



Facultad de Ciencias y Tecnologías Computacionales

SISTEMA INFORMÁTICO PARA LA GESTIÓN DE LOS PROCESOS DE CONTRATACIÓN Y FACTURACIÓN EN LA UEB SOLUCIONES CONSTRUCTIVAS Y ACABADOS

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

Autor(es): Darío Alejandro Cánovas Losada

Tutor(es): MSc. Ruth Yurina Vega Cutiño

La Habana, 23 de diciembre de 2023

“Año 65 de la Revolución”

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “Sistema Informático para la gestión de los procesos de Contratación y Facturación en la UEB Soluciones Constructivas y Acabados” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firman la presente a los ____ días del mes de ____ del año ____.

Darío Alejandro Cánovas Losada

Firma del Autor

MSc. Ruth Yurina Vega Cutiño

Firma del Tutor

AGRADECIMIENTOS

A mis profesores que durante toda mi vida estudiantil me han guiado como una segunda familia. A mis padres Marcia Losada García y Alejandro Cánovas Pérez por apoyarme en todo y cuidarme. Sin su educación no sería quien soy. A mi Padrino Horacio Chamizo García y al “tío Juan” García Prieto por su apoyo incondicional. A mi amiga tía Laritza Martínez Negrín. A mi compañera y amiga Yaimellis González Martínez.

A todos aquellos que me han brindado su apoyo,

MUCHAS GRACIAS

RESUMEN

La Unidad Empresarial de Base (UEB) Soluciones Constructivas y Acabados que pertenece al Grupo Empresarial de la Industria Alimentaria (GEIA) tiene como objeto social la terminación de las obras constructivas pertenecientes al Ministerio de la Industrias Alimentaria (MINAL). Entre los tipos de obras se encuentran; la carpintería de aluminio, pintura, falso techo, revestimientos en pladur y electricidad. La UEB en cuestión, opera en todo el país, dentro de los marcos de su ministerio, enviando brigadas de obreros especializados a donde se necesiten. Un solo comercial realiza toda la contratación y facturación de los trabajos realizados por lo que a veces necesita trasladarse físicamente al sitio de la obra llevando consigo carpetas con documentación concerniente a los trabajos realizados. Tras la identificación de varias deficiencias detectadas en la gestión de la contratación de los servicios y su facturación, surge la necesidad de informatizar el proceso de contratación y facturación en los servicios de la UEB Soluciones Constructivas y Acabados perteneciente al MINAL, para disminuir los tiempos de atención a clientes, localización de servicios asociados y estado de la documentación. Para llevar a cabo el ciclo de desarrollo del software se empleó la metodología *AUP-UCI*, como lenguaje de programación *Visual Basic para Aplicaciones (VBA)* usando el *framework* de desarrollo *.Net* y como gestor de base de datos Microsoft Office Access y como herramienta para el modelado Visual Parading

Como resultado se obtuvo una aplicación que gestiona contratos y facturas por clientes, optimizando la gestión de la contratación de los servicios y su facturación.

PALABRAS CLAVE

AUP-UCI, base de datos, contratos, facturas suplementos, UEB, VBA

Abstract

The Base Business Unit (UEB) Constructive Solutions and Finishes that belongs to the Food Industry Business Group (GEIA) has as its corporate purpose the completion of construction works belonging to the Ministry of Food Industries (MINAL). Among the types of works are: aluminum carpentry, painting, false ceiling, plaster-board coverings and electricity. The UEB in question operates throughout the country, within the framework of its ministry, sending brigades of specialized workers wherever they are needed. A single salesperson carries out all the contracting and invoicing of the work carried out, so sometimes he or she needs to physically travel to the construction site, carrying folders with documentation concerning the work carried out. After the identification of several deficiencies detected in the management of the contracting of services and their billing, the need arises to computerize the contracting and billing process in the services of the UEB Constructive Solutions and Finishes belonging to the MINAL, to reduce processing times. customer service, location of associated services and status of documentation. To carry out the software development cycle, the AUP-UCI methodology was used, as a Visual Basic for Applications (VBA) programming language using the .Net development framework and as a Microsoft Office Access database manager and as a tool for Visual Paradigm modeling

As a result, an application was obtained that manages contracts and invoices for clients, optimizing the management of contracting services and their billing.

KEYWORDS

AUP-UCI, database, contracts, supplements, invoices, VBA, UEB

TABLA DE CONTENIDO

INTRODUCCIÓN.....	9
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	14
1.2 Principales conceptos relacionados	14
1.3 Antecedentes	16
1.4 Metodología de desarrollo de software	19
1.5 Lenguaje, Herramientas y tecnologías	20
Conclusiones del capítulo	23
CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO	24
2.1 Análisis del negocio	24
2.2 Requisitos Funcionales	26
2.2.1 Descripción de requisitos funcionales.....	27
2.3 Requisitos no Funcionales	32
2.4 Casos de uso del Sistema.....	33
2.5 Arquitectura y Diseño de Software	36
Conclusiones del capítulo	43
CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA	44
3.1 Estrategia de prueba.....	44
3.2 Pruebas internas	44
3.3 Pruebas funcionales.....	49
3.4 Aplicación de la Técnica de ladov.....	55
3.5 Beneficios de la aplicación desarrollada	58
Conclusiones del capítulo	59

Tabla de contenido

CONCLUSIONES FINALES	60
RECOMENDACIONES	61
REFERENCIAS BIBLIOGRAFICAS	62

ÍNDICE DE TABLAS

Tabla 1. Especificación de requisitos funcionales	26
Tabla 2. descripción de requisitos funcionales	27
Tabla 3. Descripción del Caso de uso Gestionar Contrato.....	34
Tabla 4. Camino básico 1: 1-2-4	47
Tabla 5. Camino básico 2: 1, 2, 3, 4.....	47
Tabla 6. Camino básico 3: 1, 4.....	48
Tabla 7. Casos de prueba	50
Tabla 8. Cuadro lógico de ladv de la investigación	56

ÍNDICE DE FIGURAS

Figura 1. Proceso de contratación.....	25
Figura 2. Casos de Uso del Sistema	34
Figura 3. Modelo Vista Controlador	37
Figura 4. Modelo de Datos	38
Figura 5. Diagrama de Clase del caso de Uso Gestionar Contrato	39
Figura 6. representación del Patrón Experto	40
Figura 7. Representación del Patrón Creador	40
Figura 8. Representación del Patrón Alta Cohesión.....	41
Figura 9. Representación del Patrón Bajo Acoplamiento	41
Figura 10. Representación del Patrón Factory Method	42
Figura 11. Diagrama de Despliegue	42
Figura 12. Prueba del camino básico	45
Figura 13. Nodos.....	46
Figura 14. Valores Prueba de ladov.....	55
Figura 15. Formula de ladov	56
Figura 16. Rangos prueba de ladov	56
Figura 17. Resultado obtenido al aplicar la encuesta.....	57
Figura 18. Resultados formula de ladov.....	57

AVAL DEL CLIENTE

La Habana, 12 de marzo de 2018.

"Año 59 de LA Revolución".

A: Quien corresponda.

De: Msc. Ingeniero Víctor Recio Salles
(Director UEB Soluciones Constructivas y Acabado)

De: Lic. Danaize Martínez Montero
(Jefa Dpto. Operaciones de la
UEB Soluciones Constructivas y Acabado)

Asunto: Constancia de autoría.

Por este medio en uso de las facultades como Director y Jefa del Dpto. respectivamente hacemos constar que el técnico en soluciones ingenieras Darío Alejandro Cánovas Losada (CI: 87111708229) trabajador de UEB Soluciones Constructivas y Acabado, es el autor de la aplicación: Operaciones Módulo de Facturación y Contratos (OMFC).

Esta aplicación consta de dos módulos desarrollados en Microsoft Access, uno, para llevar la relación de facturas y las, otro, para describir los contratos/suplementos por clientes.

Se hace constar que la aplicación antes descrita se está utilizando desde 12 de marzo de 2018 y con esto se logra hacer más viable el flujo de información, automatizar y uniformar la recogida de datos y beneficia directamente al Dpto. de Operaciones de UEB Soluciones Constructivas y Acabado, perteneciente al Ministerio de la Industria Alimentaria.

Para que así conste firmamos la presente atestación.


Master en Ingeniería Víctor Recio Salles
(Director UEB Soluciones
Constructivas y Acabado)


Lic. Danaize Martínez Montero
(Jefa Dpto. Operaciones de la
UEB Soluciones Constructivas y Acabado)

nota bene: con copia en la Dirección y Dpto. Operaciones de la UEB Soluciones Constructivas y Acabado.

INTRODUCCIÓN

Cuba se encuentra inmersa en el proceso de informatizar la sociedad, por lo que se empeña en implementar de forma consciente y masiva las Tecnologías de la Información y las Comunicaciones (TIC's); en este sentido, especial atención se dedica en garantizar la concreción de estas necesidades informáticas en sus múltiples esferas. Un ejemplo de ello, es la creación del Ministerio de Comunicaciones (MINCOM)¹. Este, junto con el Grupo Empresarial de la Informática y las Comunicaciones (GEIC), tienen como objetivo impulsar, facilitar y ordenar el uso masivo de servicios y productos de las TIC, para mejorar la calidad de vida de la población en aras de perfeccionar los procesos productivos como real fuente de riquezas del país.

El Ministerio de la Industria Alimentaria, Organismo de la Administración Central del Estado (OACE), encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno, en el desarrollo de la Industria Alimentaria, no queda atrás en esta estrategia para lograr la informatización de todos los procesos a los cuales les sea factible aplicar la informatización.

Con el surgimiento de nuevas formas no estatales de producción se ha abierto un nuevo sector en la producción y los servicios que lucha por emerger como fuerza productiva en el país, lo que también, ha contribuido a estimular en estos últimos años, en Cuba, el uso de las Tecnologías de la Información. Lo anterior influye también en la necesidad de una modernización en su enfoque organizativo, en cuanto

¹ “El Ministerio de Comunicaciones de la República de Cuba, conocido por el acrónimo MINCOM, es el ministerio encargado de las telecomunicaciones, informática, industria electrónica, servicios postales, automatización, infraestructura para la transmisión de radio y televisión y la administración del espectro radioeléctrico de Cuba. Entre 2000 y 2013 se denominó Ministerio de Informática y las Comunicaciones”.

En: <https://www.mincom.gob.cu/es/perfil-institucional/nuestra-historia>

a la forma de almacenar datos de clientes, servicios a estos y producción contratada. En un futuro cercano, cuando estén bien insertados los mecanismos de control y planificación del flujo productivo, se facilitará el estudio del cliente y del mercado para incentivar el pensamiento prospectivo de las diferentes entidades.

La Unidad Empresarial de Base (UEB) Soluciones Constructivas y Acabados, perteneciente al MINAL² es una empresa cuya función básica está orientada a ofrecer a los clientes, servicios de carpintería de aluminio, acabados en pladur, falso techos, pintura, albañilería, electricidad, entre otros.

El desarrollo tecnológico a nivel mundial ha modificado drásticamente la manera de acceder a la información, y esto ha estado en constante evolución, para lograr tener acceso rápido a sus datos. En la “era de la red de redes” (Internet), se espera por parte de la entidad o persona que participe como especialista en el circuito de comunicación, un trato rápido y eficiente a la hora de brindarle servicio al cliente.

La UEB mueve decenas de contratos y suplementos por mes, facturando servicios en varios centros de costo, dentro y fuera de su ministerio. Actualmente se llevan por separado y en varios archivos, los papeles con todas las informaciones de contratos, suplementos y facturas. Esta forma de almacenar la información hace engorroso el acceso a cualquier dato necesario sobre un cliente, lo cual hace demorar incluso horas cualquier proceso. Otra de las dificultades surge cuando se necesita rastrear el estado legal de un contrato, por ejemplo, si ya fue firmado por ambas partes o en poder de quien está. Así mismo, la confección de los suplementos y facturas a fines operativos pueden detener el proceso productivo por semanas, debido a pérdida o extravío de la documentación.

² “El Ministerio de la Industria Alimentaria de la República de Cuba, conocido por el acrónimo MINAL, es el ministerio encargado de la producción de alimentos en Cuba”. En: <https://www.minal.gob.cu/ministerio>

A partir de lo planteado anteriormente se define el siguiente **problema de la investigación**: ¿Cómo gestionar el proceso de contratación y facturación en los servicios de la UEB Soluciones Constructivas y Acabados perteneciente al MINAL para disminuir los tiempos de atención a clientes, localización de servicios asociados y estado de la documentación?

Como **objeto de estudio** se define sistemas de gestión de contratos y como **campo de acción** las aplicaciones informáticas que gestionen contratos para los servicios generales de construcción y montaje de obras.

Se precisa, por tanto, como **objetivo general**: desarrollar una aplicación que permita la gestión y organización, representación y recuperación de información, para el proceso de contratación y facturación en la UEB Soluciones Constructivas y Acabados, con lo que se delimita su **campo de acción** el desarrollar sistema que permita la gestión y organización, representación y recuperación de información, para el proceso de contratación y facturación en la UEB Soluciones Constructivas y Acabados

Para guiar el desarrollo de la presente investigación surgen las siguientes **preguntas de científicas**:

1. ¿Cuáles son los antecedentes teóricos-metodológicos de la gestión de contratos y facturas?
2. ¿Cuál es el estado actual de la gestión de contratos y facturas en la UEB Soluciones Constructivas y Acabados?
3. ¿Qué características debe tener un sistema de gestión de contratos y facturas en para UEB Soluciones Constructivas y Acabados?
4. ¿Qué resultados tendrá insertar en el flujo de producción la aplicación de gestión de contratos y facturas?

Para dar cumplimiento a lo antes planteado se definen las siguientes **tareas de investigación**:

1. Análisis de los fundamentos teóricos-metodológicos que sustentan la gestión de contratos y facturas.

2. Diagnóstico de las características y limitaciones de las soluciones existentes referentes a los sistemas de contratación y facturación, para decidir si construir o aprovechar algunas de las ya implementadas.
3. Caracterización de las herramientas y tecnologías que son adecuadas para la elaboración del sistema de gestión.
4. Elaboración de los artefactos (modelación) y la documentación según la metodología de desarrollo seleccionada, para documentar la aplicación que se propone.
5. Implementación del sistema para la gestión de los procesos de contratación y facturación en los servicios al cliente de la UEB Soluciones Constructivas y Acabados.
6. Validación del sistema propuesto a partir de la realización de pruebas con la UEB.

Para desarrollar la investigación se utilizarán los siguientes métodos de investigación científica:

Métodos teóricos

Análisis histórico-lógico: Este método se empleó para el conocimiento general de las principales características de los diferentes sistemas que gestionan contratos de servicio y facturación, tanto a nivel internacional como nacional, lo que facilitó, tener una visión de su evolución y comportamiento.

Analítico sintético: El empleo de este método permitió (Sampieri & Lucio., 2003) (González., 2011)organizar su base teórica,` conocer y comprender las etapas del proceso de gestión de facturas y contratos y fragmentarlo en pequeños conceptos para poder alcanzar los objetivos trazados en la investigación.

Inductivo-Deductivo: Se utilizó para el planteamiento del objetivo y la extracción de las ideas fundamentales para la elaboración y fundamentación del trabajo de diploma. Se emplea para la identificación de la problemática y de las soluciones.

Modelación: Este método se utilizó para comprender y describir los principales procesos de sistemas que gestionan contratos de servicio y facturación en las UEB del MINAL. Lo anterior resulta condición necesaria para el diseño en proposición de

este proyecto de tesis, mediante el esbozo de los diferentes diagramas definidos en la metodología seleccionada, lo cual permitió una mejor toma de decisión para la proposición final de las funcionalidades de la aplicación.

Métodos empíricos

Entrevista: La entrevista es una técnica de recopilación de información mediante una conversación profesional. Esta técnica permite adquirir información acerca de lo que se investiga y depende en gran medida del nivel de comunicación entre el investigador y los participantes en la misma. Se utilizó con el objetivo de conocer la estructura organizativa del proceso de gestión comercial de la empresa UEB Soluciones Constructivas y Acabados, La Habana, que pertenece al MINAL. Se le realiza la entrevista a la Directora de Operaciones de la empresa y al Director.

Observación: Se utilizó para obtener una información más precisa de cómo se realiza el proceso de contratación y facturación de la UEB Soluciones Constructivas y Acabados.

Este trabajo de diploma se encuentra estructurado de la siguiente forma:

El **Capítulo 1** describe las características inherentes al proceso de contratación y facturación de la UEB Soluciones Constructivas y Acabados, analizando conceptos, herramientas y tecnologías útiles en la solución de la problemática planteada; se describen también las características de introducción y limitaciones de las soluciones existentes, para proponer que se debe desarrollar una aplicación específica a esta necesidad, y se caracterizan las herramientas y tecnologías adecuadas para la elaboración del sistema informático.

En el **Capítulo 2** documenta las fases de análisis y diseño de la solución, específicamente para el proceso de contratación y facturación de la UEB Soluciones Constructivas y Acabados.

El **Capítulo 3** describe las pruebas realizadas al sistema y los resultados alcanzados; se muestra, además, un conjunto de artefactos como el diagrama de componentes.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan los principales conceptos relacionados con la gestión de los contratos y facturación en la UEB Soluciones Constructivas y Acabados perteneciente al GEIA. Se realiza un estudio del estado del arte de sistemas informáticos que apoyan el proceso de contratación en las empresas. Entre ellos, se analizan sistemas ERP y otros sistemas a la medida que permiten crear y modificar documentos técnicos legales relacionados con los contratos. Además, se definen las herramientas, metodología y lenguajes, que se emplean en la propuesta del sistema.

1.2 Principales conceptos relacionados

Gestión es asumir y llevar a cabo las responsabilidades sobre un proceso, esto puede ser empresarial o personal, lo que incluye: La preocupación por la disposición de los recursos y estructuras necesarias para que tenga lugar en el comercio. La coordinación de sus actividades (y correspondientes interacciones) y sus semejantes.

El término gestión es utilizado para referirse al conjunto de acciones, o diligencias que permiten la realización de cualquier actividad o deseo. Dicho de otra manera, una gestión se refiere a todos aquellos trámites que se realizan con la finalidad de resolver una situación o materializar un proyecto. En el entorno empresarial o comercial, la gestión es asociada con la administración de un negocio (Management Levels and Types | Boundless Management, 2023).

Contratos Según la Real Academia Española de idioma, un contrato es un pacto o convenio, oral o escrito, entre partes que se obligan sobre materia o cosa determinada, y a cuyo cumplimiento puede ser con obligación o por autoridad. Es un documento que recoge las condiciones de este pacto o convenio. Las definiciones de contrato usualmente pueden ser establecidas en dos sentidos, tanto en términos de acuerdos o como promesas. El contrato es un acuerdo, el cual es legalmente forzoso o legalmente reconocido como creador de un deber. Otra definición usada es la que considera al contrato como una promesa o conjunto de promesas, que en caso de incumplimiento el derecho otorga un remedio o el

reconocimiento por parte del derecho, en cierto sentido, como un deber. (REAL ACADEMIA ESPAÑOLA:, 2023)

Suplemento La modificación o extinción de un contrato por acuerdo de las partes, puede hacerse constar en suplemento, entendido este como el documento que se une al contrato después de otorgado. El suplemento también puede utilizarse para concretar el contenido del contrato o prorrogar su vigencia. (MINJUS, 2013)

El **centro de costos** es una unidad del negocio que genera costos para la empresa, y también beneficios, pero de forma indirecta.

Es decir, un centro de costos es aquella área donde se producen gastos necesarios para la actividad de la compañía. Por ejemplo, la oficina de compras, a través de este centro de costo la empresa adquiere los insumos para su proceso industrial.

Los centros de costos son importantes para la gestión de la compañía porque permiten conocer el origen de los gastos efectuados por la organización. De ese modo, es posible buscar mayor eficiencia, por ejemplo, eligiendo un proveedor que ofrezca menores precios.

Asimismo, los centros de costos son clave en la reestructuración de una empresa que debe, o se propone, bajar sus gastos. Así, puede eliminar, por ejemplo, desembolsos no imprescindibles o reducir personal que no está aportando la utilidad esperada al negocio. (Westreicher & López, 2023)

Factura Según la Real Academia Española de idioma, una factura es un documento emitido por un empresario o profesional en el que se relacionan los objetos o servicios incluidos en una transacción económica, junto con su precio y otros datos de identificación de los contratantes (Real Academia de la Lengua Española, 2023).

Sistemas de gestión empresarial. La integración de diferentes áreas de las empresas como ingeniería, ventas, fabricación o compras bajo un mismo sistema de información, permitió mejorar el enfoque de la gestión empresarial incorporando otras áreas en un único sistema integral. Debido a esto, comienzan a aparecer sistemas que abordan la planificación de recursos humanos o financieros junto con la planificación de necesidades de materiales y recursos de producción. En el

contexto actual de la gestión empresarial coexisten dos tendencias complementarias: la primera es la implementación de sistemas de gestión a partir de los estándares nacionales e internacionales, que facilitan de forma separada el desempeño de cada una de las funciones técnicas asociadas a las diferentes partes interesadas, y la segunda es el diseño de esquemas de excelencia empresarial que garantiza la satisfacción de los grupos de interés implicados en el proceso. Ejemplo de ellos son los sistemas ERP (Enterprise Resource Planning), softwares por módulos que interactúan entre sí. Estos son un factor crucial para el acceso instantáneo a la información, ya que facilita el flujo de información dentro de la organización y con los proveedores, clientes y otros intervinientes de la cadena de abastecimiento. En esos módulos se incluyen funcionalidades de marketing y ventas, distribución, gestión de producción, control de inventarios, gestión de la calidad, recursos humanos, gestión financiera, contabilidad y gestión de información, entre otros. (Siigo, 2023)

1.3 Antecedentes

El análisis de los sistemas de gestión empresarial se realiza a partir de las necesidades que posee la empresa en el proceso de Contratación y Facturación. Este tiene dentro de sus actividades, el control de los pedidos realizados por los clientes, así como el análisis de las ofertas que envían a los mismos. Para la realización de estas actividades, el Comité de Contratación establece una reunión una vez por semana, donde se analizan los pedidos de los clientes y los suministros de que dispone la empresa para brindarle el servicio solicitado por el cliente. De ser aceptado se crea un expediente del cliente donde se archivan: fichas de clientes, contratos y suplementos, las ofertas y las correspondientes facturas.

Para ello se estudiaron las siguientes propuestas:

El software **Odoo**, un marco de trabajo modular y escalable bajo la licencia de código libre. Dispone de funcionalidades oficiales y módulos desarrollados por terceros. Este funciona a través de una única plataforma en la que se encuentran integrados todos los módulos. Odoo permite llevar toda la gestión de operaciones, control financiero, productivo y de recursos humanos de las empresas, ya que es

sumamente flexible para adaptarse a los procesos comerciales de éstas. Además, ofrece la creación y utilización de módulos externos completamente personalizables y la creación digital de documentos técnicos legales.

Sin embargo, éste no posee un módulo dedicado al proceso de contratación que permita abarcar la totalidad de los requisitos definidos por el cliente, tales como: un contrato de servicios que se adapte a los diferentes servicios propios de la empresa, creación de suplementos modificativas a los contratos e imprimir modelos de facturas foliadas.

El **OrbitTrade**: Es un software que se adapta a distintos escenarios para proporcionar un control y un seguimiento exhaustivo sobre el proceso de exportación/importación. Permite gestionar la generación de pedidos, realizar proyecciones, además de ofrecer información detallada sobre el estado de los pedidos y de los estados con relación a las entregas, remisiones de planta a puertos, mercancía en puertos, embarques, liquidación de exportaciones, costos y estadísticas de ventas. Dentro de sus principales funcionalidades se encuentran (Nubia Casquete Baidal, 2023):

- Diseñar contratos con clientes y/o proveedores y anexos al mismo.
- Catálogos de productos por cliente/proveedor.
- Gestión documental en cualquier fase del proceso, como por ejemplo contratos, cartas de crédito y transporte.
- Definición de flujos de procesos con sus validaciones correspondientes.
- Identificación y control de la responsabilidad de cada inspección.
- Sistema de decisión de compra que permite analizar la propuesta más adecuada de los proveedores para cada cliente.

El **Softland**: Es un software que brinda la posibilidad de realizar el proceso administrativo de contratos de una forma fácil y eficiente. Está especialmente diseñado para empresas que requieren automatizar sus procesos contractuales, tales como contratos de servicios, contratos de arrendamiento de condominios y oficinas, maquinarias, entre otros. El módulo permite administrar diferentes características de los contratos:

- Tipo de Contrato
- Estados del Contrato
- Histórico de documentos relacionados con el contrato
- Copia de contratos y recalcule de impuestos
- Razones de suspensión
- Periodicidades de facturación
- Procesos de aumento de precios
- Generación de facturas en lotes e individuales

Luego de la realización del estudio de los sistemas mencionados anteriormente, se puede concluir que:

Softland permite ver como es el funcionamiento de sistemas relacionados con el proceso de contratación a través del enfoque de trabajo orientado a la nube. El estudio de éstos, se realizó con el fin de desarrollar la propuesta de solución lo más adaptable posible, disminuyendo la dependencia entre los datos y creando la menor cantidad de relaciones posibles.

OrbitTrade al ser un software dedicado a la gestión de importaciones y exportaciones, permite ver cómo funciona el control de los pedidos a través de interfaces amigables. Como generar diferentes estadísticas por ofertas, países, proveedores, así como el análisis de resultados de la gestión de toda la operación relacionada con la actividad de importación y exportación.

Se decide utilizar características de estos programas analizados como parte de la solución ya que ninguno se adapta en su totalidad a las necesidades. Por tanto se propone desarrollar una solución desktop lo más ligera posible que no implique la instalación de programas adicionales o la intervención de personal especializado para su implementación.

1.4 Metodología de desarrollo de software

Las metodologías de desarrollo de software son indispensables en la creación o actualización de un proyecto de software. Son una parte fundamental en su ciclo de vida y constituyen el estudio y determinación del método más adecuado para el desarrollo.

En la Universidad de las Ciencias Informáticas, este proceso está enmarcado por la utilización de la metodología AUP-UCI, que es una variante del proceso Unificado Ágil (AUP, por sus siglas en inglés) de Scott Ambler y a su vez es una versión simplificada del Proceso Unificado de Rational (RUP). AUP describe de manera simple la forma de desarrollar aplicaciones de software de negocio, con el uso de técnicas ágiles y conceptos, que aún se mantienen válidos en RUP. El objetivo de hacer una variación de la metodología AUP es el de crear un entorno de trabajo, que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

AUP-UCI propone tres fases (Inicio, Ejecución y Cierre) para el desarrollo de software y 11 roles por el cual debe guiarse el equipo de desarrollo para lograr una mayor organización. Propone también 7 disciplinas (Modelado de Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación) al igual que AUP; pero a un nivel más atómico. Una vez definidos los requisitos en el modelado de negocio, (Universidad Autónoma del Estado de Hidalgo, 2022) AUP-UCI define 4 escenarios para la modelación de éstos, publicados por Sánchez, T.R.

- Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU (Sánchez, 2015).

De los cuatro escenarios que se especifican en AUP-UCI se selecciona el escenario No2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que realizan las actividades. Así modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

Una vez identificada la metodología a utilizar, se prosigue a describir los lenguajes y herramientas utilizadas en el proceso de desarrollo del software.

1.5 Lenguaje, Herramientas y tecnologías

Un lenguaje es un conjunto de símbolos y reglas estandarizados que se utilizan para modelar y desarrollar parte de un diseño de software orientado a objetos. Comúnmente son utilizados en combinación con una metodología de desarrollo de software para llegar de una especificación inicial a la implementación.

Lenguaje unificado de modelación (**UML**) 2.4: es un lenguaje destinado al modelado de sistemas y procesos. Está basado en la orientación a objetos, lo cual condujo a la creación de lenguajes de programación como Java, C++, C#, o Smalltalk. El objetivo de UML es la unificación de los métodos de modelado de objetos, por medio de la identificación y definición de la semántica de los conceptos fundamentales y la elección de una representación gráfica con una sintaxis simple, expresiva e intuitiva. En la actualidad es promovido por el OMG (Object Management Group) y se ha convertido en un lenguaje de modelación muy extendido, sobre todo por su riqueza semántica, que lo abstrae de numerosos aspectos técnicos. El UML es utilizado para el modelado de los diagramas, que se realizan durante el análisis, diseño e implementación de la propuesta de solución.

Lenguaje de Marcado Extensible (**XML**) 1.0: por definición, es un lenguaje de marcado textual, lo que significa que, a diferencia del caso de RDBMS, los elementos de datos están ordenados por naturaleza. La información de esquema proporcionada externamente, que puede o no estar presente, ayuda a evitar coacciones excesivas y costosas entre tipos de datos. Además, para hacer frente a la estructura de árbol de los documentos XML y la complejidad resultante, las

relaciones jerárquicas entre datos, se utilizan las expresiones de ruta regulares. Las referencias se pueden usar para modelar las relaciones que exceden las limitaciones de las estructuras de árbol y requieren lógicas de mapeo adicionales. Se utiliza para la implementación de las interfaces que utilizará el usuario.

Visual Basic para Aplicaciones (**VBA**) es un lenguaje de programación de Office con el que es posible crear aplicaciones nuevas y funciones personalizadas para ahorrar tiempo en programas como Excel. En este sentido, tiene como principal virtud la capacidad de automatizar casi todas las operaciones de esta aplicación, algo que lo convierte en uno de los mejores recursos a la hora de eliminar tareas repetitivas. No obstante, todo depende de la creatividad del usuario y de sus conocimientos.

.NET es una plataforma de código abierto para crear aplicaciones de escritorio, web y móviles que se pueden ejecutar de forma nativa en cualquier sistema operativo. El sistema .NET incluye herramientas, bibliotecas y lenguajes que admiten el desarrollo de software moderno, escalable y de alto rendimiento. Una comunidad de desarrolladores activa mantiene y apoya la plataforma .NET. (Saasradar, 2022)

Microsoft Access, es un programa que se encarga de la administración de las bases de datos. El mismo te ayudará a organizar y administrar un buen número de registros. Con esta herramienta también podrás diseñar aplicaciones que se puedan emplear en Access, en la modalidad Backend, con el podrás recopilar, clasificar y además manejar un buen número de datos.

JavaScript es un robusto lenguaje de programación que se puede aplicar a un documento HTML (Fernández, 2023) y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla.

Visual Paradigm 8.0: es una de las herramientas CASE utilizadas para el modelado de software. Esta herramienta cuenta con los medios necesarios para extender sus funcionalidades, pues da soporte a las extensiones de aplicación. Provee de forma libre una interfaz de programación, que permite a los desarrolladores implementar y reutilizar clases e interfaces, así como, desarrollar funciones agregadas, que son

útiles para el desarrollo de software. Se utiliza para la realización del modelado de las clases y sus relaciones a través de UML.

Conclusiones del capítulo

El estudio de los principales conceptos asociados a la investigación, permitió definir la estructura básica de la Aplicación. Además el análisis de los sistemas de gestión empresarial permitió obtener un mayor conocimiento para la definición de la propuesta de solución, así como sus principales funcionalidades, que consistió en el desarrollo de una aplicación desktop para el proceso de Contratación y Facturación de la UEB Soluciones Constructivas y Acabados. Se seleccionó el Escenario 2 de la metodología AUP-UCI como guía del proceso de desarrollo del ciclo de vida del software porque este escenario se aplica a proyectos donde el objetivo primario es la gestión y presentación de información.

CAPÍTULO II: DISEÑO DE LA SOLUCIÓN PROPUESTA AL PROBLEMA CIENTÍFICO

En el presente capítulo se analizan los procesos que se deben realizar al gestionar contratos y facturas en la UEB y se define una propuesta de solución al problema planteado. Se identifican y especifican los requisitos funcionales y no funcionales y se brinda un acercamiento a la implementación a través de la fase de elaboración, construyéndose para ello el modelo de datos. Además, se selecciona el estilo arquitectónico, se modela la arquitectura y se realiza el diagrama de despliegue para mostrar la distribución física del sistema.

2.1 Análisis del negocio

El modelo conceptual en ingeniería de software permite representar de forma abstracta los conceptos y hechos relevantes del dominio del problema, para transformarlos posteriormente en un esquema de una base de datos concreta. A continuación, se muestra el diagrama de modelo conceptual.

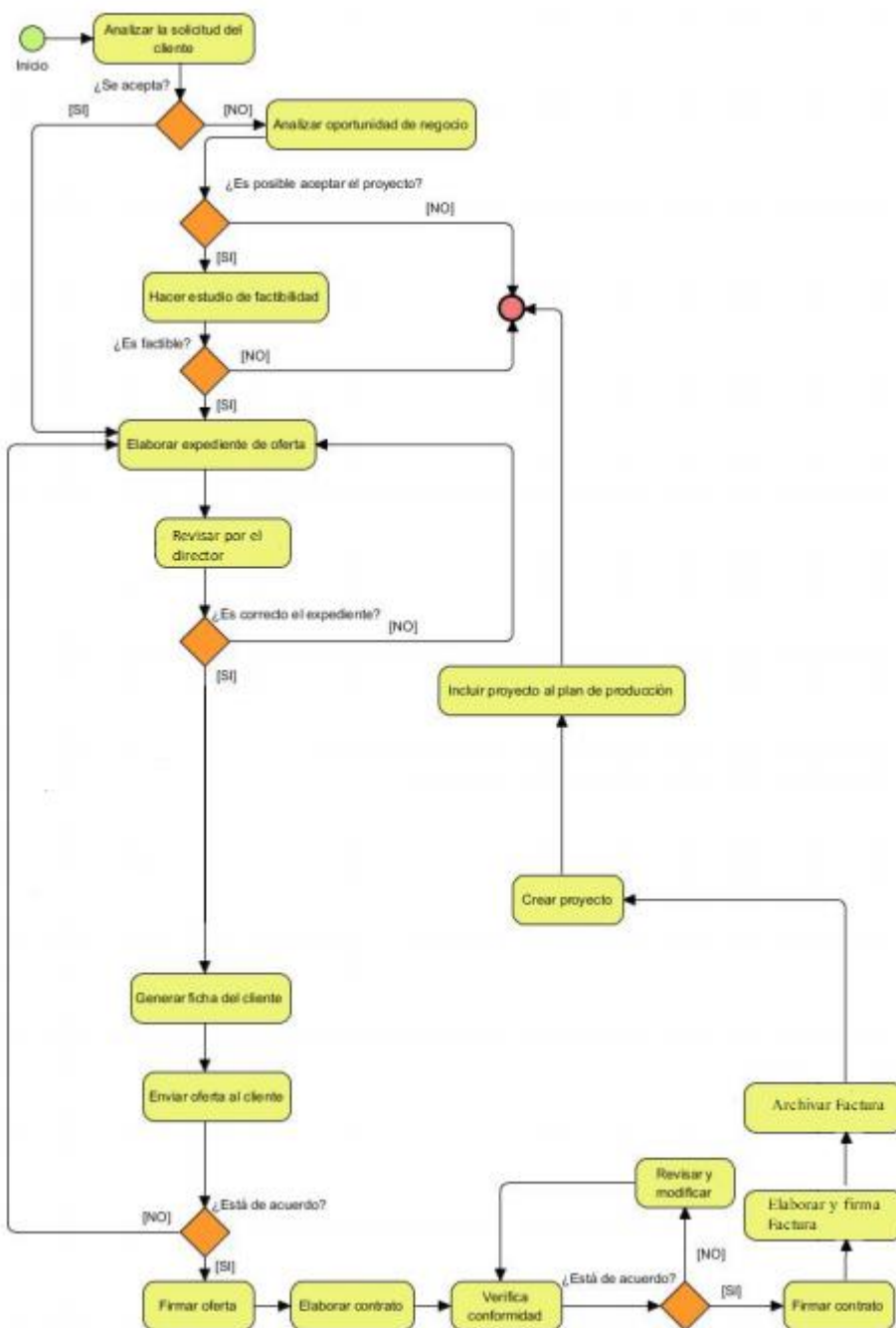


Figura 1. Proceso de contratación
(Elaboración propia)

Legenda de la figura 1

Oferta: servicios que se le van a vender al cliente

Factura: Documento legal que le informa al cliente cuanto pagar por los servicios

Contrato: obligación legal, relación cliente-entidad

Proyecto: futura obra en ejecución

La figura 1 describe el proceso de contratación y facturación para ello el cliente lleva sus necesidades al contratista, el cual evalúa si es factible realizarle una oferta inicial según lo que quiere el cliente. Posteriormente esta oferta es analizada por el director quien aprueba la creación de una ficha de cliente y luego envía la oferta al cliente para su conformidad y firma. Se elabora el contrato y se verifica junto al cliente que todo esté correcto y se firma o se modifica el contrato hasta quedar conforme ambas partes. Se confecciona y envía a la firma del cliente la factura que luego es firmada por el especialista. Se archivan ficha, contrato y factura del cliente en un expediente que luego se guarda en un gabinete junto a otros.

2.2 Requisitos Funcionales

Los requisitos funcionales son servicios que debe proporcionar el sistema, manera en que debe reaccionar a entradas y cómo se debe comportar en situaciones particulares. Describen lo que el sistema debe hacer. A continuación, se listan los requisitos funcionales asociados a la propuesta de solución. (Visure Solutions., 2023)

Tabla 1. Especificación de requisitos funcionales

(Elaboración propia)

Requisitos funcionales	Nombre
	Gestionar ficha de cliente
RF1	Crear ficha de cliente
RF2	Editar ficha de cliente
RF3	Listar ficha de cliente
RF4	Eliminar ficha de cliente
RF5	Buscar ficha de cliente
	Gestionar Contrato

RF6	Crear Contrato
RF7	Editar Contrato
RF8	Listar Contrato
RF9	Eliminar Contrato
RF10	Buscar Contrato

2.2.1 Descripción de requisitos funcionales

La descripción de requisitos especifica con mayor detalle la acción que realiza el mismo en el sistema. Contiene los pasos o las actividades que se deberán realizar para darle cumplimiento al requisito, así como las restricciones a las que está sujeto y las posibles respuestas del sistema. Es una herramienta ampliamente utilizada (Casos de uso) en el análisis y diseño de un sistema informático. A continuación, se muestra la descripción del requisito funcional: Crear Contrato.

Tabla 2. Descripción de requisitos funcionales

	Descripción del CU
Precondiciones	El usuario se ha autenticado en el sistema y tiene permisos para realizar esta acción. Se ha creado en el sistema al menos una Ficha de Cliente
Flujo de eventos	
Flujo básico Crear contrato	
	Se selecciona la opción Crear Contrato
	Se introduce y/o selecciona los siguientes datos: – Numero de contrato

	– Cliente del contrato – Representante del contrato (si lo requiere) – Resolución del ministerio que lo valida como organismo perteneciente al mismo – Fecha de dicha resolución – Domicilio legal – Código reup – Cuenta bancaria cup – Banco cup – Dirección del banco cup – Cuenta bancaria mlc – Banco mlc – Dirección del banco mlc – Licencia bancaria – Representado por (Nombre de la Persona) – Cargo – Resolución del cargo – Fecha de la resolución del cargo – Organismo que la emite (resolución del cargo) – Objeto del contrato – Valor en cup – Porcentaje a pagar por adelantado en cup – Valor en mlc
--	--

	– Porcentaje a pagar por adelantado en mlc – Microlocalización – Dirección de esa microlocalización – Vigencia – Alcance del contrato – Trabajos a realizar (posibilidad de insertar una tabla como anexos)
	Se selecciona la opción Guardar.
	El sistema valida * los datos introducidos.
	Si los datos son correctos el sistema los registra
	El sistema confirma el registro de los datos.
	Concluye el requisito
Pos-condiciones	
	Se registró en el sistema un nuevo contrato
Flujos alternativos	
Flujo alternativo. Cuando se introducen datos incorrectos.	
	El sistema señala los datos incorrectos y permite corregirlos.
	El usuario corrige los datos
Pos-condiciones	
	N/A
Flujo alternativo 7.b Cuando se dejan datos vacíos y son obligatorios.	
	El sistema señala los datos incorrectos y permite corregirlos.

	El usuario corrige los datos
Pos-condiciones	
	N/A
Flujo alternativo. El usuario cancela la acción	
	Flujo alternativo *.a El usuario cancela la acción
Pos-condiciones	
	N/A
Validaciones	
	Se valida que el Cliente muestre una lista de los clientes registrada en el sistema.
	Se valida que en representante se muestre una lista con los nombres de los representantes legales.
	Se valida que en los campos, Cup, Mlc, Porciento CUP y Porciento Mlc solo se escriban números
	Se valida que la Fecha de firma del contrato tenga el formato DD/MM/AAAA.
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito. Por ejemplo, estándares de intercambio de información.
Asuntos pendientes	Posibles mejoras al requisito.
Prototipo elemental de interfaz gráfica de usuario	

INFORMES

Contratos

Suplementos

IP 201_TI

IP 203_TP

IP 204_TA

IP 205_TFT/TPL

IP 206_TC

IP 301_TM

IP 301_TE

IP GENERAL



JUNO

OMFC Version 1.01 240820

Diseñado para la UEB Soluciones Constructivas y Acabado
Departamento de Operaciones.

Autor: Tec. Darío Alejandro Cánovas Losada

Agradecimientos: Dra.C. Marcia Losada. Dpto. Operaciones:
Ing. Msc. Victor Recio, Lic. Danayze Martínez, Tec. Keila
Rodríguez, Tec. Abel Rodríguez, Tec. Daylin Pinto y Tec. Yilian
Domenech.

Ficha

Suplementos

Facturas

Contratos

Y DE OTRA PARTE: Contrato de Prueba*	
personificada por _____	
creada por Resolución _____	de Fecha: 8/25/2020 del Organismo _____ con domicilio legal en: _____
registrada con el Código REEUP: 12070	
Cuenta Bancaria en CUP: _____	en el Banco _____ sito en: _____
y Cuenta en CUC: _____ en el Banco _____	
sito en: _____ Licencia Bancaria _____	
representado en este acto por eiem 2 _____ en su carácter de _____	
Director UEB _____ quién se encuentra debidamente acreditado por la Resolución No. 12/23	
de Fecha: 10/18/2020 emitida por Ejemplo1 _____ en lo adelante EL CLIENTE.	
AMBAS PARTES: Reconociéndose la personalidad y representación con que comparecen, convienen en suscribir el presente CONTRATO, en los términos y condiciones específicas que aquí se pactan.	
1. OBJETO DEL CONTRATO.	
1.1 El PRESTADOR se obliga a realizar y cobrar, y EL CLIENTE a recibir y pagar, bajo las condiciones aquí estipuladas, en cuanto a precios, especificaciones técnicas y particularidades que se acuerden en este Contrato, los servicios como se detalla en el Anexo no.1:	
stro: 14 1 de 159 ▶▶▶* Sin filtro Buscar	

2.3 Requisitos no Funcionales

Los requisitos no funcionales describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema; incluyen restricciones como el tiempo de respuesta, la precisión, recursos consumidos y seguridad.

A continuación, se listan los requisitos no funcionales asociados a la propuesta de solución.

Tabla 3. Requisitos no Funcionales

No.	Requisitos no funcionales
Restricción	
RnF 1	La base de datos debe permitir almacenar datos pasivos con un máximo de hasta 5 años.
RnF 2	Debe ser un fichero transportable y compacto.

RnF 3	El sistema debe permitir la exportación o importación de ficheros en formato Excel o PDF.
RnF 4	La aplicación debe ejecutarse sobre sistema operativo Microsoft Windows 7 como mínimo.
RnF 5	La aplicación debe ejecutarse sobre Microsoft Office 2010 como mínimo
RnF 6	Se empleará Microsoft Access como Gestor de Base de Datos
RnF 7	El sistema se desarrollará usando VBA
RnF 8	Espacio mínimo del disco duro: 3000 MB Procesador mínimo: 2 GHz RAM mínima: 3 GB Plataforma: PC
Calidad	
RnF 9	El componente de la interfaz de usuario debe ejecutarse en un entorno desktop.
RnF 10	El sistema debe definir un control de acceso básico.
RnF 11	Todos los usuarios del sistema operarán sobre este una vez se hayan Autenticado,

2.4 Casos de uso del Sistema

Un caso de uso de sistema es una secuencia de acciones que un sistema lleva a cabo que da lugar a un resultado de valor observable para un actor particular (alguien o algo fuera del sistema que interactúa con el sistema). (IBM, 2023)

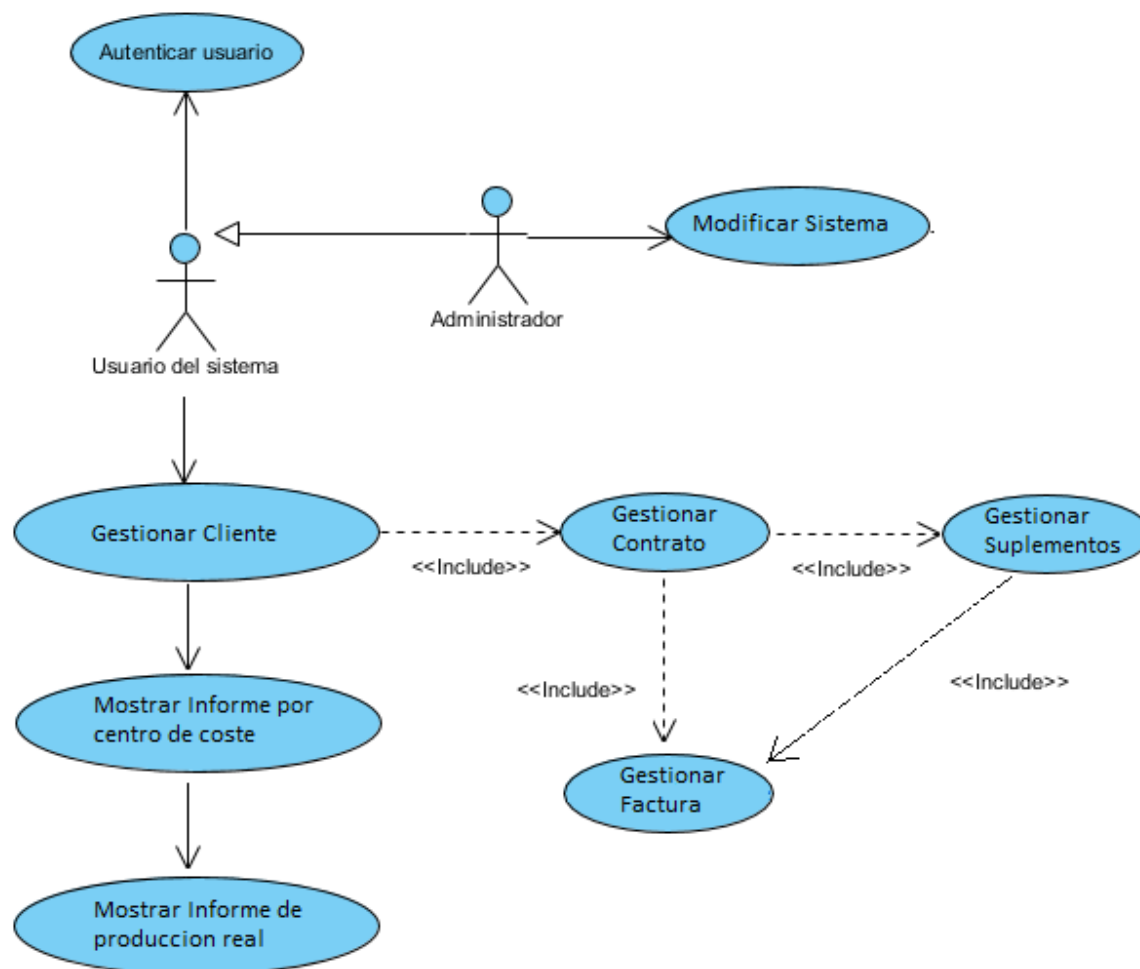


Figura 2. Casos de Uso del Sistema

A continuación se menciona el caso de uso Gestionar Contrato:

Tabla 4. Descripción del Caso de uso Gestionar Contrato

Objetivo	Gestionar Contrato
Actores	Usuario: (Inicia) Inserta, edita, elimina los datos de un Contrato en el sistema.
Resumen	El caso de uso inicia cuando el usuario del sistema selecciona la opción Contrato, el cual puede insertar, editar, listar y eliminar Contrato. El sistema realiza la acción seleccionada por el usuario y finaliza el caso de uso.

Complejidad	Alta	
Prioridad	Alta	
Precondicio- nes	El usuario debe estar autenticado en el sistema y debe poseer los permisos necesarios para ejecutar las acciones.	
Postcondicio- nes	Se inserta, edita, lista y elimina una Contrato en el sistema.	
Flujo de eventos		
Flujo básico Gestionar Cliente		
Actor		Sistema
1.	Selecciona Contrato en el menú	
2.		Permite realizar las siguientes acciones: ✓ Insertar Contrato. Ver sección 1: Insertar Contrato ✓ Editar Contrato. Ver sección 2: Modificar Contrato. ✓ Listar Contrato. Ver sección 3: Listar Contrato. ✓ Eliminar Contrato. Ver sección 4: Eliminar Contrato.
3.		Termina el caso de uso.

2.5 Arquitectura y Diseño de Software

La arquitectura y diseño del software se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, a la hora de realizar una pieza de software de cierta complejidad y como paso previo a cualquier implementación. De esta forma se dispone de una guía detallada que nos permite entender cómo van a encajar cada una de las piezas de este producto o servicio.

En arquitectura se llama patrón a cualquier solución general y reutilizable para problemas recurrentes en ingeniería del software en un contexto dado, son similares a los patrones usados en la programación, pero orientados específicamente a la estructura a un nivel superior y más genérico. Esta permite planificar nuestro desarrollo y elegir mejor un conjunto de herramientas para llevar a cabo nuestro proyecto (OpenWebinars, 2022).

Patrones arquitectónicos del software

Es una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales. Esta práctica se usa para identificar condiciones de error y problemas en el código que pueden no ser evidentes en ese momento.

Estos ayudan asegurarse de la validez del código, ya que son soluciones que funcionan y han sido probados por cuantiosos desarrolladores siendo menos propensos a errores (Miriam, 2020).

Modelo -Vista Controlador: La herramienta está planteada con una arquitectura basada en el Modelo Vista Controlador (MVC). Lo que permite manejar de forma independiente las actividades encargadas de las vistas y las clases controladoras que como su nombre da a relucir están encargadas de controlar el funcionamiento interno del sistema.

En este caso específico, el modelo-vista-controlador, tiene como principal objetivo separar los datos de una aplicación, la interfaz de usuario y la lógica de negocios

en tres componentes distintos que se relacionarán para tener como resultado la herramienta final (Mozilla.org, 2023).

- **Modelo:** El componente maneja lo referente a la persistencia de datos de la herramienta, las clases entidades el acceso a los datos, la seguridad de los datos y los elementos necesarios para manejarlos.
- **Vista:** El componente representa la interfaz gráfica para la interacción con el usuario.
- **Controlador:** Componente que contiene las clases que interactúan con la Vista recibiendo las solicitudes de eventos de los usuarios y con el Modelo registrando los cambios realizados por el mismo.

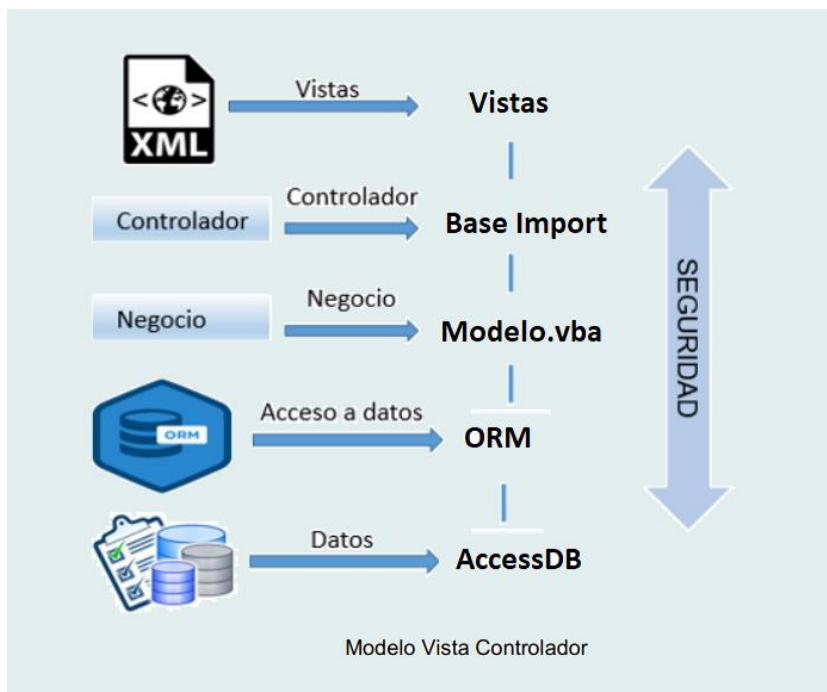


Figura 3. Modelo Vista Controlador

Modelo de datos lógicos

Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos.

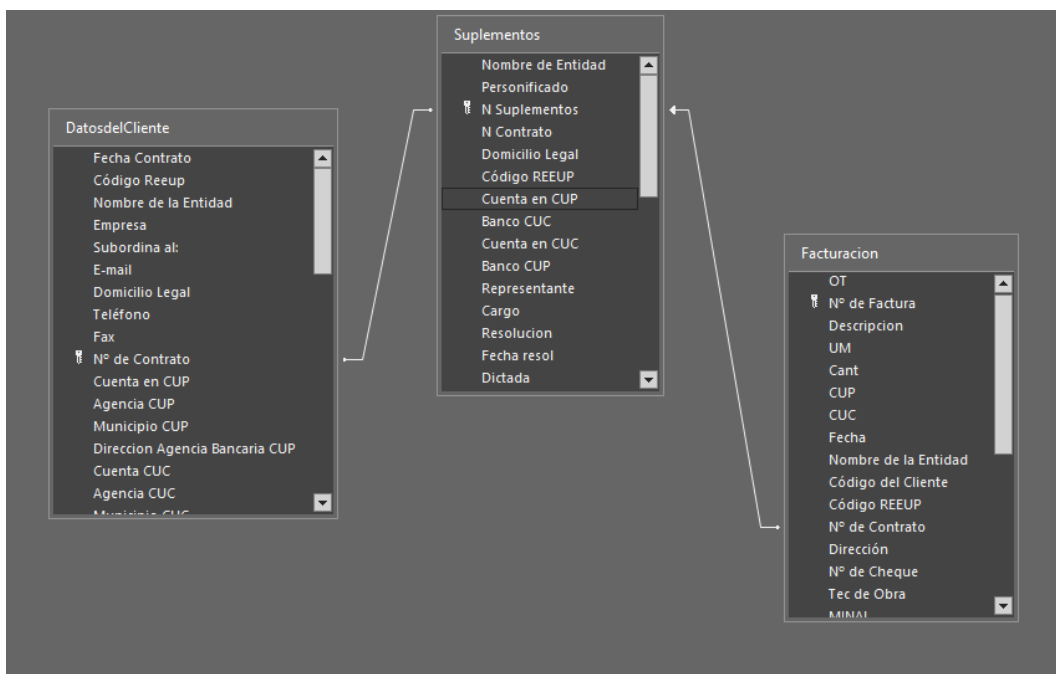


Figura 4. Modelo de Datos

Las características de un modelo de datos lógicos incluyen:

- Todas las entidades y relaciones entre ellos.
- Todos los atributos para cada entidad están especificados.
- La clave principal para cada entidad está especificada.
- Se especifican las claves externas (claves que identifican la relación entre diferentes entidades).

Los pasos para diseñar el modelo de datos lógicos son los siguientes:

- Especifique claves primarias para todas las entidades.
- Encuentra las relaciones entre diferentes entidades.
- Encuentra todos los atributos para cada entidad.
- Resuelva las relaciones de muchos a muchos.
- Normalización.

La siguiente figura el modelo de datos:

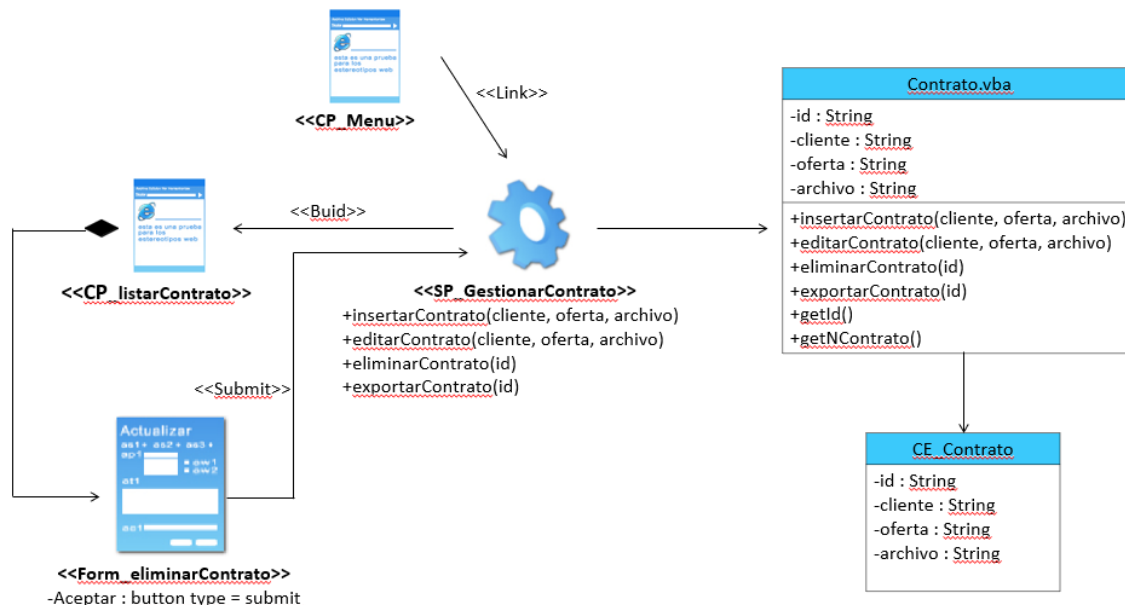


Figura 5. Diagrama de Clase del caso de Uso Gestionar Contrato

Patrones de diseño

En la ingeniería de software, un patrón constituye el apoyo para la solución a los problemas más comunes que se presentan durante las diferentes etapas del ciclo de vida del software. Los patrones de diseño describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos. En términos generales es posible decir que un patrón de diseño es una solución a un problema recurrente en el diseño de software (Guerrero et al., 2013).

Patrones GRASP

Los patrones GRASP (General Responsibility Assignment Software Patterns) son un conjunto de patrones de diseño que se utilizan en el desarrollo de software orientado a objetos para asignar responsabilidades de diseño adecuadas a las clases y objetos. Los patrones GRASP utilizados en la propuesta de solución son:

Patrón Experto: Se utiliza en la asignación de responsabilidades a las clases que cuentan con la información necesaria para llevarla a cabo. Este patrón se emplea

en las clases (Cliente, Contrato, etc.) que contienen la información necesaria para cumplir la responsabilidad asignada.

```
class Servicios(models.Model):
    oferta = models.ForeignKey(Oferta, related_name="servicio", on_delete=models.CASCADE)
    usuarioServicios = models.ForeignKey(User, related_name="usuarioServicios", on_delete=models.CASCADE)
    nombreArchivo = models.CharField(max_length=100)
    archivo = models.FileField(upload_to='serviciotempfile/', null=False)
    peso = models.CharField(max_length=20)
    subidoPor = models.CharField(max_length=100)
    class Meta:
        db_table="DatosCliente"
```

Figura 6. representación del Patrón Experto

Patrón Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos (Larman, 2002). Este patrón se emplea en las clases (insertarOferta, insertarServicio, etc.) para crear instancias de medios ya que tienen la información necesaria para ello.

```
def insertarServicios(request):
    cli=DatosClientes.objects.all()
    datos={'DatosClientes': cli}

    if request.method=='POST':

        rec=Servicios()
        rec.oferta_id=request.POST.get('idOferta')
        rec.peso=request.POST.get('peso')
        rec.subidoPor=request.user
        rec.usuarioServicios=request.user
        uploadedFile = request.FILES[archivo]
        rec.nombreArchivo = uploaded_file.name
        rec.archivo = uploaded_file

        rec.save()
        return redirect('listarServicios')
    else:
        return render(request, "servicio/insertarServicios.html, datos)
```

Figura 7. Representación del Patrón Creador

Alta Cohesión: Asigna responsabilidades de manera que una clase no realice un trabajo excesivo. Este patrón se evidencia en todas las clases a razón de que, en cada clase solo se implementan las funcionalidades que le corresponden.

```
def listarServicios(request):
    rec=Servicios.objects.filter(usuarioServicio=request.user)

    datos = {servicios : rec}
    return = render(request, servicios/listarServicios.html, datos)
```

Figura 8. Representación del Patrón Alta Cohesión

Figura xxx Representación del patrón alta cohesión

Bajo Acoplamiento: Este patrón es una medida de la fuerza con que una clase está

conectada a otras, las conoce y recurre a ellas (Larman, 2002). Se muestra en las clases (Servicios, Oferta, Factura, etc.) porque solo se relacionan con las clases necesarias y si se modifican no afectan a las demás.

```
class Factura(models.Model):

    factura = models.CharField(max_length=100)
    responsable = models.CharField(max_length=100)
    usuarioFactura = models.ForeignKey(User, related_name="usuarioFactura", on_delete=models.CASCADE)
    class Meta:
        db_table="Facturacion"
```

Figura 9. Representación del Patrón Bajo Acoplamiento

Patrones GOF

Los patrones GOF (*Gang of Four*) se refieren a un conjunto de patrones de diseño de software propuestos por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su libro "*Design Patterns: Elements of Reusable Object-Oriented Software*" publicado en 1994. Estos patrones son soluciones comunes a problemas de diseño que se encuentran frecuentemente en el desarrollo de software. El patrón GOF utilizado en la propuesta de solución es:

Factory Method: Centraliza en una clase constructora la creación de objetos de un subtipo de un tipo determinado, ocultando al usuario la casuística para elegir el subtipo que crear (Gloria et al., 2011). Se utiliza para crear objetos sin especificar su clase exacta. En la propuesta de solución se utilizó para crear una Notificación utilizando en canal seleccionado:

```

public class Notificacion{
    public Notificacion createNotificacion(String channel)
    {
        if (channel == null || channel.isEmpty())
            return null;
        switch (channel) {
            case "SMS":
                return new SMSNotificacion();
            case "EMAIL":
                return new EmailNotificacion();
            case "PUSH":
                return new PushNotificacion();
            default:
                throw new IllegalArgumentException("Canal desconocido "+channel);
        }
    }
}

```

Figura 10. Representación del Patrón Factory Method

Diagrama de despliegue

Los diagramas de despliegue UML muestran cómo los componentes de software se despliegan físicamente en los procesadores; es decir, el diagrama de despliegue muestra el hardware y el software en el sistema, así como el middleware usado para conectar los diferentes componentes en el sistema. En esencia, los diagramas de despliegue se pueden considerar como una forma de definir y documentar el entorno objetivo (Sommerville, 2011a).

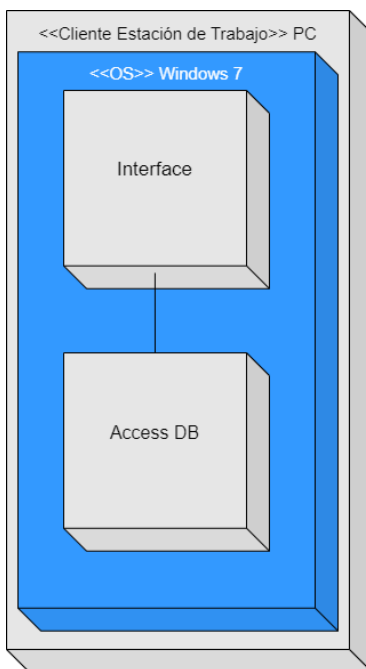


Figura 11. Diagrama de Despliegue

Conclusiones del capítulo

El modelo conceptual elaborado, a partir de los elementos o conceptos identificados, permitió representar y comprender los hechos más relevantes del dominio del problema. Descrita la propuesta de solución se pudo definir de forma más clara y detallada el problema a resolver. Fueron identificados un total de 10 requisitos funcionales y 11 requisitos no funcionales que responden a las características y necesidades del sistema de gestión. Los diagramas de casos de uso elaborados permitieron identificar y documentar los requisitos del sistema de manera clara y concisa. Se pudo comprender cómo los usuarios interactúan con el sistema y qué funcionalidades necesitan.

CAPÍTULO III: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

En el presente capítulo se realizan las pruebas internas y pruebas de aceptación teniendo en cuenta los requisitos definidos por el cliente. Las pruebas internas se realizarán utilizando las técnicas de caja blanca y caja negra y las pruebas de aceptación se realizarán con el cliente. Se aplica la técnica de ladov para la medición de la satisfacción de los usuarios con el sistema, y se describen los beneficios que aporta la solución al proceso de Contratación.

3.1 Estrategia de prueba

En la metodología AUP-UCI se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en 3 disciplinas:

Pruebas internas: En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas.

Pruebas de liberación: Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

Pruebas de aceptación: Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

En el caso del software en cuestión se aplicarán pruebas internas y de aceptación.

3.2 Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas

Unitarias:**Prueba de caja blanca**

Las pruebas de caja blanca son una filosofía de diseño de caso de prueba, que usan la estructura de control del diseño a nivel de componente para derivar los casos de pruebas. Mediante estos métodos, se pueden obtener casos de prueba que garanticen que se ejercite, por lo menos una vez, todos los caminos independientes de cada módulo; se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas; se ejecuten todos los bucles en sus límites y con sus limitaciones operacionales; y se ejerciten las estructuras internas de datos para asegurar su validez.

La técnica a aplicar es la de camino básico, la cual permite al diseñador de caso de prueba obtener una medida de complejidad lógica de un diseño de procedimiento, o usar esta como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Técnica del Camino Básico

Para obtener el conjunto de caminos independientes se construye el grafo de flujo asociado a una función y se calcula su complejidad. Para este caso, se tomó como ejemplo el método SoloNum (KeyAscii As Integer), encargado de verificar que el campo pedido solamente admita números enteros.

```
Private Sub SoloNum(KeyAscii As Integer)
1- If InStr("0123456789" & Chr(8), Chr(KeyAscii)) = 0
2- Then KeyAscii = 0
3- Else
    MsgBox "Solo números"
End If
End Sub
```

Figura 12. Prueba del camino básico

Los elementos del grafo son:

Nodos: son los círculos en el grafo, representan una o más secuencias del procedimiento.

Aristas: son las saetas y unen los Nodos representando el flujo de control del procedimiento.

Regiones: son las áreas delimitadas por las aristas y nodos.

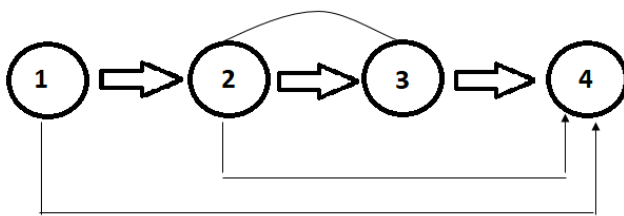


Figura 13. Nodos

Por último, se calcula la complejidad ciclomática:

La complejidad ciclomática es una medición de software con que se define la cantidad de caminos independientes del conjunto básico de un programa, brindando el número máximo de pruebas que se debe realizar para hacer que se ejecute cada sentencia al menos una vez.

La misma se calcula de la siguiente manera:

$$1. V(G) = (A - N) + 2$$

Siendo "A" la cantidad total de aristas y "N" la cantidad de nodos.

$$V(G) = (6 - 5) + 2$$

$$V(G) = 3.$$

$$2. V(G) = P + 1$$

Siendo "P" la cantidad de nodos predicados (son los nodos de los cuales parten dos o más aristas).

$$V(G) = 2 + 1$$

$$V(G) = 3.$$

$$3. V(G) = R$$

Siendo “R” la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más.

$$V(G) = 3.$$

Para este caso se obtienen 3 posibles caminos independientes y cantidad de pruebas que se deben realizar para comprobar que las sentencias se ejecuten al menos una vez.

Camino 1: 1, 2, 4.

Camino 2: 1, 2, 3, 4.

Camino 3: 1, 4.

Luego se elaboran los casos de prueba, quedando de la siguiente manera.

Tabla 5. Camino básico 1: 1-2-4

Descripción	Cuando el campo válido: valores numéricos.
Condición de ejecución	Se pretende comprobar la validez del campo.
Entrada	Que los datos sean correctos
Resultado	Queda registrado el pedido.

Tabla 6. Camino básico 2: 1, 2, 3, 4.

Descripción	Sucede cuando el campo es inválido: valores no numéricos.
--------------------	---

Condición de ejecución	Se pretende comprobar la validez del campo.
Entrada	Que los datos sean incorrectos
Resultado	No se registra la petición.

Tabla 7. Camino básico 3: 1, 4.

Descripción	Sucede cuando el campo está vacío
Condición de ejecución	Se pretende comprobar la validez del campo.
Entrada	No hay datos
Resultado	No se registra la petición.

Luego se ejecutan los casos de prueba para comparar los resultados obtenidos con los esperados, una vez comprobado que estos coinciden, se puede asegurar que todas las sentencias del método se han ejecutado al menos una vez.

Pruebas de Integración

Este tipo de pruebas evalúa la forma en que interactúan y operan varios módulos de aplicaciones de software de forma cohesiva. El sistema se divide en componentes conocidos como módulos o unidades.. Cada módulo es responsable de una tarea específica. El verdadero desafío llega cuando combinamos estos componentes para desarrollar todo el sistema de software.

Las pruebas de integración implican probar diferentes módulos de una aplicación de software como grupo: Módulo de contratos y Módulo de facturación. Una aplicación de software se compone de diferentes submódulos que trabajan juntos para diferentes funcionalidades. El propósito de las pruebas de integración es validar la integración de diferentes módulos juntos e identificar los errores y problemas relacionados con ellos.

3.3 Pruebas funcionales

Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema.

Prueba de Caja Negra

Las pruebas de caja negra o pruebas funcionales se realizan sobre la interfaz del software, comprobando las entradas y salidas de datos. Estas pruebas se realizan para comprobar que cada función es operativa, utilizando el artefacto diseño de caso de prueba, que tienen como objetivo introducir juegos de datos que ayuden a la ejecución de los casos y facilite que el sistema se ejecute en todas sus variantes. La técnica utilizada para desarrollar las pruebas de Caja Negra fue partición equivalente.

Técnica de Partición de Equivalencia

Un método que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirán la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Casos de Pruebas

Para los casos de pruebas se definieron un conjunto de estados válidos y no válidos para las condiciones de entrada del sistema (utilizando datos obligatorios).

Tabla 8. Casos de prueba

Escenario	Descripción	Numero de contrato	Empresa	Fecha	Vigencia	Personificado (campo no obligatorio)	Representado por	Respuesta del sistema	Flujo Central
EC: 1.1 Crear Contrato introduciendo datos validos	El sistema debe permitir crear un Contrato	V C-21-001/20	V Prueba	V 1/6/2020	V 2	V Prueba	V Prueba	El sistema registra Un nuevo contrato	1. Se selecciona la opción de Contrato 2. Se presiona el botón "Nuevo". 3. Se introducen los datos necesarios para crear un nuevo contrato. 4. Se selecciona el botón "Guardar"
EC: 1.2 Crear Contrato introduciendo datos inválidos	El sistema no debe permitir crear un Contrato	I Cx-21-001/20	V Prueba	V 1/6/2020	V 2	V Prueba	V Prueba	El sistema marca campo obligatorio formato incorrecto	1. Se selecciona la opción de Contrato 2. Se presiona el botón "Nuevo". 3.
		V C-21-001/20	V Prueba	V 1/6/2020	V 2	V Prueba	I Pru333eba		

		V C-21-001/20	V Prueba	V 1/6/2020	I cvf	V Prueba	V Prueba		Se introducen los datos necesarios para crear un nuevo contrato. 4.Se selecciona el botón "Guardar"
		V C-21-001/20	V Prueba	V 1/6/2020	V 2	I Pru33eba	V Prueba	El sistema crea el contrato	
EC: 1.3 Crear Contrato introduciendo datos vacíos	Se dejan campos vacíos al crear un Contrato	I vacío	V Prueba	V 1/6/2020	V 2	V Prueba	V Prueba	El sistema marca campo obligatorio vacío	1. Se selecciona la opción de Contrato 2.Se presiona el botón "Nuevo". 3. Se introducen los datos necesarios para crear un nuevo contrato. 4.Se selecciona el
		V C-21-001/20	V Prueba	V 1/6/2020	I vacío	V Prueba	V Prueba		
		V C-21-001/20	V Prueba	I vacío	V 2	V Prueba	V Prueba		
		V C-21-001/20	V Prueba	V 1/6/2020	V 2	I vacío	V Prueba	El sistema crea el contrato	

									botón "Guardar"
EC 1.4: Cancelar	Se cancela la operación de crea contrato	V (N/A)	V (N/A)	V (N/A)	V (N/A)	V (N/A)	V (N/A)	El sistema no realiza ninguna acción y cierra la pestaña	1. Se selecciona la opción de Contrato 2.Se presiona el botón "Nuevo". 3. Se introducen los datos necesarios para crear un nuevo contrato. 4.Se selecciona el botón "Cerrar"

Pruebas de Rendimiento:

Las pruebas de rendimiento deben diseñarse para garantizar que el sistema procese su carga pretendida. Esto implica efectuar una serie de pruebas donde se aumenta la carga, hasta que el rendimiento del sistema se vuelve inaceptable. Las pruebas de rendimiento se preocupan tanto por demostrar que el sistema cumple con sus requerimientos, como por descubrir problemas y defectos en el sistema (Sommerville, 2011b).

El ambiente de prueba estuvo conformado por:

- Sistema Operativo: Windows 7
- Microprocesador: Inter(R) Celeron (TM) CPU @2 0GHz
- Memoria RAM: 3 GB
- Disco Duro: 3 GB

Los resultados de las pruebas de rendimiento se consideran satisfactorios con respuestas inferiores a 1 segundo, con hardware mínimo especificado entre los requisitos no funcionales. Se demuestra que la propuesta de solución es estable, ya que se mantuvo prestando servicios todo el tiempo, sin incurrir en fallos.

Como parte de la ejecución de la prueba de caja negra, se identificó en la primera iteración 4 No Conformidades (NC) como faltas de ortografías y funciones incorrectas. Para la segunda iteración se realizaron correcciones a las 4 no conformidades pendientes de la iteración anterior y se comprobó que existe aún problemas con la admisión de caracteres raros en los campos no relacionados con las fechas. Por último, se corrigió el problema pendiente y se prosiguió al desarrollo de una tercera iteración arrojando resultados satisfactorios.

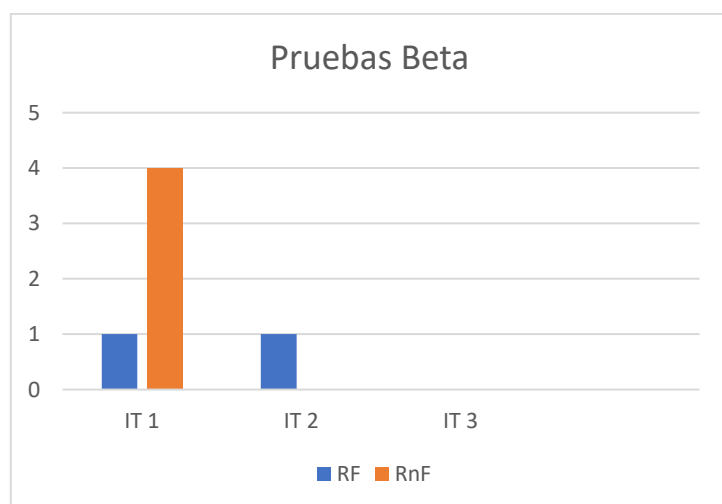
Pruebas Alpha y Beta

Las versiones alfa son las pruebas con una disponibilidad limitada que se pueden usar antes de que se lancen las versiones para el uso generalizado. El objetivo consiste en verificar su funcionalidad y recopilar los comentarios de un conjunto restringido de clientes. Normalmente, es necesario recibir una invitación para participar en las pruebas alfa, que están sujetas a las condiciones de uso previo a la disponibilidad general. Es posible que las versiones alfa no incluyan todas las funciones, no se ofrece ningún acuerdo de nivel de servicio y no hay obligación alguna de prestar asistencia técnica. No obstante, las versiones alfa son adecuadas por lo general para entornos de prueba. (Zaptest, 2023) En el caso del software en cuestión no se encontró ninguna dificultad en su fase alpha.

Los productos o las funciones en **fase beta** están listos para que los prueben o utilicen más clientes. Se suelen anunciar de forma pública. En las versiones beta no se ofrece ningún acuerdo de nivel de servicio, a menos que se indique lo contrario en los términos del producto o del programa beta en cuestión; no obstante, se ofrece una asistencia técnica limitada. Las fases beta duran unos seis meses de media.

En la primera iteración de requisitos funcionales 5 y no funcionales 12 se detectaron 1 no conformidad de los funcionales y 4 de los no funcionales.

En la segunda iteración quedaron resueltos los 4 no funcionales.



3.4 Aplicación de la Técnica de Iadov

Para validar esta investigación se utilizó la Técnica de Iadov. Esta constituye una vía indirecta para el estudio de la satisfacción, puesto que los criterios se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario aplicado y en cuya relación el sujeto se desconoce. Estas tres preguntas se relacionan a través de lo que se denomina como el “Cuadro Lógico de Iadov”.

El número resultante de la interrelación de las tres preguntas cerradas indica la posición de cada sujeto en la escala de satisfacción. A continuación, se muestra la escala:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Valores	Clasificación	Escala
+1	Clara satisfacción	A
0,5	Más satisfecho que insatisfecho	B
0	No definido y contradictorio	C
-0,5	Más insatisfecho que satisfecho	D
-1	Clara insatisfacción	E

Figura 14. Valores Prueba de Iadov

$$ISG = \frac{A (+1) + B (+0.5) + C (0) + D (-0.5) + E (-1)}{N}$$

Figura 15. Formula de ladov

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 o 6; 4; 5 y donde N representa el número total de sujetos del grupo. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción. Estos valores representados gráficamente en un eje, se aprecian de la forma siguiente:

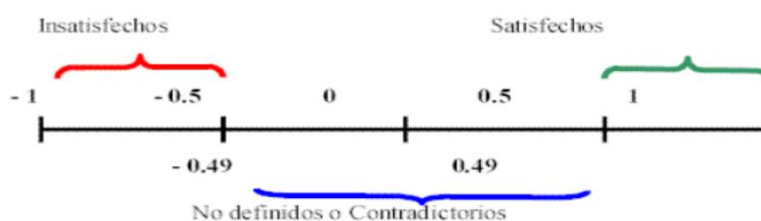


Figura 16. Rangos prueba de ladov

Tabla 9. Cuadro lógico de ladov de la investigación

¿Le satisface la concepción de este sistema para disminuir el tiempo en la obtención de la información utilizado para la gestión de contratos en la UEB Soluciones	¿Considera usted que es aconsejable no contar con un sistema que permita gestionar los contratos y facturas de la UEB Soluciones Constructivas y Acabados?		
	No	No sé	Si
	¿Utilizaría el Sistema de gestión desarrollado, los contratistas, para disminuir el tiempo en la obtención de la información utilizada en contratos para la empresa UEB Soluciones Constructivas y Acabados?		

Constructivas y Acabados?	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me gusta mucho	1	6	6	2	2	6	6	6	6
No me gusta tanto	2	2	3	2	3	2	6	3	6
Me da lo mismo	3	3	3	3	3	6	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	3	3	3	3	2	3	4

Total de cuestionados	9	Escala
Clara satisfacción	6	A
Más satisfecho que insatisfecho	1	B
No definida	1	C
Más insatisfecho que satisfecho	0	D
Clara insatisfacción	0	E
Contradictoria	1	C

Figura 17. Resultado obtenido al aplicar la encuesta

Al aplicar la fórmula para el cálculo del ISG se obtuvo el siguiente resultado:

$$ISG = \frac{6(1) + 1(0.5) + 2(0) + 0(0) + 0(0)}{9} = 0,72$$

Figura 18. Resultados formula de ladov

Luego de aplicar la técnica de ladov a través de una encuesta sobre el Sistema de Gestión para la Contratación y Facturación creado, donde se tuvo la participación de varios especialistas del centro, directamente relacionados con el proceso de

contratación, se obtuvo de los encuestados una valoración positiva con un alto Índice de Satisfacción Grupal como se observa en la Figura XX. El valor se encuentra comprendido entre 0,5 y 1, por lo que, se corrobora según la técnica de ladov, que el componente desarrollado se clasifica como Satisfactoria.

3.5 Beneficios de la aplicación desarrollada

El Sistema de Gestión para la Contratación y Facturación de ofertas de servicios a los clientes del GEIA. Este gestiona principalmente el proceso de contratación de servicios al cliente. Entre los principales beneficios que aporta el desarrollo del componente están:

- Es un sistema de gestión, lo que permitirá la contratación de forma sencilla sin necesidad de personal especializado
- Permite el acceso a datos del cliente y servicios contratados por la entidad.
- Disminuye el tiempo en la obtención de los documentos necesarios para realizar cada
- Contrato.

Conclusiones del capítulo

La definición de la estrategia de pruebas determino como se llevan a cabo los proceso de prueba de software. En este capítulo se ejecutaron las pruebas internas. El resultado de estas pruebas funcionales y no funcionales, permitió la corrección en tiempo y forma de errores y no conformidades detectadas. Se efectuó, además, la validación de la investigación mediante la técnica de ladov con la participación de especialistas del GEIA. El grado de satisfacción evaluado del componente fue de satisfactorio, lo que permite concluir que se cumplieron los objetivos planteados en la investigación.

CONCLUSIONES FINALES

- La estructura básica de la Aplicación fue definida a partir del estudio de los principales conceptos asociados a la investigación .
- La definición de la propuesta de solución partió del análisis de varios sistemas de gestión empresariales de los cuales se extrajeron las características comunes y positivas.
- La propuesta de solución resultó en el desarrollo de una aplicación desktop para el proceso de Contratación y Facturación de la UEB Soluciones Constructivas y Acabados.
- Se implementaron los módulos correspondientes en la aplicación para la localización de servicios asociados a la contratación y facturación lo cual disminuyó los tiempos de atención al cliente humanizando el trabajo.
- La selección del Escenario 2 de la metodología AUP-UCI como guía del proceso de desarrollo del ciclo de vida del software permitió concentrarse en la gestión y presentación de información.
- El modelo conceptual elaborado, a partir de los elementos o conceptos identificados, permitió representar y comprender los hechos más relevantes del dominio del problema. Los diagramas de casos de uso elaborados permitieron identificar y documentar los requisitos del sistema de manera clara y concisa. Se pudo comprender cómo los usuarios interactúan con el sistema y qué funcionalidades necesitan.
- Se determinó como se llevan a cabo los procesos de prueba de software mediante la definición de una estrategia de pruebas que unifica pruebas unitarias, de integración, funcional y de aceptación. Se efectuó, además, la validación de la investigación mediante la técnica de ladov arrojó un resultado satisfactorio, lo que permite concluir que se cumplieron los objetivos planteados en la investigación.

RECOMENDACIONES

La posibilidad de explorar otras plataformas teniendo en cuenta las restricciones de hardware del cliente.

REFERENCIAS BIBLIOGRAFICAS

- Amelot, M. (2013). VBA Access 2013: programar en Access. Ediciones ENI.
- Alexia Suite Educativa. (s. f.). Alexia Suite Educativa. (16 de 11 de 2022) obtenido de <https://www.alexiaeducaria.com/>
- Castaño Guevara, J. A. (2018). Centro de Tecnologías para la Formación (FORTES) | Uni-versidad de las Ciencias Informáticas. Centro de Tecnologías para la Formación (FORTES) | Universidad de las Ciencias Informáticas. <https://www.uci.cu/investigacion-y-desarrollo/centros-de-desarrollo/centro-de-tecnologias-para-la-formacion-fortes>
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). Non-functional requirements in software engineering (Vol. 5). Springer Science & Business Media.
- do Carmo Machado, I., McGregor, J. D., Cavalcanti, Y. C., & De Almeida, E. S. (2014). On strategies for testing software product lines: A systematic literature review. Information and Software Technology,
- MINJUS. (2013). *Decreto Ley No. 304 de Contratación Económica*. La Habana: My. Gral. Ignacio Agramonte y Loynaz.
- Fernández, Y. (22 de 11 de 2023). *Qué es el HTML5 y qué novedades ofrece* . Obtenido de <https://www.xataka.com/basics/que-html5-que-novedades-ofrece>
- Gestiopolis (12 de 11 de 2022) Estadística: Qué es, objetivos, tipos, partes, elementos •. (s. f.). Obtenido de <https://www.gestiopolis.com/que-es-estadistica-tipos-y-objetivos/>
- González., R. A. (2011). *El proceso de investigación científica* . Ciudad de La Habana: Editorial Universitaria.
- Hubspot ¿Qué es Angular? Características y ventajas. Recuperado (12 de 11 de 2022) Obtenido de <https://blog.hubspot.es/website/que-es-angular>

- IBM. (23 de 11 de 2023). *IBM*. Obtenido de <https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/lifecycle-management/6.0.3?topic=requirements-defining-use-cases>
- KeepCoding, R. (2022, septiembre 29). ¿Qué es el patrón de arquitectura MVVM? <https://keepcoding.io/blog/que-es-el-patron-de-arquitectura-mvvm/>
- Kumar, R., Maheshwary, P., & Malche, T. (2019). Inside agile family software development methodologies. *International Journal of Computer Sciences and Engineering*,
- Management Levels and Types | Boundless Management*. (19 de 11 de 2023). Obtenido de <https://courses.lumenlearning.com/boundless-management/chapter/management-levels-and-types/>
- Miriam. (2020, junio 24). Qué son los Patrones de Diseño de software / Design Patterns. Pro-file Software Services. <https://profile.es/blog/patrones-de-diseno-de-software/>
- Modelado del negocio. (s. f.). (12 de 11 de 2022), Obtenido de http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/351_modelado_del_negocio.html
- Mozilla.org. (22 de 11 de 2023). *Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web: MVC*. Obtenido de <https://developer.mozilla.org/es/docs/Glossary/MVC>
- Novoseltseva, E. (22 de 1 de 2023). *5 principales patrones de Arquitectura de Software*. Obtenido de <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/>
- Nubia Casquete Baidal, W. N. (19 de 6 de 2023). *Aplicación de las estadísticas en las empresas*. . Obtenido de *Revista Caribeña de Ciencias Sociales* (diciembre 2016): <https://www.eumed.net/rev/caribe/2016/12/estadisticas.html>
- OBJECT MANAGEMENT GROUP. (29 de 11 de 2023). *Componente de software: Definición de Componente de software y sinónimos de Componente de*

- software (español). Diccionarios y traductores sobre [diccionario.sensagent.com](https://diccionario.sensagent.com/Componente%20de%20software/es-es/) Obtenido de: <https://diccionario.sensagent.com/Componente%20de%20software/es-es/>.
- OMG (2011). *OMG Unified Modeling Language (OMG UML), Superstructure, V2.4* p.147.
- OpenWebinars. (13 de 11 de 2022). *Arquitectura de software: Qué es y qué tipos existen*. Obtenido de <https://openwebinars.net/blog/arquitectura-de-software-que-es-yque-tipos-existen/>
- PostgreSQL. (2020). Obtenido de <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>
- Pressman, R. S. (2010). *Software Engineering. A practitioner's Approach. 7th edition*. New York : McGraw Hill.
- Real Academia de la Lengua Española. (23 de 11 de 2023). *Diccionario de la lengua española, 22.ª edición*. Obtenido de <https://www.rae.es/drae2001/factura>
- REAL ACADEMIA ESPAÑOLA:. (23 de 11 de 2023). *Diccionario de la lengua española, 23.ª ed en linea*. Obtenido de <https://dle.rae.es/contrato>
- Redhat (14 del 11 de 2022,)¿Qué es la arquitectura orientada a los servicios? (s. f.). Obtenido de <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>
- Saasradar. (12 de 11 de 2022). *Marcos de trabajo o framework en diferentes etapas del desarrollo de software*. Obtenido de <https://saasradar.net/marcos-de-trabajo-oframework/>
- School, E. B. (2022). *Herramientas de programación | Euroinnova. Euroinnova Business School*. <https://www.euroinnova.edu.es/blog/herramientas-de-programacion>
- Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI. L a Habana, Cuba : s.n., 2015.*

Sampieri, R. H., & Lucio., C. F. (2003). *Metodología de la Investigación*. México, D. F.: McGraw-Hill Interamericana.

Santander Universidades(2020) Metodologías de desarrollo de software: ¿qué son?. Obtenido de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>

SG Buzz (13 de 11 de 2022) Obtención de Requerimientos. Técnicas y Estrategia. obtenido de <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>

Siigo. (23 de 11 de 2023). *¿Qué es un sistema de gestión empresarial?* Obtenido de <https://www.siigo.com/blog/empresario/que-es-un-sistema-de-gestion-empresarial/>

Statisticum Collegium. (19 de 6 de 2023) Aplicación de las estadísticas en las empresas.. Obtenido de <https://www.eumed.net/rev/caribe/2016/12/estadisticas.html>

Takanen, A., Demott, J. D., Miller, C., & Kettunen, A. (2018). Fuzzing for software security testing and quality assurance. Artech House.

Universidad Autónoma del Estado de Hidalgo. (12 de 11 de 2022). *Modelado del negocio*. Obtenido de http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/351_modelado_del_negocio.html

Visure Solutions. (13 de 11 de 2023). *Definición de requisitos: ¿Qué es y cómo aplicarlo? | Guía Completa - Soluciones Visure*. Obtenido de <https://visuresolutions.com/es/blog/definici%C3%B3n-de-requisitos/>

Westreicher, G., & López, J. F. (19 de 11 de 2023). *economipedia*. Obtenido de <https://economipedia.com/definiciones/centro-de-costos.html>

Zaptest. (23 de 11 de 2023). *Pruebas alfa: qué son, tipos, proceso, vs. pruebas beta, herramientas y mucho más*. Obtenido de <https://www.zaptest.com/es/pruebas-alfa-que-son-tipos-proceso-vs-pruebas-beta-herramientas-y-mucho-mas>

Anexos

Encuesta: Evaluación de la Técnica de ladov		
1	¿Considera usted que el sistema de gestión propuesto disminuye el tiempo en la obtención de la información utilizada en la gestión de contratos en la UEB?	Si__ No__
2	¿Considera usted que es aconsejable no contar con un sistema que permita gestionar los contratos y facturas de la UEB Soluciones Constructivas y Acabados?	Si__ No sé__ No__
3	¿Utilizaría el Sistema de gestión desarrollado, los contratistas, para disminuir el tiempo en la obtención de la información utilizada en contratos para la empresa UEB Soluciones Constructivas y Acabados?	Si__ No sé__ No__
4	¿Considera usted que la solución propuesta permite disminuir el tiempo en la obtención de la información utilizada en la gestión de contratos en la UEB?	Si__ No sé__ No__
5	¿Le satisface la concepción de este sistema para disminuir el tiempo en la obtención de la información utilizado para la gestión de contratos en la UEB Soluciones Constructivas y Acabados?	Si__ No sé__ No__