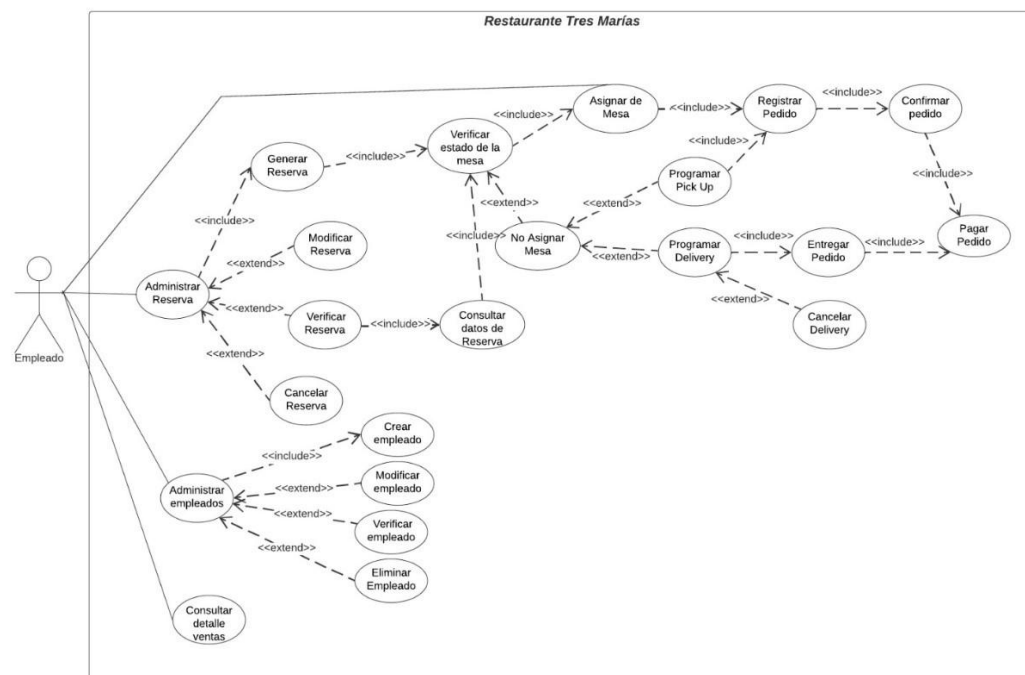
4. Requerimientos funcionales y no funcionales

Proceso de Negocio	STRQ	Requerimiento	RF / RNF	Requisito	UC	Caso de Uso	Actor
ATENCIÓN CLIENTE	1	El sistema debe contemplar la creación del usuario empleado	RF	El sistema debe permitir administrar un empleado	EMP-01	Administrar empleado	Empleado
			RF	El sistema debe permitir editar los datos de un empleado	EMP-02	Modificar empleado	Empleado
			RF	El sistema debe permitir validar un empleado	EMP-03	Verificar empleado	Empleado
			RF	El sistema debe permitir crear un empleado	EMP-04	Crear empleado	Empleado
			RF	El sistema debe permitir eliminar un empleado	EMP-05	Administrar empleado	Empleado
	2	Contemplar el proceso de reserva del cliente	RF	El sistema debe permitir generar una reserva	EMP-06	Generar reserva	Empleado
			RF	El sistema debe permitir modificar una reserva	EMP-07	Modificar reserva	Empleado
			RF	El sistema debe permitir verificar una reserva	EMP-08	Verificar reserva	Empleado
			RF	El sistema debe permitir cancelar una reserva	EMP-09	Cancelar reserva	Empleado
			RF	El sistema debe permitir asignar una mesa a un cliente que se reserva con anticipación o sea un cliente casual	EMP-10	Asignar mesa	Empleado
			RF	El sistema debe permitir consultar los datos de una reserva	EMP-11	Consultar datos reserva	Empleado
			RF	El sistema debe permitir verificar el estado de una mesa	EMP-12	Verificar estado mesa	Empleado

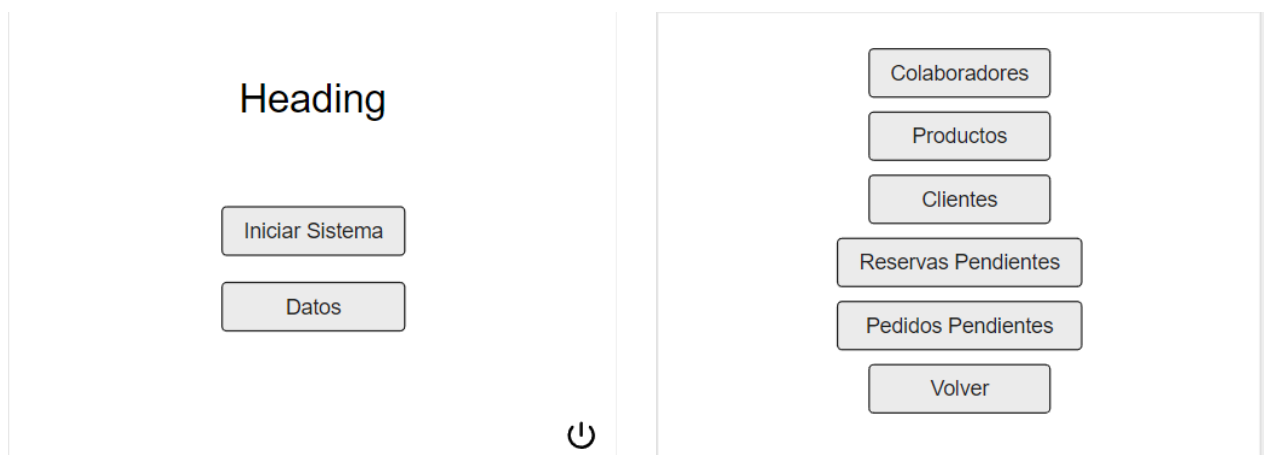
	3	Contemplar el pedido de la reserva	RF	El sistema debe permitir registrar el pedido	EMP-1 3	Programar pickup	Empleado
			RF	El sistema debe confirmar el pedido	EMP-1 4	Confirmar pedido	Empleado
			RF	El sistema debe poder cancelar un delivery	EMP-1 5	Cancelar delivery	Empleado
	4	Contemplar la forma de atención	RF	El sistema debe validar un pickup	EMP-1 6	Programar pickup	Empleado
			RF	El sistema debe validar un delivery	EMP-1 7	Programar delivery	Empleado
	5	Contemplar el uso de métodos de pago	RF	El sistema debe permitir el uso de métodos de pago	EMP-1 8	Pagar pedido	Empleado
	6	Consultar detalle de las ventas	RF	El sistema debe permitir consultar detalle de ventas	EMP-1 9	Consultar detalle de ventas	Empleado
	6	Contemplar visualización de notificaciones de escritorio	RNF	El sistema debe permitir visualizar notificaciones de escritorio	-	-	-
	7	Contemplar permiso para mantener inicio de sesión abierta	RNF	El sistema debe permitir mantener inicio de sesión abierta	-	-	-
	8	Contemplar el uso del lenguaje de programación JAVA	RNF	El sistema debe utilizar para su programación lenguaje PHP	-	-	-
	9	Contemplar el uso de base de datos	RNF	El sistema debe considerar utilizar una base de datos MySQL	-	-	-
	10	El sistema debe estar operativo las 24 horas y los 7 días de la semana	RNF	El sistema debe estar operativo las 24 horas y los 7 días de la semana	-	-	-
	11	Contemplar la generación de un log de errores	RNF	El sistema debe generar un log de errores	-	-	-

	12	Contemplar respaldo en la nube	RNF	El sistema debe almacenar información de respaldo en la nube	-	-	-
	13	Contemplar soporte para los sistemas operativos Mac y Windows	RNF	El sistema debe ser soportado por los sistemas operativos Mac y Windows	-	-	-
	14	Contemplar soporte en navegadores como: Mozilla y Chrome	RNF	El sistema debe ser manejado en navegadores como: Mozilla y Chrome	-	-	-

5. Diagrama de casos de uso del sistema



6. Diseño Web del aplicativo



Dato 1

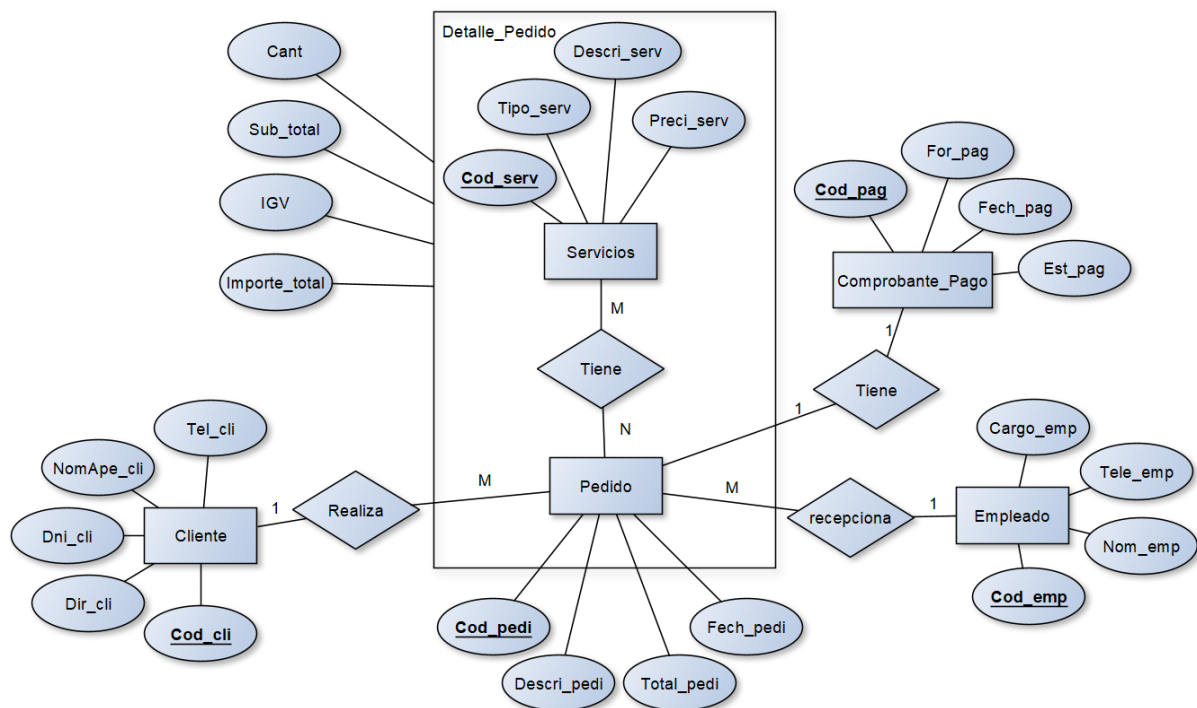
Dato 2

Actual Time: dd/mm/yy hh:mm

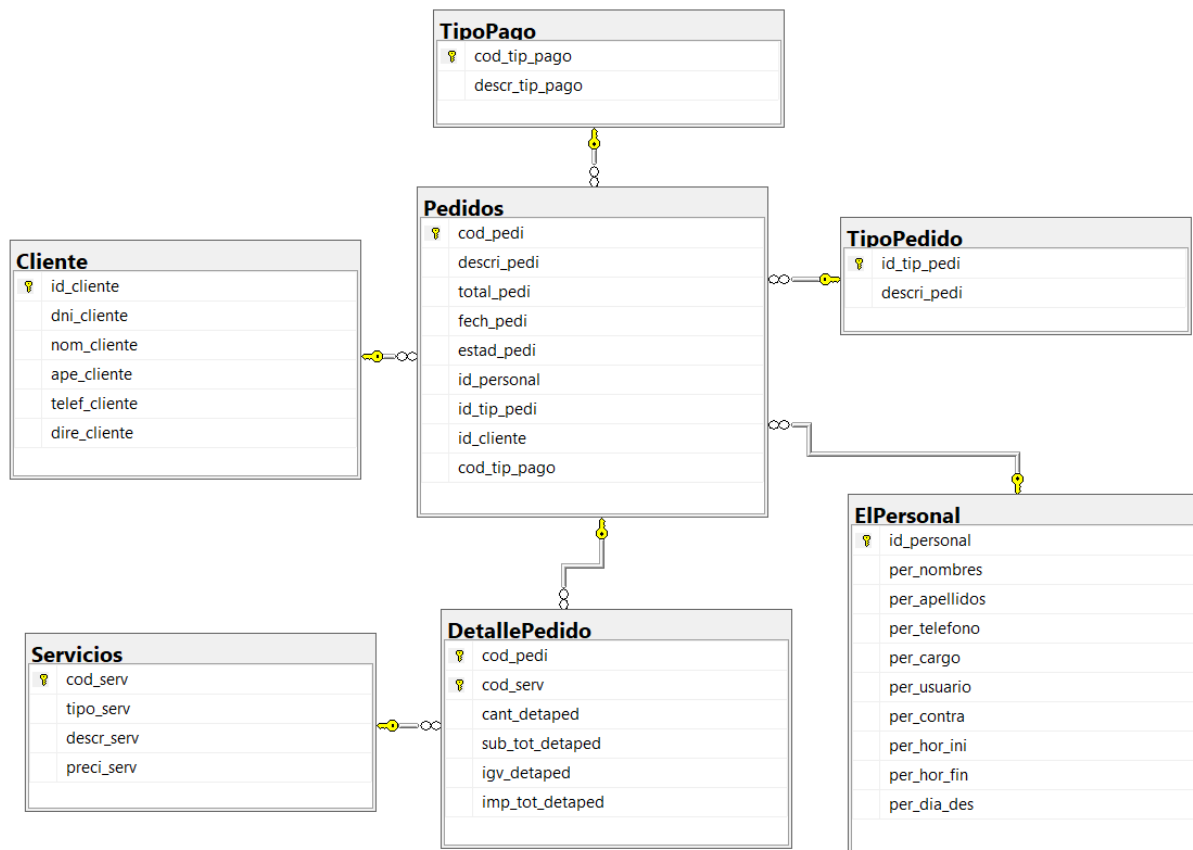
<input type="button" value="1"/>	<input type="button" value="2"/>	<input type="button" value="3"/>	<input type="button" value="4"/>
<input type="button" value="5"/>	<input type="button" value="6"/>	<input type="button" value="7"/>	<input type="button" value="8"/>
<input type="button" value="9"/>	<input type="button" value="10"/>	<input type="button" value="11"/>	<input type="button" value="12"/>

7. Modelo de la base de datos

a. Modelo Entidad - Relación



b. Modelo relacional



8. Mantenimientos y procesos

<https://github.com/QuantumRevenant/ProjectTheta>

9. Revisión de bibliografía. Mínimo 10 papers

Tema: Patrones de Diseño

Nombre de la investigación	Autores	Fecha	Resumen de la investigación	Obtenido de
Pattern Languages in HCI: A critical review	Andy Dearden; Janet Finlay	21/01/2006	En este artículo se realiza una revisión crítica de los patrones y lenguajes de patrones en la interacción humano-computadora (HCI, por sus siglas en inglés). Estos elementos han recibido una atención significativa en HCI en los últimos años debido a su potencial como herramientas para desarrollar y comunicar información y conocimiento que respalde un buen diseño. La revisión tiene como	SCOPUS

			objetivo examinar los antecedentes de los patrones y lenguajes de patrones en HCI, y situar los lenguajes de patrones en relación con otros enfoques para el diseño de interacción. Y solo nos centraremos en el patron de diseño.	
Challenges and Common Solutions in Smart Contract Development	Kannengieser, Niclas; Lins, Sebastian; Sander, Christian; Winter, Calus; Frey, Hellmuth; Sunyaev, Ali	11/11/22	Los contratos inteligentes utilizando tecnología de contabilidad distribuida (DLT) tienen el potencial de formalizar acuerdos, pero se enfrentan a desafíos de programación y seguridad. Este artículo hace mención sobre diferentes entrevistas y revisiones que se realizó con el fin de identificar diferentes problemas lo cual a su vez dan propuestas a distintas soluciones, incluyendo Patrones de Diseño de Software(SDP). Estos patrones ayudan a los desarrolladores a mejorar sus prácticas y a integrar contratos inteligentes en diferentes protocolos DLT.	SCOPUS
Optimising Energy Consumption of Design Patterns	Adel Nouredine, Ajitha Rajan	24/05/15	En esta investigación, los patrones de diseño mejoran la productividad y mantenibilidad del software. Sin embargo, tienen un alto consumo de energía. La visión es detectar y transformar automáticamente los patrones durante la compilación para mejorar la eficiencia energética sin cambiar las prácticas de codificación. Proponemos transformaciones del compilador para los patrones Observer y Decorator.	
Landscape of Architecture and Design Patterns for IoT Systems	Hironori Washizaki,	10/10/20	Esta investigación describe la proliferación de Internet de las Cosas (IoT) , los diseñadores necesitan patrones de diseño de sistemas y software de IoT para ayudar en el diseño de soluciones escalables y replicables. Se realiza una revisión de 143 patrones de	

			arquitectura y diseño de IoT en 32 artículos. Se analizan varias de sus características. En el 57% de los patrones investigados no son específicos de IoT, lo que indica que en ambos casos se suelen diseñar con patrones convencionales.	
Design Patterns for Developing Dynamically Adaptive Systems	Andres J. Ramirez; Betty H.C. Cheng	8/5/2010	El artículo se centra en el desarrollo de sistemas dinámicamente adaptables y presenta patrones de diseño que pueden ayudar a los desarrolladores a crear software más flexible y adaptable. Resalta los desafíos que enfrentan los desarrolladores al tratar de construir sistemas que satisfagan todos los requerimientos y restricciones que puedan surgir. Para ello se presenta un proceso de desarrollo basado en modelos de sistemas dinámicamente adaptables llamado "Zhang-Cheng", el cual utiliza patrones de diseño recopilados de mas de treinta implementaciones de proyectos relacionados con la adaptación.	
Notable design patterns for domain-specific languages	Spinellis, Diomidis	1/2/2001	El artículo se enfoca en los patrones de diseño para DSL(Domain-Specific Lenguaje) y propone ocho patrones recurrentes que se han identificado como útiles para el diseño e implementación de DSL. Para cada patrón descrito proporcionan un nombre que usarán para describirlo, ilustran su estructura usando un simple diagrama UML, lo clasifican como creacional, conductual o estructural, resaltan las situaciones en las que se puede aplicar, describen como el patrón apoya sus objetivos y proporcionan ejemplos y pautas prescriptivas para la implementación del patrón.	

Design Pattern as a Service for Blockchain-Based Self-Sovereign Identity	Liu, Yue; Lu, Qinghua; Paik, Hye-Young; Xu, Xiwei; Chen, Shiping; Zhu, Liming	9/2020-10/2020	El artículo aborda la falta de diseños arquitectónicos sistemáticos en los sistemas de identidad auto-soberana (SSI) basados en blockchain. Se destaca que las soluciones actuales son de granularidad gruesa y presentan riesgos de seguridad. El artículo propone identificar los ciclos de vida de tres objetos principales de SSI y presenta patrones de diseño de granularidad fina para el desarrollo de aplicaciones. También se propone una arquitectura de plataforma de SSI que utiliza patrones de diseño como servicios para mejorar la escalabilidad y seguridad. Se implementa un prototipo y se evalúa su viabilidad y escalabilidad.	SCOPUS
Nonlinear finite-element analysis software architecture using object composition	McKenna, Frank; Scott, Michael H.; Fenves, Gregory L.	2010	El artículo se centra en la composición de objetos en el análisis de elementos finitos en ingeniería de software. Se destaca que esta técnica ofrece ventajas sobre la herencia de clases para lograr una arquitectura flexible. En este enfoque, se utilizan clases separadas que encapsulan algoritmos fundamentales de elementos finitos y se comunican entre sí mediante patrones de diseño. Se implementan algoritmos de búsqueda de raíces, métodos de integración temporal, manejadores de restricciones, solucionadores de ecuaciones lineales y enumeradores de grados de libertad como componentes intercambiables utilizando el patrón de estrategia. Los patrones de puente y método de fábrica permiten la variación independiente de los objetos del modelo de elementos finitos y los objetos que implementan los procedimientos de solución numérica. Los patrones de	SCOPUS

			<p>adaptador e iterador permiten el ensamblaje de ecuaciones a través de interfaces abstractas, sin exponer los detalles de almacenamiento o cálculo. Los diagramas de secuencia documentan la interoperabilidad de las clases de análisis para resolver ecuaciones no lineales de elementos finitos, demostrando que la composición de objetos con patrones de diseño proporciona un enfoque general para desarrollar y mejorar el software de elementos finitos no lineales.</p>	
<p>GOF (the gang of four) design patterns in the context of process development of web-oriented applications</p>	<p>Carlos A. Guerrero, Johanna M. Suárez y Luz E. Gutiérrez</p>	<p>2013</p>	<p>El objetivo del artículo es proporcionar un precedente sobre el uso de patrones de diseño GOF en la industria del software en Colombia. Se utiliza como base el catálogo de patrones descrito en el libro "Design Patterns: Elements of Reusable Object-Oriented Software" de Gamma et al. Se busca identificar los patrones de diseño utilizados en la actualidad a través de la construcción de criterios de selección y la colaboración de expertos en el campo.</p> <p>El artículo destaca que aplicar patrones de diseño se considera una buena práctica que mejora la calidad del proceso de desarrollo y del producto final, además de estandarizar el código fuente para facilitar el mantenimiento. Los patrones de diseño son soluciones a problemas recurrentes en el diseño de software, y se clasifican en patrones creacionales, patrones estructurales y patrones de comportamiento.</p> <p>El estudio encontró dificultades debido al alto nivel de abstracción del catálogo GOF, y los expertos señalaron que el uso efectivo de los</p>	<p>SCOPUS</p>

			<p>patrones de diseño requiere años de experiencia. Además, se identificó que la estructura del catálogo no proporciona ejemplos claros de implementación, lo que dificulta su comprensión.</p> <p>En las conclusiones, se destaca que el catálogo GOF no establece una categorización exclusiva de los patrones de diseño para aplicaciones web. Se advierte sobre el error de considerar los patrones de diseño como la solución universal para cualquier problema de desarrollo de software, ya que no todos los casos son aptos para su uso.</p>	
<p>Desarrollo e implementación de un sistema basado en patrones de diseño para optimizar los servicios de pesaje en la Empresa Balanzas Vegasystems SAC</p>	<p>Lopez Vasquez, Giankarlos</p>	<p>09/04/2020</p>	<p>El artículo aborda la importancia de optimizar estos servicios para garantizar la precisión y eficiencia en los procesos de pesaje. El autor propone utilizar patrones de diseño como una estrategia para organizar y estructurar el sistema, aprovechando las mejores prácticas y soluciones probadas en el campo del diseño de software. El artículo describe el proceso de desarrollo e implementación del sistema, incluyendo el análisis de requisitos, el diseño de la arquitectura, la selección de los patrones de diseño adecuados y la construcción del sistema en base a estos patrones. Los resultados obtenidos demuestran mejoras significativas en la precisión y eficiencia de los servicios de pesaje, lo cual contribuye al crecimiento y éxito de la empresa Balanzas Vegasystems SAC.</p>	<p>GOOGLE ACADÉMICO</p>

