



UNICIT
UNIVERSIDAD IBEROAMERICANA
DE CIENCIA Y TECNOLOGIA



Miembro de Asociación de Universidades Privadas de Centroamérica y Panamá
Miembro de la Asociación Universitaria Iberoamericana de Postgrado



Informe Final

**Creación de Sistema de Control de Ventas y Facturación para el
área de restaurante en Asociación Club Terraza.**

Autor:

MSc. Silvia Ticay - Docente de la Facultad de Ingenierías y Arquitectura

Coautores:

Edmara Junieth Jarquín Malespín.
Javier Alejandro Moisés Flores Fonseca

Asesor:

MSc. José Dagoberto Mejía Flores – Director de Investigación y Posgrado

Fecha de Entrega: Junio 2022.

Resumen

El presente proyecto consiste en la creación de un sistema de control de ventas y facturación para el área de restaurante en Asociación Club Terraza que permita obtener una mayor eficiencia en el manejo y control de la información a nivel organizacional, ya que actualmente estos procesos se realizan de forma manual lo cual conlleva a cometer errores y pérdida de tiempo.

El sistema cuenta con formularios, consultas y reportes que permiten verificar de forma más eficiente las facturas y ventas además de contar con datos exactos de los productos disponibles. Se implementaron los módulos de facturación y control de ventas en el lenguaje de programación Visual Basic y la base de datos en el sistema manejador de bases de datos SQL Server.

Se realizó un análisis en cada proceso de facturación y control de ventas para lograr identificar los datos, requisitos, necesidades, requerimientos y la lógica de negocio para automatizar dichos procesos. Se diseñó la base de datos de acuerdo a los parámetros y funcionalidades del sistema de tal manera que el usuario obtenga información actualizada y precisa.

La investigación es aplicada de campo porque se logró resolver la problemática del sistema actual mediante la creación de un nuevo sistema. La metodología de desarrollo utilizada es el modelo clásico o cascada permitiendo conocer el objetivo de trabajo para la culminación de cada una de las fases que tiene este modelo.

Este software presenta una interfaz gráfica clara y concisa lo que permite al usuario familiarizarse con el entorno fácilmente. Agiliza el proceso de atención al cliente, evita errores de facturación y ayuda a obtener una mejor organización de la información.

Índice

I.	Introducción	1
II.	Antecedentes	2
III.	Objetivos	3
3.1	. Objetivo General	3
3.2	. Objetivos específicos	3
IV.	Preguntas de Investigación	4
V.	Justificación	5
VI.	Limitaciones	7
VII.	Supuestos Básicos	8
VIII.	Categorías, temas y patrones emergentes de la investigación	9
IX.	Marco teórico	10
A.	Estado del Arte	10
B.	Perspectiva teórica asumida	12
9.1.	Sistemas	12
9.2.	Clasificación de sistemas	12
9.3.	Sistemas de Información	13
9.4.	Actividades de los sistemas de información	13
9.6.	Tipos de Software	14
9.6.1.	Software de aplicación	14
9.6.2.	Software de programación	15
9.6.3.	Software de sistema	15
9.6.4.	Ingeniería de software	15
9.6.5.	Requerimientos del software	16
9.6.6.	Tipos de requerimientos	16
9.6.7.	Capas de la ingeniería de software	17
9.6.8.	Metodologías para el desarrollo de software	18
9.6.9.	Modelos de ciclo de vida de desarrollo de software	18
9.6.10.	Modelo en Cascada	18

9.6.11. Modelo en V	19
9.6.12. Modelo de desarrollo incremental	20
9.6.13. Modelo en Espiral	21
9.6.14. Modelo Iterativo	22
9.6.15. Modelo de Prototipo	23
9.7. Desarrollo de sistema	24
9.8. Facturación	24
9.9. Control de ventas	25
9.10. Sistema de control de ventas y facturación	25
9.11. Metodologías ágiles	26
9.11.1. Tipos de metodología ágiles	26
9.11.2. Ventajas de las metodologías ágiles	27
9.8. Lenguaje de programación	28
9.9. Programación orientada a objetos	31
9.9.1. Pilares fundamentales de la programación orientadas a objetos	31
9.9.2. Ventajas de la programación orientadas a objetos	32
9.10. Visual Basic	33
9.11. Base de Datos	34
9.12. Tipos de modelos de base de datos	35
9.13. Sistemas de gestión de base de datos	38
9.13.1. Sistemas de gestión de bases de datos relacionales	38
9.13.2. Sistemas de Gestión de base de datos no relacionales	39
9.14. Normalización	39
9.14.1. Ventajas de la normalización de datos	40
9.15.1. Niveles de normalización de una base de datos	40
9.15.2. Primera forma normal	41
9.15.4. Tercera forma normal	41
X. Diseño metodológico	42
10.1. Enfoque cualitativo asumido y su justificación	42

10.2. Población	42
10.3. Muestra teórica y sujetos del estudio	42
10.4. Métodos y técnicas de recolección de datos utilizados	43
10.5. Criterios de calidad aplicados: credibilidad, confiabilidad y triangulación	43
10.6. Métodos y técnicas para el procesamiento de datos y análisis de información	44
XI. Discusión de resultados o hallazgos	46
11.1. Fase 1: Análisis del sistema	46
11.1.1. Definición del problema	46
11.1.2. Proceso de facturación	47
11.2. Identificar las necesidades del cliente	47
11.2.1. Análisis técnicos y económicos	47
11.3. Viabilidad del sistema	47
11.4. Viabilidad Técnica	48
11.5. Viabilidad Económica	48
11.6. Viabilidad operacional	49
11.7. Funciones del sistema	50
11.8. Fase 2: Análisis de los requisitos del software	51
11.8.1. Requisitos solicitados por el cliente	51
11.9. Fase 3: Diseño del software	53
11.9.1. Diseño de la base de datos	53
11.10. Diccionario de datos	60
11.11. Diagrama Entidad- Relación	65
11.12. Diseño de interfaz de Usuario	66
11.13. Fase 4 Desarrollo del software	72
11.13.1. Programación	72
11.13.2. Codificación	73
11.13.3. Documentación	73
11.14. Fase 5: Prueba y mantenimiento	73

11.15. Fase 6: implementación del software	74
XII. Referencias Bibliográficas	75
XIII. Anexos	79

Índice de Ilustraciones

Ilustración 1. Capas de la ingeniería de software	17
Ilustración 2. Modelo de cascada.	19
Ilustración 3. Modelo en V	20
Ilustración 4. Modelo de desarrollo incremental	21
Ilustración 5. Modelo en espiral	22
Ilustración 6. Modelo Iterativo	23
Ilustración 7. Modelo de prototipo	24
Ilustración 8. Relación de las formas normales	40
Ilustración 9. Proceso de facturación	47
Ilustración 10. Diagrama Entidad - Relación	66
Ilustración 11. Login del sistema.	66
Ilustración 12. Menús principales del sistema. Con la sección Herramientas e Inicio.	67

Índice de Tablas

Tabla 1. Estructura organizativa en Asociación Club Terraza	42
Tabla 2. Componentes Físicos de la empresa	48
Tabla 3. Presupuesto proyecto	48
Tabla 4. Resultados técnicos PC	49
Tabla 5. Presupuesto de implementación del sistema	49
Tabla 6. Formulario de análisis de proceso de facturación	52
Tabla 7. Datos recopilados Facturación	54
Tabla 8. Productos	54
Tabla 9. Detalle Venta	55
Tabla 10. Ventas	55

I. Introducción

Asociación Club terraza es un club social, ubicado en la ciudad de Managua Nicaragua. Tiene como misión brindar a sus socios, con el mayor nivel de excelencia, los servicios y actividades en bares, restaurantes, eventos, áreas de entretenimiento y deportes, en un ambiente de confort, ameno, seguro y con calidad de clase mundial. Actualmente carece de un sistema automatizado de control de ventas y facturación.

Un sistema de control de ventas ayuda a recopilar información sobre los ingresos reales y comparar los resultados. Un sistema de facturación es un software informático para gestionar y controlar el envío y recepción de facturas de bienes y servicios que brinda la empresa. Son de vital importancia para un buen manejo e inspección de los recursos económicos que puede generar la empresa.

La creación del sistema de control de ventas y facturación para el área de restaurante en Asociación Club Terraza, inicia con 3 fases importantes del ciclo de vida del sistema. Se efectuó el análisis y recolección de la información de los procesos de facturación manuales, que ponían en práctica al momento de realizar dicho proceso. Una vez que conocemos el desempeño del sistema, se creó la base de datos, se realizó el diseño para solventar las carencias que se presentaban al hacer uso de métodos manuales. Al tomar los requisitos y los productos de la fase de desarrollo, se implementó este sistema en el área de restaurante en Asociación Club Terraza realizando pruebas de validación, las cuales resultaron satisfactorias.

Club Terraza ha decidido invertir en el desarrollo de un software que pueda suplir la demanda actual en los procesos de facturación y control de ventas, que se llevan a cabo en el área de restaurante. Cuenta con un sistema manual de control de ventas y facturación el cual ha venido presentando diversos inconvenientes que pueden llegar a demorar los tiempos de atención.

Debido a esta problemática se desarrolló un nuevo sistema automatizado de control de ventas y facturación logrando solventar la situación que ocurría al momento de realizar estos procesos, obteniendo mayores beneficios, al contar con herramientas automatizadas que agilicen el procesamiento de la información.

II. Antecedentes

Asociación Club Terraza, es un club social, que tiene como objetivo primordial procurar unir a todos sus socios, facilitándoles lugares de estancia y esparcimiento para la convivencia social y la práctica del deporte, con el fin de mantener vivo un espíritu social en apoyo de la enseñanza, desarrollo y fomento de los deportes, la cordialidad y la amistad entre sus socios.

La construcción de la sede actual del Club Terraza se inicia en mayo del año 2002. La primera etapa es inaugurada en marzo del 2003 y en octubre del 2005 se concluye la construcción del nuevo Club Terraza.

A partir del 2003 inició sus operaciones con un personal limitado que realizaba los procesos de manera manual de control de venta y facturación en el área de restaurante. Esto eran un inconveniente ya que creaban un proceso lento dando como resultado una mala calidad en el servicio, insatisfacción en los cliente y pérdidas económicas.

En el transcurso de los años con el crecimiento del club y la apertura total de las instalaciones se empezó a hacer uso de métodos manuales combinado con procesos digitales, como lo son: hojas de cálculo para llevar un mejor control de venta, ya que el personal fue creciendo y era una forma más rápida y ordenada de guardar la información en el momento.

Club terraza no ha contado anteriormente con un sistema automatizado de control de ventas y facturación se desea implementar un software simple y amigable con el usuario, puesto que se ha venido experimentando ciertos problemas al momento de facturar y han ocurrido confusiones mientras se llevan a cabo estos procesos.

Debido al completo menú caracterizado por una gran variedad de sabores nacionales e internacionales y la incorporación de nuevos socios, hoy en día existen 5 ambientes de restaurante y uno en proceso de apertura. Como consecuencia de estas expansiones en el club, se vieron en la necesidad de automatizar estos procesos de control de venta y facturación.

III. Objetivos

3.1. Objetivo General

- Desarrollar e implementar un sistema de control de ventas y facturación para el área de restaurante en Asociación Club Terraza.

3.2. Objetivos específicos

- Diagnosticar la situación actual de las gestiones que se llevan a cabo en el área de restaurante en Asociación Club Terraza.
- Diseñar la base de datos que agilice el almacenamiento de la información de manera eficiente.
- Desarrollar una interfaz gráfica que facilite el uso del sistema de control de ventas y facturación.
- Diseñar y codificar un software de control de ventas y facturación en base a los requerimientos reales en el área de restaurante en Asociación Club Terraza.

IV. Preguntas de Investigación

¿Cómo puede la implementación de un sistema de control de ventas y facturación mejorar la precisión y eficiencia en la gestión de pedidos y la emisión de facturas en el área de restaurante de la Asociación Club Terraza?

¿De qué manera un sistema automatizado de control de ventas y facturación puede contribuir a una mejor gestión de inventarios y a la generación de informes financieros detallados y precisos en el área de restaurante de la Asociación Club Terraza?

V. Justificación

La implementación de un sistema de control de ventas y facturación en el área de restaurante de la Asociación Club Terraza se justifica por varias razones clave que impactan tanto en la operatividad interna del establecimiento como en la satisfacción del cliente.

Mejora de la Eficiencia Operativa: Actualmente, la gestión manual de las ventas y la facturación en el restaurante es propensa a errores humanos, lo que genera inconsistencias en los registros y pérdidas de tiempo en la resolución de problemas. Un sistema automatizado reducirá significativamente estos errores, agilizando el proceso de toma de pedidos y emisión de facturas, lo que se traducirá en una operación más fluida y eficiente.

Optimización del Control de Inventarios: La falta de un sistema integrado dificulta el seguimiento preciso de los inventarios, lo que puede llevar a la escasez o exceso de productos. Un sistema de control de ventas y facturación permitirá una mejor gestión de inventarios, proporcionando datos en tiempo real que faciliten la toma de decisiones sobre reabastecimientos y reduciendo costos operativos asociados a la mala gestión de existencias.

Mejora en la Experiencia del Cliente: Un sistema eficiente de control de ventas y facturación contribuye directamente a mejorar la experiencia del cliente, al reducir los tiempos de espera y asegurar la precisión en la facturación. Esto, a su vez, puede aumentar la satisfacción del cliente y fomentar su lealtad, así como mejorar la reputación del restaurante.

Generación de Informes Financieros Confiables: La capacidad de generar informes financieros detallados y precisos es crucial para la administración del restaurante. Un sistema automatizado proporcionará datos exactos y en tiempo real, facilitando la elaboración de informes financieros que pueden ser utilizados para análisis de desempeño, planificación estratégica y cumplimiento de obligaciones fiscales.

Adaptación a la Competitividad del Mercado: En un entorno altamente competitivo, los restaurantes necesitan adoptar tecnologías que les permitan optimizar sus operaciones y ofrecer un mejor servicio. La implementación de un sistema de control de ventas y facturación posicionará al restaurante de la Asociación Club Terraza en un lugar de ventaja competitiva, al adoptar prácticas tecnológicas avanzadas que mejoren su eficiencia y calidad de servicio.

La creación de un sistema de control de ventas y facturación no solo aborda las deficiencias actuales en la gestión operativa del restaurante, sino que también aporta beneficios significativos en términos de eficiencia, control, satisfacción del cliente y competitividad. Estos factores justifican plenamente la inversión y el esfuerzo en la implementación de dicha tecnología.

VI. Limitaciones

El proyecto presenta varias limitaciones que deben considerarse. En primer lugar, existe una fuerte dependencia de la tecnología, ya que la implementación del sistema de control de ventas y facturación requiere que tanto el hardware como el software funcionen correctamente. Cualquier fallo técnico podría interrumpir los procesos de ventas y facturación, afectando negativamente la operación del restaurante. Además, la curva de aprendizaje del personal es una limitación significativa. Los empleados del Club Terraza necesitarán tiempo para familiarizarse y adaptarse al nuevo sistema automatizado, lo que podría disminuir temporalmente la eficiencia operativa.

Otro aspecto crítico es el costo de implementación. La inversión inicial necesaria para el desarrollo, instalación y mantenimiento del sistema puede ser considerable, impactando el presupuesto del club a corto plazo. Asimismo, el sistema requerirá mantenimiento regular y posibles actualizaciones para asegurar su funcionamiento óptimo y adaptarse a nuevos requerimientos o mejoras tecnológicas. Por último, la seguridad de los datos almacenados es esencial. Cualquier brecha de seguridad podría comprometer información sensible y afectar la confianza de los socios del club.

VII. Supuestos Básicos

Para la implementación exitosa del sistema, se basaron en varios supuestos básicos. Se asume que el Club Terraza dispone de los recursos tecnológicos necesarios, incluyendo hardware adecuado y una conexión a internet estable. También se espera que el personal esté dispuesto a colaborar y adaptarse a la nueva tecnología, participando en las capacitaciones necesarias para el uso efectivo del sistema. Otro supuesto es que la demanda de servicios en el restaurante se mantendrá o aumentará, lo que justifica la inversión en un sistema automatizado.

Además, se cree que la implementación del sistema traerá beneficios a largo plazo en términos de eficiencia operativa, reducción de errores y mejor manejo de la información. Finalmente, se supone que el Club Terraza tendrá acceso a soporte técnico confiable para resolver cualquier problema que surja con el nuevo sistema, garantizando su funcionamiento continuo y eficaz.

VIII. Categorías, temas y patrones emergentes de la investigación

La investigación ha identificado varias categorías clave. La primera es la automatización de procesos, que abarca la implementación de sistemas tecnológicos, la reducción del trabajo manual y la mejora de la eficiencia operativa. La segunda categoría es la gestión de información, que incluye el manejo de bases de datos, el acceso a información precisa y actualizada, y la generación de reportes y consultas eficientes. Otra categoría importante es la mejora en la atención al cliente, destacando la reducción de tiempos de espera, la minimización de errores en la facturación y la mejora en la experiencia del usuario. Finalmente, se destaca el desarrollo de software, con un enfoque en el uso de Visual Basic y SQL Server, el diseño y desarrollo de bases de datos, y la realización de pruebas de validación.

Dentro de estos temas, se han identificado varios patrones emergentes. Uno de ellos es la mejora continua, que subraya la necesidad constante de actualizar y mejorar el sistema para adaptarse a las nuevas demandas y tecnologías. Otro patrón es la integración de tecnología avanzada para optimizar los procesos operativos en sectores tradicionales como el de la restauración. Además, el cambio organizacional destaca el impacto que la implementación de nuevas tecnologías tiene en la estructura y funcionamiento de la organización. Por último, la orientación al cliente refleja la tendencia a desarrollar sistemas que no solo mejoren la eficiencia interna sino que también ofrezcan una mejor experiencia al cliente final.

IX. Marco teórico

A. Estado del Arte

El estado del arte en la creación de sistemas de control de ventas y facturación para restaurantes abarca una amplia gama de tecnologías y metodologías que han evolucionado considerablemente en los últimos años. A continuación, se presentan los principales desarrollos y tendencias en este ámbito:

Sistemas de Punto de Venta (POS):

Los sistemas de punto de venta han sido la piedra angular de la gestión de ventas y facturación en los restaurantes. Estos sistemas han avanzado desde simples cajas registradoras a sofisticadas plataformas que integran múltiples funciones, incluyendo la gestión de inventarios, la emisión de facturas electrónicas, la gestión de relaciones con clientes (CRM) y el análisis de datos en tiempo real. Ejemplos destacados de estos sistemas son Toast, Square, y Lightspeed, los cuales ofrecen soluciones específicas para el sector de la restauración.

Facturación Electrónica:

La implementación de la facturación electrónica ha sido impulsada por regulaciones fiscales en muchos países, lo que ha llevado a los restaurantes a adoptar soluciones que faciliten la emisión de facturas digitales. Esta tecnología no solo asegura el cumplimiento normativo, sino que también mejora la eficiencia y reduce los errores asociados con la facturación manual. Sistemas como Invoice2go y Zoho Invoice se han destacado en este ámbito, proporcionando herramientas robustas para la gestión de facturación electrónica.

Integración de Gestión de Inventarios:

La integración de la gestión de inventarios con sistemas de ventas y facturación ha permitido a los restaurantes mantener un control preciso de sus existencias, optimizando el proceso de reabastecimiento y reduciendo el desperdicio. Soluciones como MarketMan y Upserve ofrecen capacidades avanzadas para el seguimiento de inventarios, proporcionando alertas en tiempo real sobre niveles de stock y análisis de consumo de productos.

Análisis de Datos y Reportes:

La capacidad de generar reportes detallados y realizar análisis de datos es crucial para la toma de decisiones informadas en los restaurantes. Los sistemas modernos de control de ventas y facturación incluyen módulos de análisis que permiten a los gerentes evaluar el desempeño del negocio, identificar tendencias y oportunidades de mejora, y tomar decisiones estratégicas basadas en datos concretos. Herramientas como Tableau y Power BI se utilizan ampliamente para visualizar y analizar datos en el contexto de la restauración.

Experiencia del Usuario y Personalización:

La mejora de la experiencia del usuario, tanto para el personal del restaurante como para los clientes, es una tendencia clave. Sistemas intuitivos y fáciles de usar, como TouchBistro y Clover, han sido diseñados con interfaces amigables que facilitan el entrenamiento del personal y mejoran la eficiencia operativa. Además, la personalización de las interacciones con los clientes, a través de programas de fidelización y recomendaciones personalizadas, se ha vuelto un componente importante en la gestión de relaciones con clientes.

Tecnologías Móviles y en la Nube:

La adopción de tecnologías móviles y basadas en la nube ha transformado la gestión de ventas y facturación en los restaurantes. Estos sistemas permiten a los restaurantes operar de manera más flexible y accesible, con la capacidad de gestionar operaciones desde cualquier lugar y en cualquier momento. Soluciones como Revel Systems y Lavu han liderado esta tendencia, ofreciendo plataformas móviles y basadas en la nube que soportan todas las funciones críticas del restaurante.

El estado del arte en los sistemas de control de ventas y facturación para restaurantes muestra una evolución significativa hacia la integración, la eficiencia y la personalización. La adopción de estas tecnologías no solo mejora la operatividad interna de los restaurantes, sino que también eleva la experiencia del cliente y facilita una gestión más efectiva del negocio. Para la Asociación Club Terraza, la implementación de un sistema moderno y robusto en este contexto puede ofrecer ventajas competitivas sustanciales y una mejora integral en sus operaciones diarias.

El marco teórico que fundamenta esta investigación, presenta los principales conceptos asociados a los sistemas de información, su clasificación y características más generales, posteriormente se define el lenguaje de programación empleado, base de datos, conceptos y terminología.

B. Perspectiva teórica asumida

9.1. Sistemas

Según Van Gigch, un sistema se define como " una unión de partes o componentes, conectados en una forma organizada. Las partes se afectan por estar en el sistema y se cambian si lo dejan. La unión de partes hace algo (muestra una conducta dinámica como opuesto a permanecer inerte).

Es un conjunto organizado de cosas o partes interactuantes e interdependientes, que se relacionan formando un todo unitario y complejo. Cabe aclarar que las cosas o partes que componen al sistema, no se refieren al campo físico (objetos), sino más bien al funcional. De este modo las cosas o partes pasan a ser funciones básicas realizadas por el sistema. Podemos enumerarlas en: entradas, procesos y salidas.

9.2. Clasificación de sistemas

De acuerdo a su constitución, los sistemas se clasifican de la siguiente manera: Sistemas físicos o concretos: estos sistemas están compuestos por elementos tangibles, por ejemplo, maquinas, equipos u objetos. En informática, nos referimos a estos sistemas como el hardware.

Sistemas abstractos: son sistemas conformados por elementos cognitivos, por ejemplos los planes, las hipótesis y las ideas. Estos sistemas son conocidos como software en el campo de la informática.

Respecto a su relación con el medio ambiente los clasificamos como abiertos o cerrados:

Sistemas abiertos: Los sistemas abiertos propuestos por Von Bertalanfy, son aquellos que mantienen una relación con su entorno interno e influyen sobre estos y también reciben influencias del entorno externo. Este tipo de sistemas se intercambia materia, energía o información con el ambiente.

Sistemas cerrados: Son sistemas que carecen de relación con el entorno externo y son estudiados por la física clásica, los mismos que se estudian aislados de su entorno. Es decir que son sistemas en los cuales el intercambio de materia, energía o información con ambiente es considerado como nulo.

9.3. Sistemas de Información

Peralta (2008) define sistema de información como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Peña (2006) Nos dice que un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones.

9.4. Actividades de los sistemas de información

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. (Peralta, 2008)

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Suele ser almacenada en estructuras de información denominadas archivos.

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros.

9.5. Software

Según el instituto de ingenieros eléctricos y electrónicos, (IEEE, según sus siglas en inglés):

Software es la suma total de los programas de ordenador, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo y un producto de software es un producto diseñado para un usuario. En este contexto, la Ingeniería de Software (SE según sus siglas en inglés) es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software.

Según Pressman (2010) El software de computadora es el producto que construyen los programadores profesionales y al que después le dan mantenimiento durante un largo tiempo. Incluye programas que se ejecutan en una computadora de cualquier tamaño y arquitectura, contenido que se presenta a medida que se ejecutan los programas de cómputo e información descriptiva tanto en una copia dura como en formatos virtuales que engloban virtualmente a cualesquiera medios electrónicos.

9.6. Tipos de Software

9.6.1. Software de aplicación

Contiene todos y cada uno de esos programas y utilidades que derivan de una programación de software y, que cumplen una tarea específica, en casi cualquier área de la vida diaria, que se usan a través de dispositivos móviles y computadores.

Dentro de esta categoría se incluyen multitud de tipos de herramientas y contenido, dado que la definición es tan flexible que abarca desde algo tan sencillo como un editor de texto de los que usamos en el ordenador para tomar notas, hasta un videojuego, los programas empresariales, las herramientas de diseño de imagen o artísticas, las bases de datos y las herramientas de cálculo, etc.

9.6.2. Software de programación

Son aquellas herramientas que utilizan los profesionales como base para la elaboración de programas. Estos lenguajes de programación son imprescindibles en el día a día, debido a que se trata de las herramientas de trabajo del programador.

A través de conocimiento lógico y de lenguaje de programación orientada a objetos, es posible diseñar utilidades digitales para que ejerzan diversas funciones.

9.6.3. Software de sistema

Dentro de los tipos de software, el de sistema es el más importante. Es quién le permite al usuario usar la interfaz del sistema operativo que viene incorporado al dispositivo. Se desarrolla el software de sistema como herramienta que sirve a modo de motor del equipo en el cual se instala.

Son los sistemas operativos de los dispositivos y tienen dos funciones principales. Una de ellas es servir como interfaz de control al usuario, convirtiendo en una herramienta funcional los datos de programación. La otra consiste en crear el sistema por el cual el hardware podrá utilizarse tanto de forma independiente como a nivel de componentes y de accesorios conectados (desde la memoria interna hasta el disco duro o un teclado).

9.6.4. Ingeniería de software

Bauer (2014) La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales.

La ingeniería de software es: La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software;

es decir, la aplicación de la ingeniería al software. La ingeniería de software es una tecnología con varias capas.

9.6.5. Requerimientos del software

Leite (2015) Los requerimientos del software es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos.

Pressman (2010) destaca que muchos enfoques han sido propuestos para recabar los requerimientos en forma colaborativa. Por el cual cada uno utiliza un escenario un poco diferente, siendo variantes de los siguientes lineamientos básicos:

- Tanto ingenieros como otros participantes dirigen e intervienen en las reuniones.
- Se establecen reglas para la preparación y participación.
- Se sugiere una agenda con suficiente formalidad para cubrir todos los puntos importantes, pero con la suficiente informalidad para que se estimule el libre flujo de ideas.
- Un “facilitador” (cliente, desarrollador o participante externo) controla la reunión.

9.6.6. Tipos de requerimientos

Somerville (2005) Los requerimientos funcionales: Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares.

Se entiende que son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Cuando hablamos de las entradas, no necesariamente hablamos sólo de las entradas de los usuarios. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos. En algunos casos, los requisitos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer.

Requerimientos no funcionales: Representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Son restricciones de los servicios o funciones ofrecidos por el sistema.

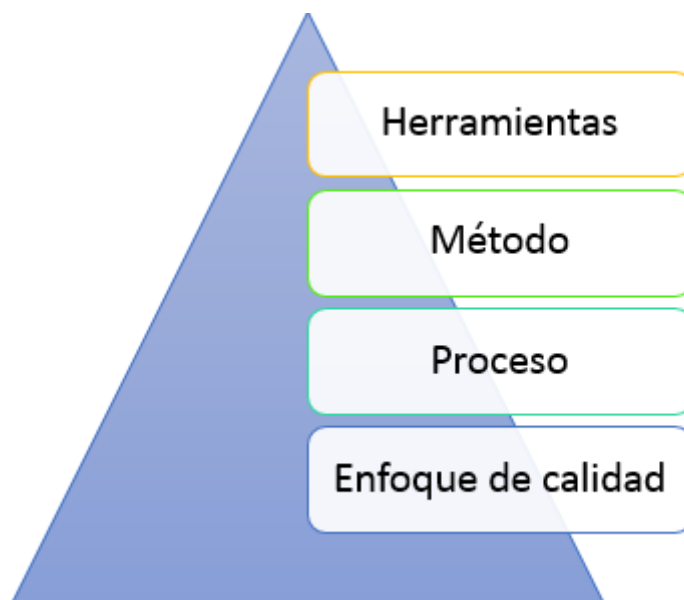
Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir.

9.6.7. Capas de la ingeniería de software

Pressman (2010). La ingeniería de software es una tecnología de varias capas las mismas que deben asegurar la calidad del producto, El fundamento para la ingeniería de software es la capa proceso. El proceso define una estructura que debe establecerse para la obtención eficaz de tecnología de ingeniería de software.

El proceso de software forma la base para el control de la administración de proyectos de software, y establece el contexto en el que se aplican métodos técnicos, se generan productos del trabajo.

Ilustración 1. Capas de la ingeniería de software



9.6.8. Metodologías para el desarrollo de software

Universitat Carlemany (2020) La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático, lo cual es imprescindible trabajar con una metodología por una cuestión de organización que sirven para controlar el desarrollo del proyecto.

El trabajo con una metodología de desarrollo de software permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar.

9.6.9. Modelos de ciclo de vida de desarrollo de software

Intelequia (2020) La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con grandes posibilidades de éxito.

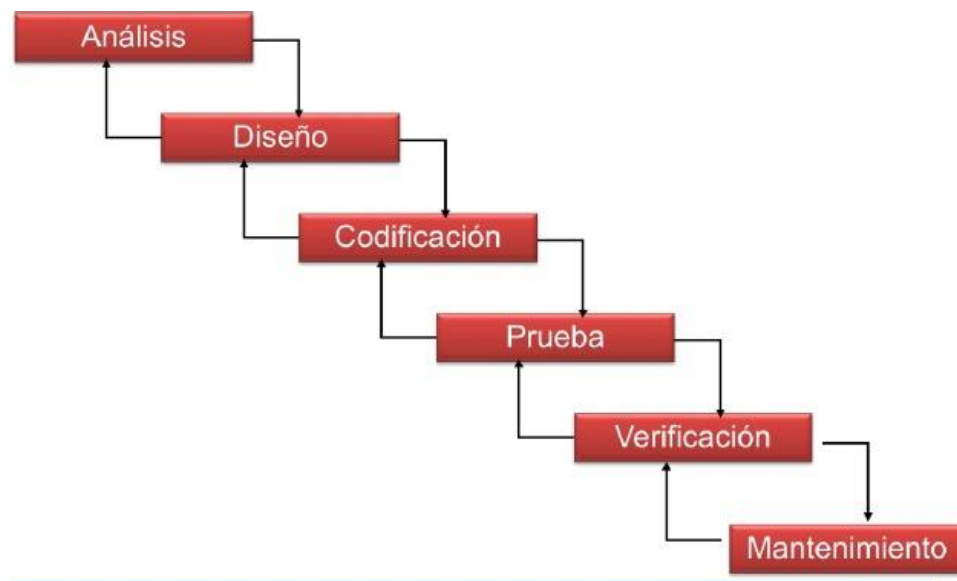
Una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado.

9.6.10. Modelo en Cascada

Permite organizar el trabajo en vertical, de arriba a abajo. Esto significa que se realiza una actividad por fases secuenciales y que no es posible pasar a la siguiente hasta que no se haya verificado la anterior.

La principal característica del modelo de cascada es que sigue una secuencia lineal, esto permite identificar unas etapas específicas a las cuales se les debe dar cumplimiento en orden, a medida tal que se van implementando las adecuaciones pertinentes y las pruebas para mitigar los errores que se puedan presentar.

Ilustración 2. Modelo de cascada.

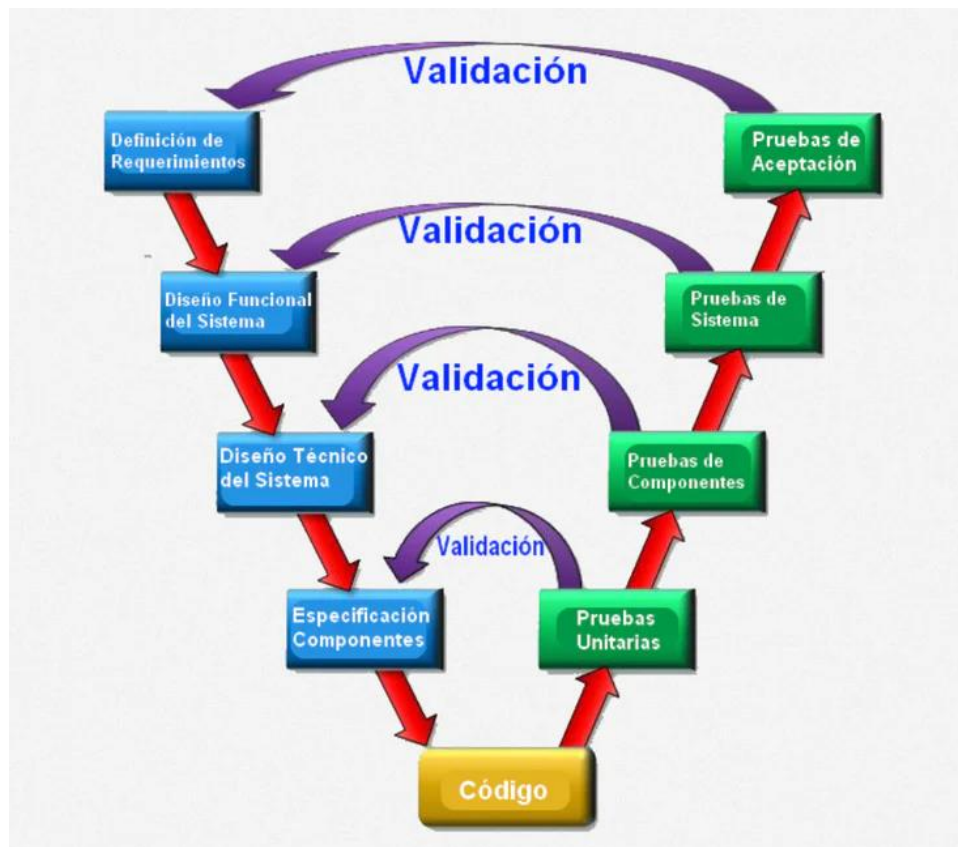


9.6.11. Modelo en V

El modelo V es una representación gráfica de un ciclo de vida de desarrollo de sistemas. Se utiliza para producir modelos rigurosos de ciclo de vida de desarrollo y modelos de gestión de proyectos.

El modelo V resume los pasos principales que se deben tomar junto con los entregables correspondientes dentro del marco de validación del sistema computarizado o el desarrollo del ciclo de vida del proyecto. Describe las actividades a realizar y los resultados que deben producirse durante el desarrollo del producto.

Ilustración 3. Modelo en V

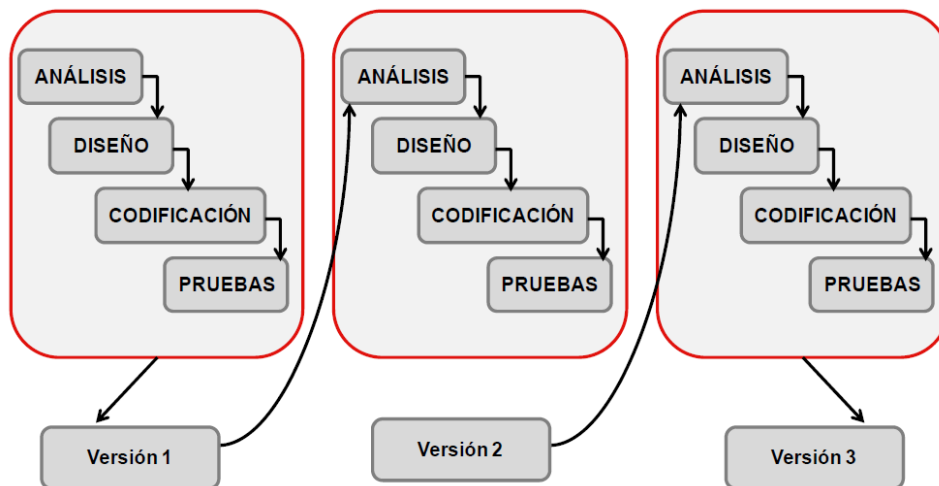


9.6.12. Modelo de desarrollo incremental

El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.

Cuando se utiliza un modelo incremental, el primer incremento es a menudo un producto esencial, sólo con los requisitos básicos. Este modelo se centra en la entrega de un producto operativo con cada incremento. Los primeros incrementos son versiones incompletas del producto final, pero proporcionan al usuario la funcionalidad que precisa y también una plataforma para la evaluación.

Ilustración 4. Modelo de desarrollo incremental

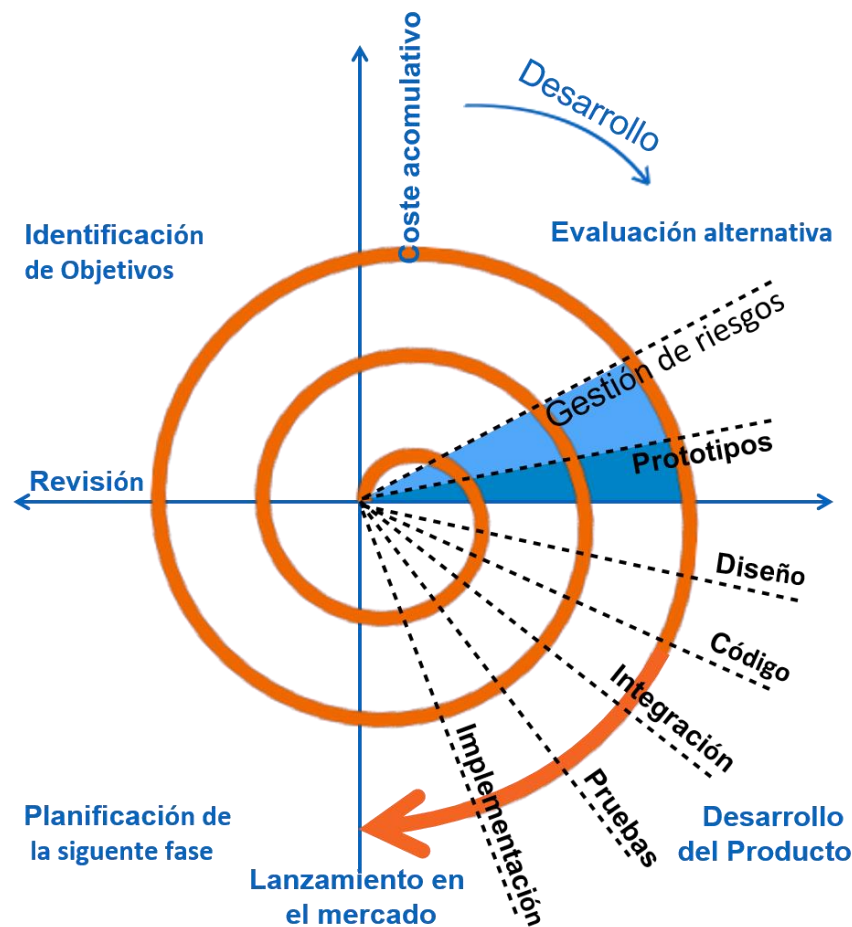


9.6.13. Modelo en Espiral

Boehm (1986) Parte de la base de que el desarrollo de aplicaciones se debe llevar a cabo en un ciclo iterativo que se debe repetir tantas veces como sea necesario hasta alcanzar el objetivo. Gracias a las valoraciones regulares de los riesgos y a los controles rutinarios del producto intermedio, el modelo en espiral minimiza considerablemente el riesgo de fracaso en los proyectos de software.

Universitat carlemany (2020) Cuenta con cuatro fases distintas y el objetivo es acercarse a lo que desea el cliente. Cuanto más nos acercamos al centro, más cerca estaremos de ello. El resultado es de buena calidad, porque es el propio cliente el que cierra el círculo dando su validación o indicando si son necesarias otras modificaciones. (Universitat carlemany 2020).

Ilustración 5. Modelo en espiral



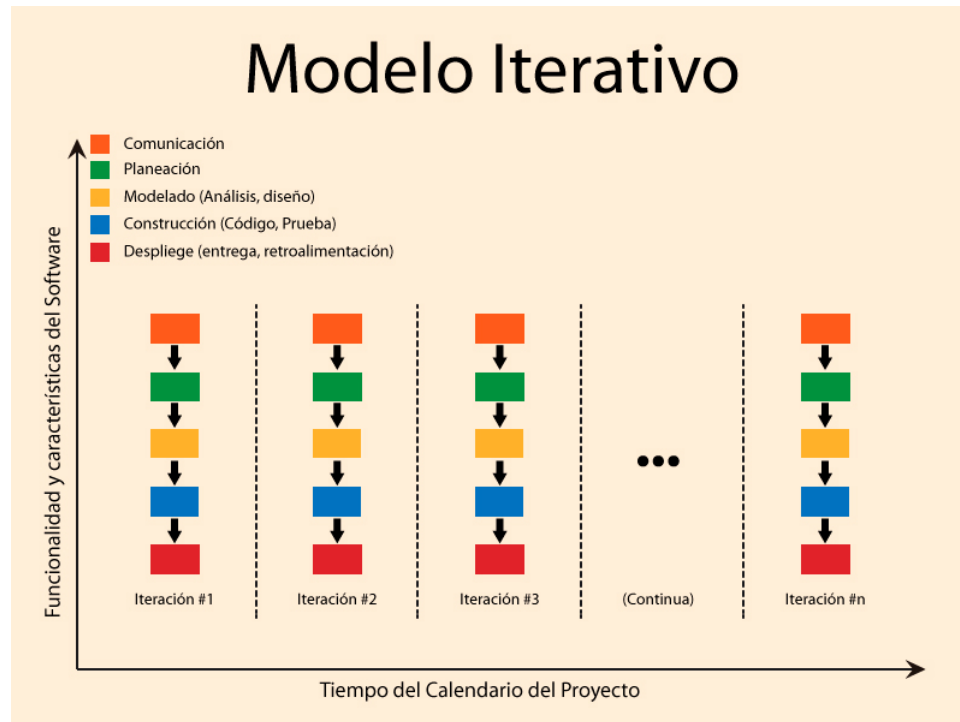
9.6.14. Modelo Iterativo

Pressman (2005) Es un modelo derivado del ciclo de vida en cascada. Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos.

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras. Estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.

Una de las principales ventajas que ofrece este modelo es que no hace falta que los requisitos estén totalmente definidos al inicio del desarrollo, sino que se pueden ir refinando en cada una de las iteraciones.

Ilustración 6. Modelo Iterativo



9.6.15. Modelo de Prototipo

El modelo de prototipos, también conocido como modelo de desarrollo evolutivo, es usado principalmente en proyectos de desarrollo de software. Este modelo se utiliza para dar al usuario una vista preliminar de lo que será el sistema.

Dentro de los recursos que se buscan reducir al utilizar este modelo son el tiempo, pues el prototipo debe ser construido a la brevedad posible, y dinero ya que en el prototipo no se deben invertir muchos recursos.

Ilustración 7. Modelo de prototipo



9.7. Desarrollo de sistema

Un proceso de desarrollo de sistemas tiene como propósito la producción eficaz y eficiente de un producto de software que satisfaga los requerimientos del usuario. Este proceso es intensamente intelectual, basado por la creatividad y juicio de las personas involucradas.

El desarrollo de sistemas incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento, la elaboración de manuales y la implementación del sistema.

9.8. Facturación

La facturación de una empresa es un documento que presenta un detalle de los servicios prestados o los productos vendidos y que se da al comprador o cliente para pedir su pago.

Como señala Sage (2021) el proceso de facturación se relaciona con la elaboración, registro, envío y cobro de las facturas, en el cual una factura es un documento que se reflejan los datos de una transacción mercantil y contiene información que se detallan de la siguiente manera:

- Los datos relativos al emisor y receptor de la factura (nombre, NIF, dirección).
- La información sobre la transacción realizada (es el concepto de la factura).

- El importe facturado y el desglose por cada concepto.
- Los impuestos aplicables.
- El número de factura, la fecha y el lugar.
- El modo y plazo de pago.

Sage (2022) La facturación es el día a día de cualquier negocio y puede ser la clave para conocer:

- El nivel de liquidez que tiene el negocio.
- De dónde provienen los ingresos.
- Qué productos o servicios se venden más.
- Qué posibilidades de inversión existen.
- Qué decisiones estratégicas se pueden tomar.

9.9. Control de ventas

Clavijo (2022) El control de ventas es un registro que asegura la productividad de la fuerza de ventas de una empresa, asociado a los productos y los vendedores. Este control permite que un negocio se mantenga alerta y coherente en sus acciones para mantenerse, así como vender más.

Es importante el uso de control de ventas por que da la oportunidad para que una empresa aborde de forma oportuna los problemas y quejas de los clientes. Así, se genera una imagen más positiva de la organización ante los ojos de los clientes y del mercado en general.

Salesforce (2022) Uno de los principales beneficios del control de ventas es el acompañamiento de los negocios genera informes que permiten que el gestor sepa exactamente cómo está el escenario de la empresa y que los equipos de ventas utilicen los números para mejorar su productividad y pensar en mejores acciones de ventas.

9.10. Sistema de control de ventas y facturación

Los sistemas de controles de ventas comprenden una serie de actividades, entre las que se encuentran las políticas, la organización y los procedimientos que utiliza la empresa con el objetivo de asegurar la mayor cantidad de ingresos posibles. Por tal motivo, la implantación de un sistema de control de ventas y

facturación es vital para la consecución de los objetivos económicos de cualquier empresa.

Dentro de las metas administrativas que persigue la implantación de este tipo de sistemas se encuentran el incremento de las ventas, asegurando que se cumplen con cada una de las políticas previamente establecidas.

Con estos tipos de sistemas se consigue que la facturación realizada tenga los precios correctos, evitando así cualquier tipo de error, tanto en el cobro como en los artículos que se despachan a los consumidores.

Entre otras de las funcionalidades que poseen este tipo de programas se encuentran la actualización de los precios de los artículos, la gestión de productos compuestos.

9.11. Metodologías ágiles

Hurtado (2021) define las metodologías ágiles como:

Son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

9.11.1. Tipos de metodología ágiles

Metodología ágil XP

Esta herramienta es muy útil sobre todo para empresas que están en proceso de consolidación, puesto que su principal objetivo es ayudar en las relaciones entre los empleados y clientes. La clave del éxito es potenciar las relaciones personales, a través, del trabajo en equipo, fomentando la comunicación y eliminando los tiempos muertos.

La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance.

Metodología SCRUM

Se basa en una estructura de desarrollo incremental, esto es, cualquier ciclo de desarrollo del producto y/o servicio se desgrana en «pequeños proyectos» divididos en distintas etapas: análisis, desarrollo y testing. Esta metodología

permite abordar proyectos complejos que exigen una flexibilidad y una rapidez esencial a la hora de ejecutar los resultados.

Metodología Kanban

Consiste en la elaboración de un cuadro o diagrama en el que se reflejan tres columnas de tareas; pendientes, en proceso o terminadas. Este cuadro debe estar al alcance de todos los miembros del equipo, evitando así la repetición de tareas o la posibilidad de que se olvide alguna de ellas. (Hurtado, 2021)

Metodología Lean

Es considerada tanto una metodología de trabajo como una filosofía centrada en maximizar el valor del cliente y minimizar el desperdicio. Esto traducido a los procesos de fabricación “pull” radica en producir solo lo necesario y en el momento adecuado. Es decir, consiste en extender la metodología al lanzamiento de nuevas empresas al mercado.

Metodología Design Sprint

Una metodología de Google que está favoreciendo a los perfiles profesionales del mundo agile. Se trata de un proceso que dura 5 días en el que el negocio tiene que resolver todas las cuestiones relacionadas con diseño, prototipado, testeo de clientes. La idea es que el trabajo se elabora en etapas de sprints en las que meses de trabajo se pueden reducir en pocas semanas, en vez de esperar a lanzar un producto para entender si la idea es buena, el prototipo proporciona antes la información para evitar posibles errores.

9.11.2. Ventajas de las metodologías ágiles

Sotomayor (2021) ventajas que nos brinda la gestión ágil de proyectos:

Mejora de la calidad del producto: Fomentan el enfoque proactivo de los miembros del equipo en la búsqueda de la excelencia del producto.

Mayor satisfacción del cliente: El cliente está más satisfecho al verse involucrado y comprometido a lo largo de todo el proceso de desarrollo. Mediante varias demostraciones y entregas, el cliente vive a tiempo real las mejoras introducidas en el proceso.

Trabajo colaborativo: La división del trabajo por distintos equipos y roles junto al desarrollo de reuniones frecuentes, permite una mejor organización del trabajo.

Uso de métricas más relevantes: Las métricas utilizadas para estimar parámetros como tiempo, coste, rendimiento, etc. son normalmente más reales en proyectos ágiles que en los tradicionales.

Mayor control y capacidad de predicción: La oportunidad de revisar y adaptar el producto a lo largo del proceso ágil, permite a todos los miembros del proyecto ejercer un mayor control sobre su trabajo.

Reducción de costes: La gestión ágil del proyecto elimina prácticamente la posibilidad de fracaso absoluto en el proyecto, porque los errores se van identificando a lo largo del desarrollo en lugar de esperar a que el producto esté acabado y toda la inversión realizada.

9.8. Lenguaje de programación

Rock Content (2019) El lenguaje de programación es un lenguaje formal que, mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para crear programas que controlen el comportamiento físico y lógico de una máquina.

Raffino (2018) Es un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Dicho lenguaje está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado.

Los lenguajes se clasifican de la siguiente manera:

Lenguajes de bajo nivel

Se trata de lenguajes de programación que están diseñados para un hardware específico y que por lo tanto no pueden migrar o exportarse a otros computadores.

Sacan el mayor provecho posible al sistema para el que fueron diseñados, pero no aplican para ningún otro. Están totalmente orientados a la máquina entre ellos: lenguaje máquina y lenguaje ensamblador.

Lenguaje de alto nivel.

Son lenguajes de programación que aspiran a ser un lenguaje más universal, por lo que pueden emplearse indistintamente de la arquitectura del hardware, es decir, en diversos tipos de sistemas.

Lenguajes de nivel medio.

Este término no siempre es aceptado, que propone lenguajes de programación que se ubican en un punto medio entre los dos anteriores: pues permite operaciones de alto nivel y a la vez la gestión local de la arquitectura del sistema.

Otra forma de clasificación.

Lenguajes imperativos.

Menos flexibles, dada la secuencialidad en que construyen sus instrucciones, estos lenguajes programan mediante órdenes condicionales y un bloque de comandos al que retornan una vez llevada a cabo la función.

Lenguajes funcionales.

También llamados procedimentales, estos lenguajes programan mediante funciones que son invocadas conforme a la entrada recibida, que a su vez son resultado de otras funciones. Raffino (2018).

Ejemplos de lenguajes de programación

Visual Basic

Bembibre (2009) define Visual Basic como un lenguaje de programación guiado por eventos que permite mayor operatividad y mejores resultados. Con el objetivo de entregar a los usuarios un paquete de utilidades simples y accesibles.

La creación de interfaces gráficas para diferentes utilidades es una de las principales funciones del Visual Basic y es por esto que es altamente usado en espacios profesionales donde se requieren soportes gráficos para mayor organización de los contenidos y materiales

Entre sus principales características están:

- Separa la creación de la interfaz gráfica y el código
- Una barra de herramientas
- Un explorador de proyectos
- Toda programación está orientada a eventos

Cobol

Su nombre es un acrónimo para Common Business-Oriented Language (Lenguaje común orientado a los negocios) y se trata de un lenguaje de programación universal creado en 1959, orientado principalmente a la informática de gestión, es decir, empresarial.

El lenguaje COBOL se desarrolló con el objetivo principal de conseguir un lenguaje de programación que se pudiese utilizar en cualquier tipo de ordenador, y que además fuese más sencillo que sus predecesores a la hora de ejecutarlo.

Fortran

Es un lenguaje de programación de alto nivel de propósito general, procedimental e imperativo, que está especialmente adaptado al cálculo numérico y a la computación científica.

Es utilizado, por una parte, debido a la existencia de numerosas bibliotecas de funciones utilizables en FORTRAN, por otra parte, porque existe compiladores FORTRAN potentes que producen ejecutables muy rápidos. No obstante, se reemplaza cada vez más, incluso para aplicaciones científicas, por los lenguajes C y C ++.

Java

Java es un lenguaje sencillo y orientado a objetos, que permite el desarrollo de aplicaciones en diversas áreas, como seguridad, animación, acceso a bases de datos, aplicaciones cliente-servidor, interfaces gráficas, páginas Web interactivas y desarrollo de aplicaciones móviles, entre otras.

Una de sus principales características es la creación de módulos reutilizables, que funcionan sin la necesidad de conocer su estructura interna. Esto permite al

usuario añadir nuevos módulos, además de obtener programas independientes de la plataforma en la cual fueron desarrollados, gracias a la implementación de la llamada Máquina Virtual de Java (JVM).

9.9. Programación orientada a objetos

La programación Orientada a objetos se define como un paradigma de la programación una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

Este consiste en la utilización de objetos que representen una entidad en el mundo real al crear dichos objetos es posible añadirles tanto propiedades como comportamientos. La programación orientada a objetos disminuye los errores y promueve la reutilización del código.

9.9.1. Pilares fundamentales de la programación orientadas a objetos

Baltazar, Bailón (2021): Pilares fundamentales de la programación orientada a objetos:

Abstracción: Es un procedimiento que permite la elección de una determinada entidad de la realidad, sus características y funciones que desempeñan, la cual es representada mediante clases que contienen atributos y métodos de dicha clase.

Encapsulamiento: Se acostumbra a proteger la información o el estado de los atributos para que no se pueda ver o modificar la información del objeto sin el mecanismo adecuado. Para ello, se utilizan métodos para recuperar la información y asegurar que la información proporcionada sea consistente con el objeto; y a su vez, poder asignar un nuevo valor y verificar que no afecte la integridad del objeto.

Herencia: Nos permite reutilizar el código programado en cada clase heredando o extendiendo las características de un objeto a sus descendientes o derivados.

Polimorfismo: En una colección de objetos con herencia, si las clases especializadas de una superclase tienen un método con la misma definición o firma, reaccionarán de la manera adecuada cuando reciban el mismo mensaje. (Baltazar, Bailón 2021)

9.9.2. Ventajas de la programación orientadas a objetos

Certus (2022) Entre las ventajas de la programación orientada a objetos están:

- Por medio de la herencia, podemos eliminar código redundante y extender el uso de clases existentes.
- Podemos construir programas a partir de módulos de trabajo estándar que se comunican entre sí, en lugar de tener que empezar desde cero la escritura del código. Esto conduce a un ahorro en el tiempo de desarrollo y a una productividad mayor.
- El principio de ocultación de los datos ayuda al programador a construir programas seguros que no pueden ser invadidos por código de otras partes del programa.
- Es posible tener múltiples copias de un objeto que coexisten sin ninguna interferencia.
- Es posible transformar objetos del dominio del problema a otros en los programas.
- Es fácil dividir el trabajo de un proyecto basado en objetos.
- El enfoque del diseño centrado en los datos nos permite capturar más detalles de un modelo de forma que se puedan implementar.
- Se pueden mejorar y actualizar fácilmente los sistemas orientados a objetos desde los pequeños hasta los grandes sistemas.
- Las técnicas de pase de mensajes para la comunicación entre objetos simplifican mucho las descripciones de las interfaces con sistemas externos.
- Se puede gestionar fácilmente la complejidad del software.

Elementos de la programación orientada a objetos

Vásquez (2020) La POO se compone de los siguientes 4 elementos:

Clases: Las clases pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos construir N cantidad de objetos.

Propiedades: Las propiedades son las características de una clase, tomando como ejemplo la clase humanos, las propiedades podrían ser: nombre, el género, la altura, color de cabello, color de piel, etc.

Métodos: Los métodos son las acciones que una clase puede realizar, siguiendo el mismo ejemplo anterior, estas podrían ser: caminar, comer, dormir, soñar, respirar, nadar, etc.

Objetos: Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales. Técnicamente, los objetos son instancias de una clase, vendría siendo cuando ya le colocamos un “nombre” a nuestras clases (moldes). Por ejemplo: El objeto “Arnell”, quien es una instancia de la clase humanos.

9.10. Visual Basic

Revista Informática (2015): El lenguaje de programación Visual Basic es uno de los lenguajes de programación que utiliza una interfaz visual es decir que nos permite programar en un entorno gráfico, nos permite realizar un gran número de tareas sin escribir código, simplemente realizando operaciones con el ratón sobre la pantalla de la computadora.

Las sintaxis que utiliza este lenguaje de programación provienen del conocido BASIC, pero completada con comandos y códigos de otros lenguajes más modernos. Este lenguaje de programación Visual Basic tiene un apartado dedicado a la Programación Orientada a Objetos.

Es un lenguaje muy apropiado para el manejo de bases de datos. Muchas empresas lo utilizan para la gestión de sus bases de datos porque su utilización es sencilla y abundan los programadores de este lenguaje.

De este lenguaje de programación han surgidos algunos derivados como: El VBScript es un lenguaje predeterminado para el Active Server Pages (ASP) que es un lenguaje de programación web. O el Visual Basic.NET que es un lenguaje de similares características a las del C#. (Revista Informática, 2015).

Ventajas de desarrollar en este entorno.

Es uno de los lenguajes más utilizados, así que es muy sencillo encontrar información, documentación y fuentes para los proyectos.

Tiene acceso a casi toda la API de Windows y puede usar la plataforma de sistemas Windows.

Añade soporte para ejecutar scripts en las aplicaciones con VBScript o JScript, gracias a Microsoft Script Control.7.

Al ser de Windows, integra su diseño e implementación de formularios.

Características de Visual Basic

Separación de la creación de la interfaz gráfica y el código.

Una barra de herramientas, con los controles (se podían añadir muchos o crear de propios) necesarios para la creación de formularios. Sólo había que arrastrar y redimensionarlos dentro de la interfaz. Los controles comunes eran labels, textbox, button, checkbox, picturebox, combobox, frame, timer, etc.

Un explorador de proyectos, para ver todos los formularios, controles, código, etc. de nuestro proyecto y un panel de propiedades, dependiendo del formulario o control seleccionado. De esta manera se podía modificar sus características en modo diseño.

En la parte inferior una Ventana inmediato, que mientras se depuraba podías escribir directamente funciones simples o el valor contenido de las variables, y nos ayuda a la hora de corregir errores en nuestro código.

Pero la característica principal de visual Basic es que toda la programación está orientada a eventos.

9.11. Base de Datos

Oracle (2022) define una base de datos como: Una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático que está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos.

Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

Modelos de base de datos.

Lucidchart(s.f) Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten. La mayoría de los modelos de datos se pueden representar por medio de un diagrama de base de datos acompañante.

9.12. Tipos de modelos de base de datos

Modelo de base de datos jerárquico

Se caracteriza por presentar los datos en una estructura de árbol invertido, donde cada registro tiene un único nodo raíz, del que surgen otros nodos (registros); los nodos en un mismo nivel son nodos padres, cada nodo padre tiene el mismo nodo raíz, y puede tener nodos hijos, pero los nodos hijos solo pueden tener un nodo padre. Este modelo se emplea poco actualmente.

Modelo relacional

Este modelo es el que emplean las bases de datos relacionales y ordena los datos en tablas (relaciones) compuestas por columnas y filas. Cada columna alberga un atributo de la entidad (nombre, dirección, fecha de nacimiento...); a los atributos de una relación se los llama dominio.

Escogiendo un atributo en concreto o una combinación de varios tenemos una clave primaria, a la que se puede hacer referencia en otras tablas, en las que será una clave externa.

Modelo de red

El modelo en red de base de datos parte del modelo jerárquico, pero aquí se permiten las relaciones de uno a muchos o de muchos a muchos entre registros vinculados, teniendo registros principales múltiples.

El modelo se crea a través de conjuntos de registros relacionados; cada uno de estos conjuntos consiste en un registro propietario o principal y uno o más registros miembros o secundarios.

Modelo de base de datos orientado a objetos

El modelo de la base de datos orientada a objetos define la base de datos como una colección de objetos utilizados en la programación orientada a objetos (es decir, que emplear lenguajes como C++ o Java, por ejemplo). Este modelo de base de datos utiliza tablas también, pero no solo se limita a ellas y permite almacenar información muy detallada sobre cada objeto.

Modelo entidad-relación

El modelo entidad-relación es básicamente el paso previo a uno modelo de bases datos relacional, puesto que se trata de un diagrama elaborado a través de unos elementos básicos y su relación entre ellos:

- Entidades (son los objetos que se representan en la base de datos).
- Atributos (son el contenido de la entidad, sus características). A los atributos se les asigna un clave para distinguirlos de los demás registros.
- Relación (el vínculo que define la dependencia entre varias entidades).
- Cardinalidad (es la participación entre entidades, que pueden ser uno a uno, uno a varios o varios a varios).

Gestores de base de datos

Un sistema gestor de base de datos (SGBD) o Database Management System (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos.

Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones.

En la siguiente clasificación Marín (2019) define los distintos gestores de bases de datos que son muy conocidos:

My SQL

Es el sistema gestor de bases de datos relacional por excelencia. Es un SGBD multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además, es el más usado en aplicaciones creadas como software libre.

Las principales ventajas de este Sistema Gestor de Bases de datos son:

- Facilidad de uso y gran rendimiento
- Facilidad para instalar y configurar
- Soporte multiplataforma
- Soporte SSL

María BD

Este SGBD es una derivación de MySQL que cuenta con la mayoría de características de este e incluye varias extensiones. Nace a partir de la adquisición de MySQL por parte de Oracle para seguir la filosofía Open Source y tiene la ventaja de que es totalmente compatible con MySQL.

Entre las principales características de este Sistema Gestor de Bases de datos se encuentran:

- Aumento de motores de almacenamiento
- Gran escalabilidad
- Seguridad y rapidez en transacciones
- Extensiones y nuevas características relacionadas con su aplicación para Bases de datos NoSQL.

SQLite

Es una biblioteca utilizada en multitud de aplicaciones actuales ya que es open source y las consultas son muy eficientes.

Las principales características de SQLite son:

- El tamaño, al tratarse de una biblioteca, es mucho menor que cualquier SGBD
- Reúne los cuatro criterios ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) logrando gran estabilidad
- Gran portabilidad y rendimiento

Microsoft SQL Server

Es un sistema gestor de bases de datos relacionales basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Es un sistema propietario de Microsoft. Sus principales características son:

- Soporte exclusivo por parte de Microsoft.
- Escalabilidad, estabilidad y seguridad.
- Posibilidad de cancelar consultas.
- Potente entorno gráfico de administración que permite utilizar comandos DDL y DML.
- Aunque es nativo para Windows puede utilizarse desde hace ya un tiempo en otras plataformas como Linux o Docker.

Oracle

Oracle es un database management system utilizado principalmente por las grandes empresas para que puedan controlar y gestionar una gran cantidad de contenidos desde un solo archivo.

Así se consigue minimizar el riesgo de pérdida de información, además de reducir costes. Entre sus acciones se pueden destacar las siguientes: soporte de transacciones, estabilidad y soporte multiplataforma de Windows, Linux, Mac OS X, Unix y BDS.

9.13. Sistemas de gestión de base de datos

PowerData (2015) Un sistema manejador de bases de datos (SGBD, por sus siglas en inglés) o Data Base Management System (DBMS) es una colección de software muy específico, orientado al manejo de base de datos, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.

9.13.1. Sistemas de gestión de bases de datos relacionales

Naeem (2020) El sistema de gestión de bases de datos o el sistema de gestión de bases de datos relacionales más disponible comercialmente y para toda la empresa que se utiliza en la actualidad es el lenguaje de consulta estructurado

(base de datos SQL) para acceder a la base de datos. Otros sistemas de administración de bases de datos relacionales ampliamente utilizados para empresas incluyen la base de datos Oracle, MySQL, PostgreSQL (una base de datos relacional de código abierto) y Microsoft SQL Server. Las estructuras RDBMS se usan comúnmente para realizar cuatro operaciones básicas: CRUD (crear, leer, actualizar y eliminar), que son fundamentales para respaldar la administración de datos consistente.

9.13.2. Sistemas de Gestión de base de datos no relacionales

AyudaPlay (s.f) Las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas. Esto no significa que no puedan usar el lenguaje SQL, pero no lo hacen como herramienta de consulta, sino como apoyo. Por ello también se les suele llamar NoSQL o «no solo SQL».

Otra de sus principales características es que no trabajan con estructuras definidas. Es decir, los datos no se almacenan en tablas, y la información tampoco se organiza en registros o campos.

9.14. Normalización

Balda (2016) El proceso de normalización de bases de datos consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

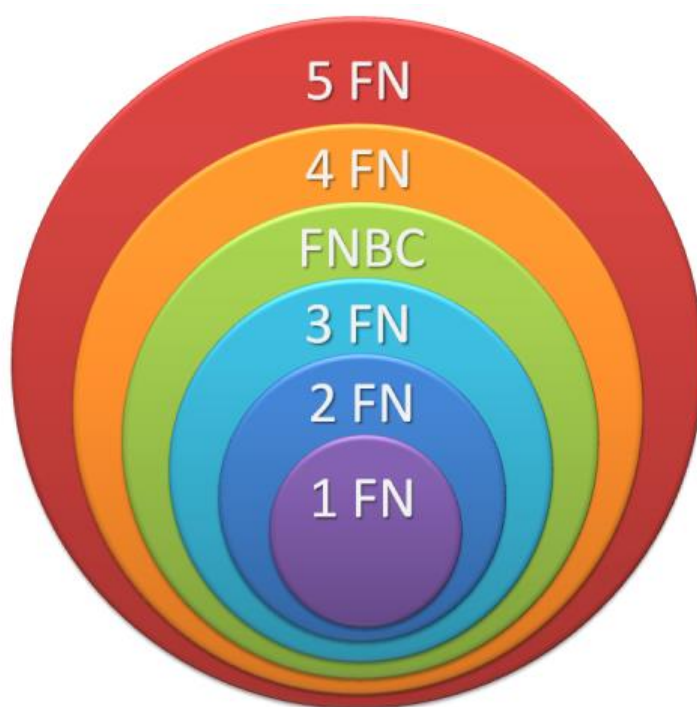
Las bases de datos relacionales se normalizan para:

Evitar la redundancia de los datos.

Disminuir problemas de actualización de los datos en las tablas.

Proteger la integridad de los datos.

Ilustración 8.Relación de las formas normales



9.14.1. Ventajas de la normalización de datos

El enriquecimiento de una base de datos supone una serie de ventajas incuestionables para la empresa. En primer lugar, al evitar duplicidades se simplifica notablemente la gestión y actualización de datos en los registros. Esto también se traduce, en términos generales, a una mejora en la productividad del equipo, además de una mejor gestión de la base de datos.

También facilita la integración de datos con fuentes externas y, en última instancia, refuerza la veracidad y la seguridad de la información con la que se trabaja.

La información y los datos suelen ser el mayor activo de una compañía, máxime a día de hoy en pleno auge de la data. Por ello, la normalización de datos es clave en la gestión de los sistemas de información de una organización, tanto en grandes como pequeñas empresas.

9.15.1. Niveles de normalización de una base de datos

Aunque existen muchos niveles de normalización, son solo tres los más habituales en las empresas. Nos referiremos a ellos como “Formas Normales” y

cada una aplica unas reglas o criterios que determinan el nivel de vulnerabilidad de los datos a posibles inconsistencias o anomalías. A mayor nivel de forma normal, mayor normalización.

9.15.2. Primera forma normal

En primer lugar, se cuenta con tablas separadas para cada conjunto de datos relacionados. Luego se debe asignar una clave primaria a cada grupo de datos, que no contenga atributos nulos.

9.15.3. Segunda forma normal

A los requisitos de la primera forma normal se suma la gestión de múltiples registros. Es decir, si un dato aplica a diferentes localizaciones, debemos crear tablas separadas y relacionarlas a través de una clave ajena o externa.

9.15.4. Tercera forma normal

Para alcanzar este nivel, los datos del mismo registro deben estar estructurados de modo que todos los campos dependan de la clave primaria. Este nivel es conveniente en tablas de datos que requieren de continua actualización. Así evitamos desglosar la información en diferentes tablas.

X. Diseño metodológico

10.1. Enfoque cualitativo asumido y su justificación

El tipo de investigación es aplicada de campo. Aplicada porque se resolvió la problemática del sistema actual mediante la creación de un nuevo sistema. De campo, porque se realizó una exploración de las áreas de control de ventas y facturación.

De la misma forma posee un enfoque mixto, porque se analizaron datos cualitativos y cuantitativos. Ya que proporciona una metodología de investigación que permite analizar el impacto económico, su rentabilidad y los beneficios que traerá a Asociación Club Terraza.

10.2. Población

Según Tamayo (2012) la población es la totalidad de un fenómeno de estudio, incluye la totalidad de unidades de análisis que integran dicho fenómeno y que debe cuantificarse para un determinado estudio integrando un conjunto N de entidades que participan de una determinada característica, y se le denomina la población por constituir la totalidad del fenómeno adscrito a una investigación. En este sentido la población abarca la totalidad de cinco empleados que brindan servicio en el área de restaurante de Asociación Club Terraza y el área de Informática.

10.3. Muestra teórica y sujetos del estudio

Según Sampieri la muestra es una parte o subconjunto de elementos que se seleccionan previamente de una población para realizar un estudio. Supervisor de Informática, supervisor área de restaurante.

El tipo de muestreo en esta investigación es no probabilístico e intencional ya que la población es pequeña

Tabla 1. Estructura organizativa en Asociación Club Terraza

Sujetos	Población	Muestra	%
Área Informática	2	1	100
Área Restaurante	1	1	100
Meseros	2	1	100

10.4. Métodos y técnicas de recolección de datos utilizados

La técnica de recolección de información utilizada fue encuesta empleando como instrumento la entrevista al supervisor del área de informática y el supervisor del área de restaurante para obtener información que posteriormente a través de un análisis mixto logremos generar las conclusiones.

La entrevista fue semiestructurada con el fin de extraer información de manera directa con el entrevistado, con el propósito de que el dialogo sea de manera abierta, con mayor libertad y flexibilidad para responder las preguntas con sinceridad, sin presión y así obtener información más precisa.

MARSHALL y ROSSMAN (1989) definen la observación como "la descripción sistemática de eventos, comportamientos y artefactos en el escenario social elegido para ser estudiado

La observación como herramienta técnica nos permitió determinar el análisis y objeto de estudio, además ayuda a determinar de forma sistemática como se procesa la información y se registran los datos en los procesos de facturación y control de ventas en Asociación Club Terraza.

10.5. Criterios de calidad aplicados: credibilidad, confiabilidad y triangulación

La credibilidad del proyecto de creación del sistema de control de ventas y facturación para el área de restaurante en Asociación Club Terraza se aseguró mediante un proceso riguroso de recolección y análisis de datos. Se realizaron entrevistas y encuestas con el personal del restaurante para entender los problemas y necesidades actuales del sistema manual. Además, se llevaron a cabo sesiones de prueba y validación con usuarios finales durante el desarrollo del sistema para asegurar que los resultados sean precisos y reflejen la realidad operativa del restaurante. El uso de tecnologías probadas y confiables, como SQL Server y Visual Basic, también contribuye a la credibilidad del sistema.

Para garantizar la confiabilidad del sistema, se implementaron procedimientos de pruebas exhaustivas en cada fase del desarrollo. Estas pruebas incluyeron pruebas unitarias, integrales y de sistema para identificar y corregir errores antes

de la implementación final. Además, se establecieron procedimientos de mantenimiento y soporte continuo para asegurar que el sistema opere de manera consistente y eficiente a lo largo del tiempo. La documentación detallada del proceso de desarrollo y de los resultados de las pruebas también contribuye a la confiabilidad del proyecto, proporcionando una base sólida para futuras actualizaciones y mejoras.

La triangulación se aplicó mediante el uso de múltiples métodos de recolección y análisis de datos para asegurar la validez de los resultados. Se combinaron datos cuantitativos (como las métricas de ventas y tiempos de procesamiento) con datos cualitativos (como las opiniones y comentarios del personal y los usuarios). Además, se revisaron estudios de caso y literatura existente sobre sistemas similares en la industria de la restauración para comparar y contrastar los hallazgos. Esta combinación de diferentes fuentes y métodos permitió obtener una visión más completa y precisa del problema y de las posibles soluciones.

10.6. Métodos y técnicas para el procesamiento de datos y análisis de información

Para el procesamiento de datos y el análisis de la información en el desarrollo del sistema de control de ventas y facturación, se utilizaron varios métodos y técnicas:

Recolección de Datos Primarios:

Entrevistas y Encuestas: Se realizaron entrevistas estructuradas y encuestas con el personal del restaurante para identificar problemas y necesidades específicas del sistema actual.

Observación Directa: Se observaron los procesos operativos en el restaurante para entender mejor los flujos de trabajo y los puntos críticos.

Análisis de Requisitos:

Diagramas de Flujo: Se crearon diagramas de flujo para mapear los procesos actuales y los flujos de trabajo propuestos con el nuevo sistema.

Análisis de GAP: Se realizó un análisis de brechas para identificar las diferencias entre el sistema actual y los requisitos del nuevo sistema.

Diseño de Base de Datos:

Modelado ER (Entidad-Relación): Se utilizó el modelado ER para diseñar la estructura de la base de datos, asegurando una organización lógica y eficiente de los datos.

Normalización de Datos: Se aplicaron técnicas de normalización para reducir la redundancia y mejorar la integridad de los datos.

Desarrollo y Pruebas:

Prototipos: Se desarrollaron prototipos iterativos del sistema para recibir retroalimentación temprana y hacer ajustes necesarios.

Pruebas Unitarias e Integrales: Se llevaron a cabo pruebas unitarias para cada módulo del sistema y pruebas integrales para asegurar que todos los componentes funcionen juntos correctamente.

Análisis de Datos:

Estadísticas Descriptivas: Se utilizaron técnicas estadísticas descriptivas para analizar los datos de ventas y facturación, identificando patrones y tendencias.

Generación de Reportes: Se diseñaron reportes personalizados para presentar la información de manera clara y útil para la toma de decisiones.

Implementación y Validación:

Pruebas Piloto: Se implementó el sistema en un entorno piloto antes de su lanzamiento completo, permitiendo identificar y resolver cualquier problema en un entorno controlado.

Capacitación de Personal: Se realizaron sesiones de capacitación para asegurar que el personal del restaurante esté familiarizado con el nuevo sistema y pueda utilizarlo de manera efectiva.

XI. Discusión de resultados o hallazgos

El software de facturación realizado para Asociación Club Terraza, se desarrolló haciendo uso de la metodología de cascada permitiendo conocer el objetivo del trabajo para poder concluir cada una de las fases que tiene este modelo secuencial.

Por consiguiente, este modelo facilita el tener una visión claro de cuáles son los requisitos por parte del cliente, además que permite conceptualizar un mejor análisis de las ventajas y desventajas que el software puede tener para la empresa.

En las fases de este modelo se puede crear un método de desarrollo y saber cuáles son los alcances que pueda tener el software, así como también la funcionalidad de este para establecer estándares de calidad del software, logrando el objetivo de cumplir y brindar una solución óptima para la problemática del Club.

11.1. Fase 1: Análisis del sistema

Esta etapa es esencial para la determinación de los objetivos y problemas que se desean cumplir y dar solución, de la misma forma se detallan los requisitos del proyecto para su implementación, esto permite orientar y definir los problemas que tiene el sistema de facturación no automatizada que se ha utilizado en Club Terraza.

11.1.1. Definición del problema

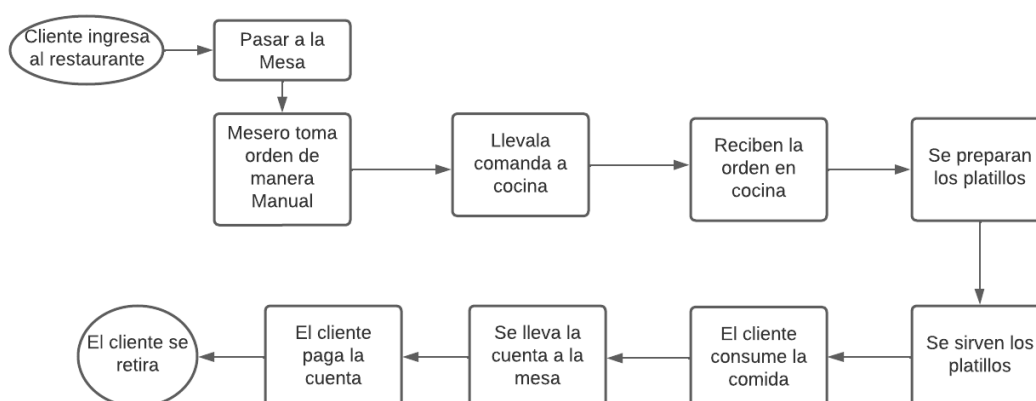
Club Terraza desea automatizar sus procesos de facturación ya que presenta descontrol en la información.

El proceso de facturación que se lleva a cabo en las áreas de restaurantes en Club Terraza, se realiza de forma manual haciendo uso de calculadoras, formatos en hojas de papel, entre otros, los cuales ayudan a agilizar el proceso, sin embargo, no cumplen las expectativas para una mayor eficiencia en el tiempo del servicio.

11.1.2. Proceso de facturación

El proceso de facturación inicia cuando un cliente hace un pedido al restaurante, el mesero toma la orden y la lleva a cocina, posteriormente, una vez que el pedido está listo, el mesero lo lleva a la mesa, el cliente come y una vez que termina procede a hacer el pago. En caja el mesero le dice el pedido, la cajera lo recepciona y realiza los cálculos de forma manual.

Ilustración 9. Proceso de facturación



11.2. Identificar las necesidades del cliente

Se puede apreciar que existe una mala gestión al momento de hacer la facturación. Donde se encontró que no se cuenta con la debida organización, resultando en inconvenientes para tener una mejor optimización del tiempo y recursos del Club, generado informes poco precisos y desactualizados al supervisor.

11.2.1. Análisis técnicos y económicos

- Los costos de desarrollo y equipo para la implementación no deben exceder los \$1800 dólares americanos.
- El sistema deberá ser entregado a mediados del mes de Julio

11.3. Viabilidad del sistema

Un estudio de viabilidad en el desarrollo de un software es el que se encarga de constatar la disponibilidad de los recursos de la empresa, tomando en cuenta

tres puntos muy importantes: económica, operacional y técnica, donde se plantean los parámetros globales estructurados por puntos del proyecto.

11.4. Viabilidad Técnica

En este proceso se realizó un análisis y registro de los diferentes equipos con los que cuenta Club Terraza, se hizo una verificación de los datos técnicos de cada equipo si cumplen con los requerimientos para poner en marcha el sistema de facturación.

Tabla 2. Componentes Físicos de la empresa

Equipo	Cantidad
PC Intel NUC NUC8i3BEH	1
Impresora Epson l3110	1
Cable Ethernet Cat 6	1

De acuerdo a las condiciones y los equipos con los que la empresa cuenta se ha determinado que estos son los necesarios para una buena optimización en cuanto a la implementación del sistema.

11.5. Viabilidad Económica

En este punto se realiza la descripción de los componentes, requerimientos técnicos y equipos estimados para la implementación del software, quedando así notificada la parte administrativa del Club, la cual acepta la inversión de los costos del proyecto, debido a que estos tendrán un gran beneficio al generar mayores ganancias económicas.

En la siguiente tabla se detallan los componentes adicionales para obtener un mejor rendimiento y optimización del sistema.

Tabla 3. Presupuesto proyecto

Nombre	Precio con IVA	cantidad	Total
Sistema de Facturación e Inventario	\$ 500	1	\$ 500
	TOTAL		\$ 500

Tabla 4. Resultados técnicos PC

Características: PC con el sistema instalado
Procesador: Intel® Core™ i3-9100
Memoria RAM: 8 GB DDR4-SDRAM
Almacenamiento: 1 TB Disco Duro
Windows 10 Pro

Tabla 5. Presupuesto de implementación del sistema

Descripción	Precio
Desarrollo del Software	\$ 500
Total	\$ 500

11.6. Viabilidad operacional

Club Terraza, con implementación de un sistema de facturación obtendrá mayor capacidad en el flujo de la información, ya que al realizar todos los procesos de forma manual existe una problemática para poder agilizar dichos procesos puesto que no cuentan con un sistema automatizado.

En cada uno de los módulos se garantiza calidad e integridad de la información, y un buen rendimiento en el funcionamiento, así como también la presentación de interfaz de diseños amigables para el usuario logrando así el objetivo de contar con una interfaz de usuario de fácil uso, reforzando que el usuario tenga una rápida adaptabilidad al sistema.

El sistema está desarrollado haciendo uso del entorno de programación de Visual Basic. El cual genera grandes aspectos positivos al agilizar y automatizar los procesos de facturación, dichos aspectos son mencionados a continuación:

1. Posee una curva de aprendizaje muy rápida.
2. Integra el diseño e implementación de formularios de Windows.
3. Permite usar con facilidad la plataforma de los sistemas Windows, dado que tiene acceso prácticamente total a la API de Windows, incluidas librerías actuales.

4. Es uno de los lenguajes de uso más extendido, por lo que resulta fácil encontrar información, documentación y fuentes para los proyectos.

Apoyado en las diferentes ventajas previamente mencionadas, el software brinda una excelente solución en la organización, optimización y eficiencia de los tiempos que se emplean en los procesos internos de Asociación Club Terraza.

Viabilidad legal

La viabilidad legal determina la existencia de alguna posible restricción de carácter legal en lo relativo a la realización efectiva de una inversión en un proyecto. El resultado determinará su desarrollo y las dificultades que puede llegar a afrontar.

En el presente proyecto se respeta y se hace cumplir la ley de los derechos de autor cumpliendo con todas las prerrogativas que establece, para evitar multas o demandas a la hora de implementar el sistema.

Se utilizó Visual Studio, entorno de desarrollo integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C# y Visual Basic.

Demstrado en los puntos anteriores, tanto los estudios de viabilidad técnica, legal, financiera económica y operativa resultan ser favorables para el proyecto, lo que determina su viabilidad. El proyecto resulta ser una buena herramienta para la empresa, además a un costo no mayor que puede ser recuperado a mediano plazo.

11.7. Funciones del sistema

El supervisor del área de informática de la empresa es el que asigna los roles de trabajo para cada colaborador en la asociación club terraza.

Supervisor: Obtendrá cargos administrativos del sistema, podrá gestionar y modificar información siguiendo todas las normativas de la empresa en cada módulo para ser modificado o actualizado.

El sistema cuenta con características propias utilizadas para el proceso de generar factura, agrega automáticamente los detalles de venta que hayan sido completados en el proceso facturación.

Cajero: Podrá generar factura e imprimirla.

Base de datos: Almacenara la información de los módulos de facturación y control de ventas. Esta información será gestionada con el gestor de base de datos de SQL Server Management.

11.8. Fase 2: Análisis de los requisitos del software

11.8.1. Requisitos solicitados por el cliente

El supervisor de tecnología de información solicita que el sistema realice acciones específicas para cada uno de los módulos.

Facturación

1. Desarrollar módulo de facturación automatizado que sea fácil de utilizar que logre recopilar todos los datos necesarios para generar la factura.
2. Agregar módulo de registro de clientes y de actualizar sus datos.
3. Agregar módulo de registro de productos que se encuentren abastecidos.
4. Agregar módulo de registro de empleados
5. Se podrá generar un documento digital de la factura con su detalle.
6. Para la facturación se verificará si el cliente ya se encuentra previamente registrado en el sistema.

Restricciones

1. No se podrá editar datos de la factura una vez que el proceso este completado.
2. Los campos del detalle de factura: producto y precio no podrán ser actualizados.
3. No se podrá modificar datos del usuario que emite la factura.

Descripción de la información.

Se describirán las características de interfaz de usuario en entrada y salida de la información en el sistema. Una vez teniendo claro la función de cada uno de los usuarios o empleados que interactúan con el sistema tengan sus roles definidos.

Un estudio de campo por medio de la observación nos permite delimitar las acciones y permiso que tiene cada usuario a nivel interno en cada proceso.

Tabla 6. Formulario de análisis de proceso de facturación

Información		Si	No	Nota
1	Se tiene registro de clientes.		x	
2	Los datos de factura tienden mostrar errores o inconsistencias.		x	
3	Se tiene registro de los empleados.		x	
4	La información de venta se respalda de forma segura en la empresa.	x		
5	La factura siempre se encuentra organizada para brindar informe.	x		

Se tiene como resultado del formulario de análisis del proceso de factura que:

- No se cuenta con un registro de cliente que sea ordenado.
- En los cálculos o productos entregados del detalle de venta pueden ocurrir errores de escritura o cantidades a lo que conlleva un mal control.
- No existe un control de la información para el área de tecnología de información.

- Para realizar búsquedas de facturas conlleva mucho tiempo.

Es por ello que se necesita un software que sea capaz de gestionar y corregir dichas dificultades y mejore la buena administración y procesos que se realizan en la empresa.

11.9. Fase 3: Diseño del software

11.9.1. Diseño de la base de datos

En el proceso de recolección de datos y análisis de la información que se utilizan en Club Terraza para llevar un orden en el flujo de la información, se pudo apreciar que existe un descontrol de los datos, resultando en una limitante para poder contar con una buena organización de las entidades con sus atributos. Se lleva a un punto efectivo para el estudio de la información.

Normalización de la base de datos

La normalización en una base de datos consiste en designar y aplicar una serie de reglas para poder organizar los datos de una base de datos. La normalización es un punto clave muy importante al momento de desarrollar la base de datos de un sistema ya que mediante esta se puede evitar cualquier tipo de anomalía, redundancia y borrado que pudiera haber en el diseño de tablas y en la relación entre la información.

Del proceso de normalización se obtiene un conjunto de datos estructurado estable con modelo lógico fácil de escalar de acuerdo a los requerimientos de la empresa.

En el proceso de analizar los datos se obtuvieron datos relacionados directamente con la facturación en el área de restaurante de Asociación Club Terraza.

Tabla 7. Datos recopilados Facturación

Numero factura	Fecha	Mesero	Cliente	Producto	Cantidad	Precio	Categoría	Total1
106	08-Jun	Mario	Francisco	Club Sándwich	1	C\$245	Alimentos	C\$ 245
107	09-Jun	Juan	Edwin	Vigorón Terraza	1	C\$125	Alimentos	C\$ 125

Al momento de realizar la normalización es muy importante que no existan datos duplicados, de esta forma se enriquece la base de datos de manera clara, optimizando espacio de almacenamiento y agilizando tiempos de respuesta de la base de datos.

Primera forma normal (1FN)

Esta primera forma normal nos ayuda a eliminar los valores repetidos y no atómicos dentro de una base de datos.

Formalmente, una tabla está en primera forma normal si y solo si:

1. Elimine cualquier grupo de datos que se repita (es decir, tenga cuidado con las columnas o filas duplicadas dentro de la misma tabla)
2. Cree tablas separadas para cada grupo de datos relacionados
3. Cada tabla debe tener una clave principal (es decir, un campo que identifica cada fila con un valor único no nulo).

Las siguientes tablas están en primera forma normal:

Tabla 8. Productos

Productos						
idproducto	Nombre	descripción	Precio venta C\$	existencia	categoría	Costo
1	Club Sándwich	Con papas	245	12	Alimento	190
2	Vigorón Terraza	Chicharrón crujiente	150	10	Alimentos	80

Tabla 9. Detalle Venta

Detalle Venta					
iddetalleV enta	ideventa	idproducto	cantidad_producto	Nombre	precio_uni C\$
1	106	1	1	Club Sándwich	245
2	107	2	1	Vigorón Terraza	150

Tabla 10. Ventas

Venta						
IdVenta	fecha	nom_ cliente	dir_ cliente	dni	nom_vendedor	Nrodocumento
106	08-06-2022	Francisco Romero	Carretera sur	1325	Mario Gabriel pavón	0000009
107	09-06-2022	Edwin Martínez	Altamira	1460	Juan José Arias	0000010

La tabla venta y detalle venta definen sus atributos que se relacionan, por tanto, se debe incluir cada registro dependiente por naturaleza de los procesos de facturación.

En la tabla venta se define datos al flujo de información y producto, si se sabe que en una factura se puede agregar muchos productos al detalle de factura. Utilizando el método alternativo de convertir en Primera Forma Normal.

De modo que se encuentren funcionalmente dependientes tal como ocurre en el proceso de venta, no puede agregar productos a la venta si no existen los productos.

Segunda Forma Normal (2FN)

Una tabla se encuentra en Segunda Forma Normal cuando está en Primera Forma Normal y no contiene dependencias parciales. Por consiguiente, una tabla Primera Forma Normal automáticamente está en Segunda Forma Normal si su clave primaria está basada solamente en un atributo simple. Las tablas a

Segunda Forma Normal nos permitirán eliminar cualquier anomalía que pudiera existir en una relación.

Las siguientes tablas se presentan en Segunda Forma Normal ya que se crean tablas independientes para conjuntos de valores que se apliquen a varios registros. Y se relacionan tablas con una clave externa.

Productos						
idproducto	Nombre	descripción	Precio venta C\$	existencia	categoría	Costo
1	Club Sándwich	Con papas	245	12	1	190
2	Vigorón Terraza	Chicharrón crujiente	150	10	1	80

Empleado								
idempleado	Nombre	Apellido	DNI	Rol	Sexo	Estado	Dirección	Estado Civil
1	Javier	Flores	2321	admin	Masculino	Activo	Altamira	Solero
2	Enrique	Dávila	2185	Mesero	Masculino	Activo	Los Robles	Casado
3	Edmara	Jarquín	2565	admin	Femenino	Activo	San Judas	Soltera

DetalleVenta					
correlativo	idventa	codigo_Producto	cantidad_producto	Nombre_producto	precio_unitario C\$
1	106	1	2	Club Sándwich	245
2	107	2	2	Vigoron Terraza	150

Venta					
no_factura	fecha	nom_Cliente	dni	Empleado	Nrodocumento

106	08-06-2022	Francisco Romero	1325	Mario Gabriel pavón	0000009
107	09-06-2022	Edwin Martínez	1460	Juan José Arias	0000010

Clientes					
Idcliente	DNI	Nombre	Apellidos	Teléfono	Dirección
1	1325	Francisco	Romero	88658232	Carretera Sur
2	1460	Edwin	Martínez	86572345	Altamira

En este proceso de normalización se identifican cada una de la dependencia de claves foráneas, en las que se coloca la clave foránea con la finalidad de combinar la inserción en cada registro.

Tercera Forma Normal (3FN)

Una relación se encuentra en Tercera Forma Normal si, y solo si, se encuentra en 2FN y si los atributos no clave dependen de forma no transitiva de la clave primaria.

Para analizar si una relación no se encuentra en 3FN buscamos un subconjunto de atributos que no pertenezca a la clave primaria y que, al cambiar el valor de un atributo, necesariamente cambia el valor de los demás atributos del subconjunto.

En la que es necesario extraer los atributos del cual dependen funcionalmente, la teoría expone que debe existir un atributo que permita relacionar las N tablas que se deriven en el proceso.

Las siguientes tablas se presentan en Tercera Forma Normal ya que se han eliminado los campos que no dependen de la clave. Y el contenido de un grupo

de campos se aplica a más de un registro en una determinada tabla, es recomendable colocar estos campos en una tabla independiente.

Productos						
idproducto	id categoría	Nombre	descripción	Existencia	Precio venta C\$	costo
1	1	Club Sándwich	Con papas	12	245	190
2	1	Vigorón Terraza	Chicharrón crujiente	10	150	80

Empleado								
id empleado	Nombre	Apellido	DNI	Id cargo	Sexo	Estado	Dirección	Estado Civil
1	Javier	Flores	2321	1	Masculino	Activo	Altamira	Solero
2	Enrique	Dávila	2185	2	Masculino	Activo	Los Robles	Casado
3	Edmar a	Jarquín	2565	1	Femenino	Activo	San Judas	Soltera

DetalleVenta				
IDdetalle venta	idventa	codigo_Producto	cantidad_producto	precio_uni C\$
1	106	1	2	245
2	107	2	2	150

Venta

idventa	fechaventa	Id cliente	dni	idEmpleado	Nrodocument o
106	08-06-2022	1	1325	1	0000009
107	09-06-2022	2	1460	2	0000010

Clientes					
Idcliente	DNI	Nombre	Apellidos	Teléfono	Dirección
1	1325	Francisco	Romero	88658232	Carretera Sur
2	1460	Edwin	Martínez	86572345	Altamira

Usuario			
Idcliente	idempleado	Usuario	Contraseña
1	1	jflores	123
2	2	Edavila	asd

Categoría	
idcategoría	Descripción
1	Alimentos
2	Bebidas

Cargo	
idcargo	Descripción
1	admin
2	Mesero

11.10. Diccionario de datos

Un diccionario de datos es un conjunto de definiciones que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema. Trata de documentar los metadatos más ligados a su almacenamiento en la base de datos.

En otras palabras, incluye todos los aspectos técnicos como el tipo de dato, formato, longitud, posibles valores que puede tomar e, incluso, transformaciones sufridas, sin olvidar la definición de cada campo.

A continuación, se presenta el diccionario de datos para el sistema:

Cliente					
Columna	Tipo y longitud de dato	Nulo	Valores Predeterminado	Enlace s a	Significado
Idcliente (Primaria)	int	No	Default		Identificador primario a nivel de base de datos
dni	int	No	Default		Identificar el cliente en filtro de búsqueda, dato Único del cliente
Nombre	nvarchar(50)	No	Default		Almacena nombre del cliente
Apellido	nvarchar(50)	No	Default		Almacena apellidos del cliente
Dirección	nvarchar(50)	Sí	Default		Domicilio del cliente
Teléfono	int	No	Default		Número de teléfono del cliente
DetalleVenta					

Columna	Tipo y longitud de dato	Nulo	Valores Predeterminado	Enlaces a	Comentarios
iddetalleventa (Primaria)	int	No	Default		Código de todas las transacciones de ventas en el sistema
idproducto	int	No	Default	producto -> idproducto	Identifica el producto que se agrega al detalle para realizar las operaciones a nivel de código
idventa	int	No	Default	venta-> idventa	Identifica a que factura pertenece el detalle
Cantidad	int(11)	No	Default		Ingres las N cantidad de producto que adquiere el detalle.
preciounitario	decimal(4,2)	No	Default		Precio asignado por área administrativa

Venta					
Columna	Tipo y longitud de dato	Nulo	Valores Predeterminado	Enlaces a	Comentarios

idventa (Primaria)	int	No	Default		Código único en el sistema sirve para llevar el control de facturas emitidas en el sistema
idempleado	int	No	NULL	Empleadp-> idempleado	Ingresa el código del empleado para almacenar datos únicos del proceso.
idcliente	int	No	NULL	cliente -> idcliente	Ingresa el código del cliente para almacenar datos únicos del proceso.
Nrodocumento	nvarchar(50)	No	NULL		Identificar número de factura en filtro de búsqueda, dato Único de la factura
Tipodocumento	nvarchar(40)	No	NULL		Identifica el tipo de factura
FechaVenta	date	No	NULL		Fecha que se emite la factura, ayuda para filtrar información de ventas.

Producto					
Columna	Tipo y longitud de dato	Nulo	Valores Predeterminado	Enlaces a	Significado
idproducto (Primaria)	int	No	Default		Código único que ayuda a identificar el producto
idcategoría	int	No	NULL	categoría -> idcategoría	Ingresa código de la categoría para almacenar Datos únicos del proceso

nombre	nvarchar(50)	No	NULL		Nombre del producto en el sistema, ayuda en el filtro de información
descripción	nvarchar(60)	No	NULL		Breve descripción del producto
stock	int	No	Default		Cantidad de stock del producto en el sistema
preciocompra	decimal(4, 2)	Sí	NULL		Almacena precio definido por el área administrativa
costo	decimal(4, 2)	No	NULL		Almacena el costo de compra o de elaboración del producto

Empleado				
Columna	Tipo y longitud de dato	Valores Predeterminado	Enlaces	Significado
IdEmpleado (Primaria)	int(11)			Código que se genera ingresar datos de nuevos usuarios
idcargo	int	NULL	Cargo-> idcargo	Define los permisos de actualizar, eliminar o agregar datos al sistema.
dni	int	NULL		Identificador del empleado
nombre	nvarchar(50)	NULL		Nombre identificador en el sistema
apellidos	nvarchar(50)	NULL		apellido identificador en el sistema
sexo	nvarchar(30)	NULL		Almacena el sexo del empleado
dirección	nvarchar(50)	NULL		Verifica el estado del usuario si está activo o inhabilitado en el sistema

Estado civil	nvarchar(30)	NULL		Conocer estado civil del empleado para ciertos beneficios que ofrece el área administrativa
estado	nvarchar(30)	NULL		Verifica el estado del usuario si está activo

Categoría					
Columna	Tipo y longitud de dato	Nulo	Predeterminado	Enlaces a	Significado
Idcategoría (Primaria)	int	No	Default	categoría	Código que se genera para identificar la categoría del producto
Nombre	nvarchar(50)	No	Default		Almacena el nombre de la categoría

Cargo					
Columna	Tipo y longitud de dato	Nulo	Predeterminado	Enlaces a	Significado
Idcargo (Primaria)	int	No	Default	categoría	Código que se genera para identificar cargo del empleado
Nombre	nvarchar(50)	No	Default		Almacena el nombre del cargo del empleado

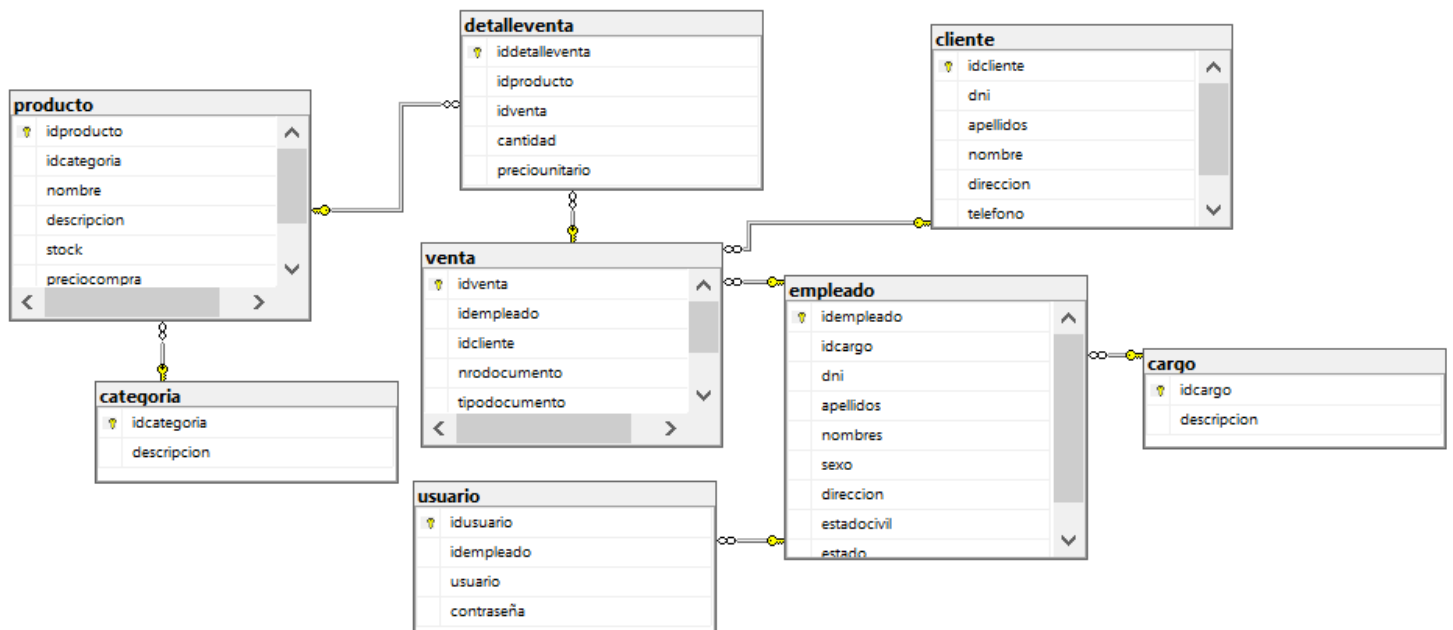
Cargo					
Columna	Tipo y longitud de dato	Nulo	Predeterminado	Enlaces a	Significado
Idcargo (Primaria)	int	No	Default	categoría	Código que se genera para identificar cargo del empleado
Nombre	nvarchar(50)	No	Default		Almacena el nombre del cargo del empleado

usuario					
Columna	Tipo y longitud de dato	Nulo	Predeterminado	Enlaces a	Significado
idusuario (Primaria)	int	No	Default		Código que se genera para identificar al usuario
idempleado	int		Default	Empleado-> idempleado	Almacena el código del cargo del empleado
Usuario	nvarchar(50)		Default		Usuario para ingresar al sistema
Contraseña	nvarchar(50)		Default		Contraseña para ingresar al sistema

11.11. Diagrama Entidad- Relación

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. La siguiente imagen presenta el diagrama entidad – relación de la base de datos para el sistema.

Ilustración 10. Diagrama Entidad - Relación



11.12. Diseño de interfaz de Usuario

A continuación, se presentan imágenes del entorno de interacción, cómo es y la estructura de la interfaz gráfica del sistema.

Ilustración 11. Login del sistema.

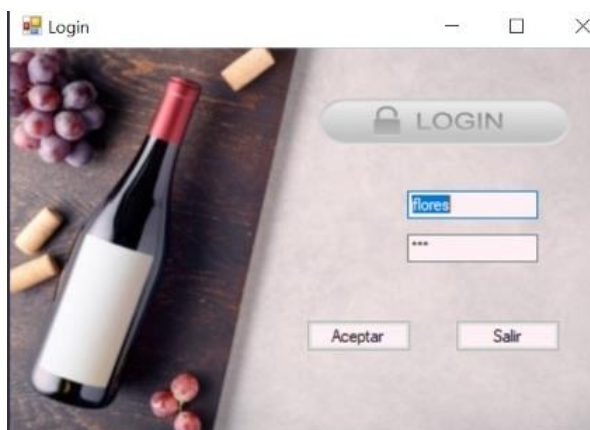
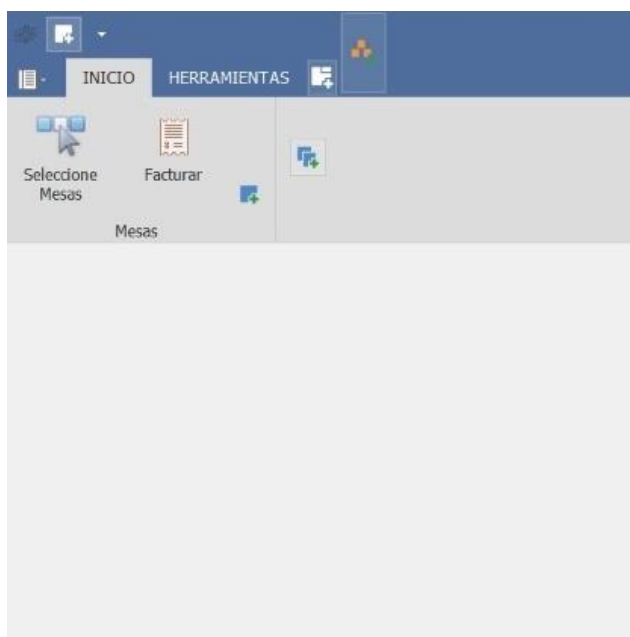
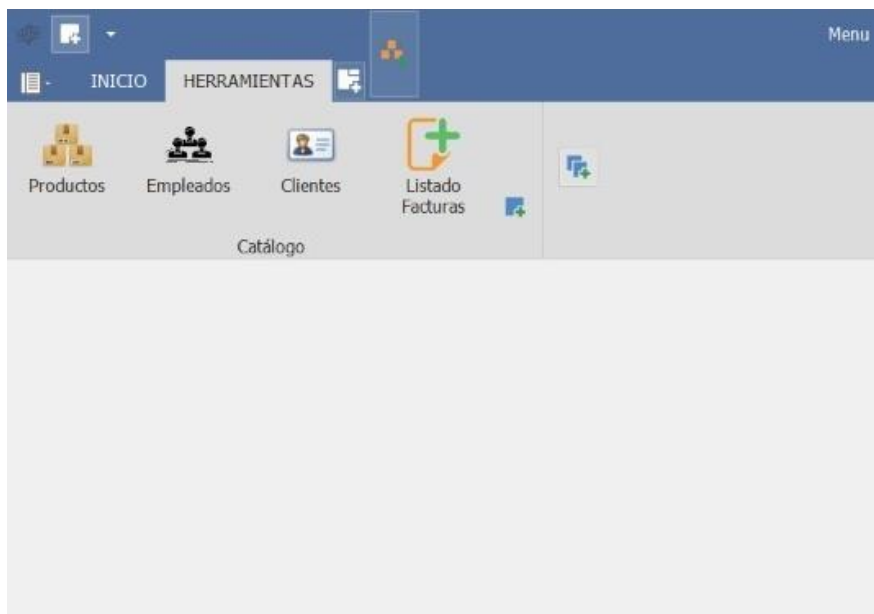
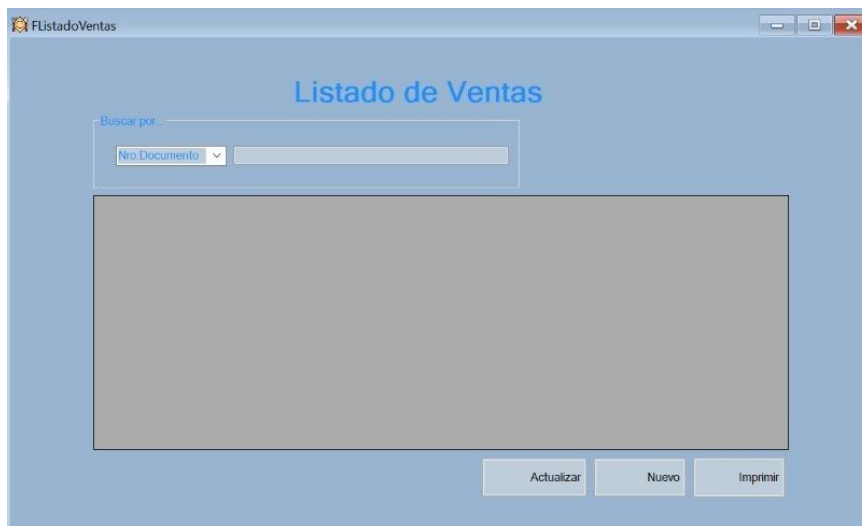


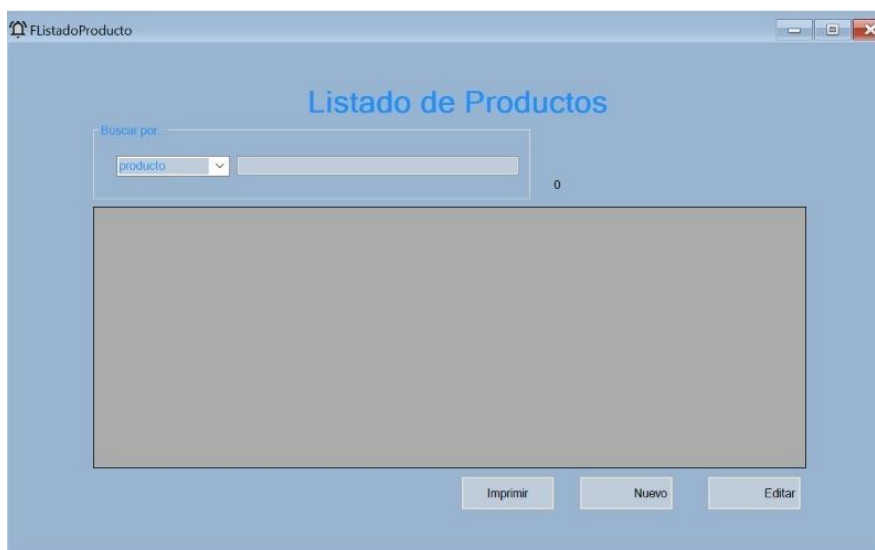
Ilustración 12. Menús principales del sistema. Con la sección Herramientas e Inicio.



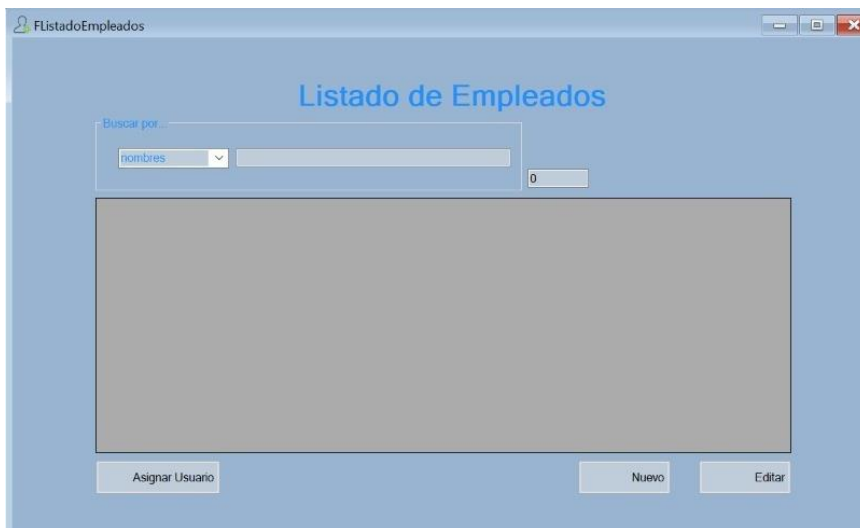
Listado de ventas. Acá observamos una lista de las ventas, las cuales podemos buscar por medio del N.º de documento, agregamos, actualizamos y mandamos a imprimir.



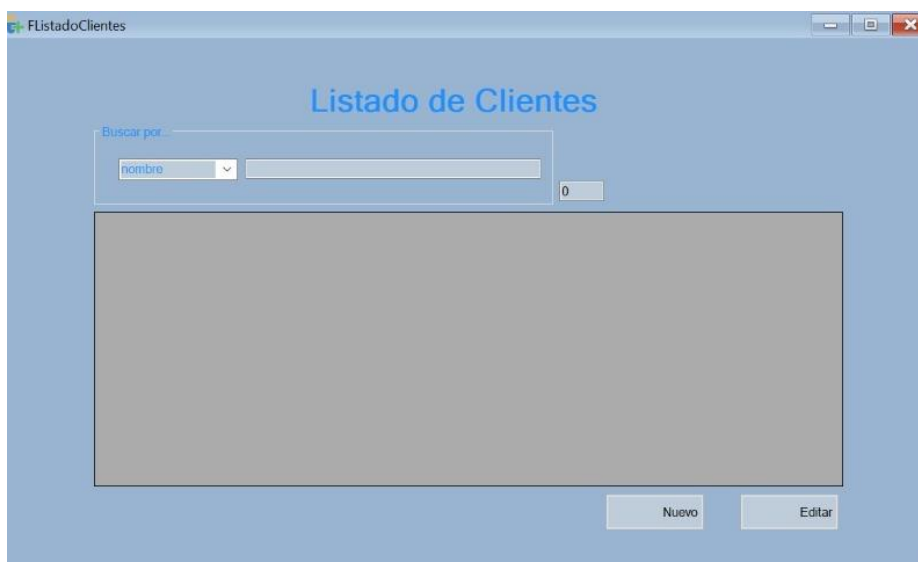
Listado de productos. Acá observamos un listado de los productos, los cuales podemos buscar por medio del N.º d producto, agregamos un nuevo producto, editamos un producto existente y mandamos a imprimir.



Listado de empleados. Acá observamos una lista de los empleados que laboran, los cuales podemos buscar por medio de su nombre, agregamos un nuevo empleado, actualizamos y podemos asignar un usuario.



Listado de Clientes. Acá observamos una lista de los clientes del club terraza, los cuales podemos buscar por medio del su nombre, agregamos un nuevo cliente o bien lo editamos.



Formulario para agregar un nuevo empleado

The screenshot shows a Windows-style window titled "FEmpleado" with a blue header bar. The window contains a form with the following fields and controls:

- Cargo:** A dropdown menu with a downward arrow, followed by a button labeled "Cargo".
- Nombre:** A single-line text input field.
- Apellidos:** A single-line text input field.
- Sexo:** A dropdown menu with "Masculino" selected and a downward arrow, followed by an empty single-line text input field.
- Dni:** A single-line text input field.
- Direccion:** A single-line text input field.
- Estado Civil:** A dropdown menu with "Soltero(a)" selected and a downward arrow.
- Estado:** A dropdown menu with "Activo" selected and a downward arrow.
- Buttons:** Two buttons at the bottom, "Aceptar" and "Salir", each with a light blue gradient.

Formulario para agregar un nuevo cliente

The screenshot shows a Windows-style window titled "FCliente" with a blue header bar. The window contains a form with the following fields and controls:

- Nombres:** A single-line text input field.
- Apellidos:** A single-line text input field.
- Direccion:** A single-line text input field.
- Telefono:** Two single-line text input fields side-by-side.
- DNI:** A single-line text input field.
- Buttons:** Two buttons at the bottom, "Aceptar" and "Salir", each with a light blue gradient.

Formulario para agregar un nuevo usuario.

The screenshot shows a Windows-style window titled "FUsuario". Inside, there are two main sections. The first section, "Datos del Empleado", contains three input fields: "Nombres y Apellidos:" (a wide field), "Dni:" (a standard field), and "Cargo:" (a standard field). The second section, "Usuario", contains two input fields: "Usuario:" and "Contraseña:". To the right of the password field is a checkbox labeled "Ver Contraseña". At the bottom of the form are two buttons: "Grabar" and "Salir".

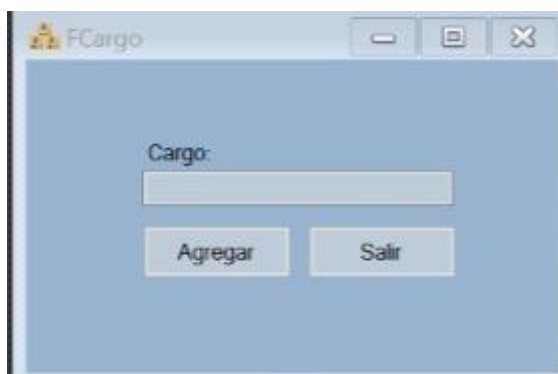
Formulario para agregar un nuevo producto.

The screenshot shows a Windows-style window titled "FProducto". It contains several input fields arranged vertically: "Categoria:" (with a dropdown arrow and a "Categoria" button next to it), "Nombre:" (a wide field), "Descripción:" (a wide field), "Stock:" (a standard field), "Precio Compra:" (a standard field), and "Precio Venta:" (a standard field). At the bottom are two buttons: "Aceptar" and "Salir".

Categoría, formulario creado para asignar la categoría a un producto.

The screenshot shows a Windows-style window titled "FCategoria". It contains a single input field labeled "Categoria:". Below the field are two buttons: "Agregar" and "Salir".

Formulario creado para asignar los cargos.



11.13. Fase 4 Desarrollo del software

La fase de desarrollo del software se centra en la funcionalidad del sistema de control de ventas y facturación para Asociación Club Terraza, empleando el lenguaje de programación Visual Basic y haciendo uso del IDE Visual Studio 2017.

Para contar con un excelente flujo de datos se hizo uso de un gestor de base de datos basado en el lenguaje Transact-SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

SQL Server es uno de los principales sistemas de gestión de bases de datos relacional del mercado que presta servicio a un amplio abanico de aplicaciones de software destinadas a la inteligencia empresarial y análisis sobre entornos corporativos.

11.13.1. Programación

Una de las principales funciones del lenguaje de programación es escribir el código que permita la comunicación usuario máquina, por eso al momento de desarrollar es vital hacer uso de un lenguaje de programación apropiado que se adapte a las necesidades del programador, permitiendo obtener código fácil de interpretar.

Para desarrollar el sistema de facturación para el área de restaurante en asociación Club Terraza se hizo uso de Visual Basic como lenguaje base para codificación y depuración del sistema.

Durante la programación se utilizaron varias herramientas que permitieron agilizar el proceso al programar el sistema, tal es el caso de DevExpress, este ofrece a los desarrolladores de aplicaciones una de las suites más completas de componentes de interfaz de usuario (UI) en todas las plataformas .NET tales como Windows Forms, MVC, ASP.NET, Silverlight y Windows XAML.

También se aplicó a la programación Crystal Reports, este es una herramienta que apoya a los programadores a lograr un mejor análisis de la información a través de informes creados con formatos enriquecidos a partir de prácticamente cualquier fuente de datos.

11.13.2. Codificación

Definir las pautas a seguir del sistema para llevar a cabo registros y agrupaciones que nos ayuda a facilitar el registro y la transmisión de información, así como a controlar todos los procesos.

En esta fase hay una conversión de todas las partes lógicas de la interfaz convirtiéndola en líneas de código. En esta etapa de escritura de la solución del problema a un código reconocible para la computadora.

La codificación debe mantener un patrón de desarrollo, estructurando la parte visual y el código además de definir la identificación de cada elemento para evitar errores lógicos del sistema. Para mantener la integridad de datos en el sistema se realiza una validación de manera que proporcione información confiable y correcta.

11.13.3. Documentación

Se muestra todos aquellas técnicas o normas que el usuario debe de cumplir para observar los procesos que se pueden realizar y los resultados que podemos obtener. Añadiendo diseños al sistema para una mejor interpretación.

11.14. Fase 5: Prueba y mantenimiento

Se realizan pruebas unitarias por cada uno de los módulos de datos, haciendo uso de datos reales brindados por Club Terraza, con el propósito de verificar el correcto funcionamiento de los módulos y la integración de los datos en estos.

En esta fase del desarrollo del sistema ya funcional se realizan pruebas a los diferentes procesos con lo que cuenta cada módulo, Se realiza el análisis de los

resultados de los procesos creados por el sistema, constatando el buen funcionamiento de cada función que brinda el sistema al usuario.

Se hizo la verificación de pruebas directas en el módulo de facturación para verificar que el sistema cumpla con los requerimientos solicitados por el cliente. Verificando que sí cumple con la parte fundamental de la lógica de negocio de Club Terraza.

11.15. Fase 6: implementación del software

En la implementación del sistema de facturación y control de ventas es donde se presenta las combinaciones de código e interfaces funcionales, obteniendo así un software de fácil uso y comprensión para el usuario final que solvento las necesidades que presentaba la Asociación Club Terraza.

Para el proceso de desarrollo se completó en un lapso de 3 meses cumpliendo con los objetivos establecidos de validación de datos ingresados en el sistema obteniendo un buen control al agregar, verificar o realizar un proceso de facturación y control de ventas.

XII. Referencias Bibliográficas

- Alarcón. (2021). Los conceptos fundamentales sobre Programación Orientada Objetos explicados de manera simple. Recopilado de: <https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>
- Bailón. Baltazar. (2021). *Características de la POO. En Algoritmos y codificación. Portal Académico del CCH, UNAM.* Recopilado de: <https://portalacademico.cch.unam.mx/cibernetica1/algoritmos-y-codificacion/caracteristicas-POO>
- Bembibre (2009). Definición de Visual Basic. Recopilado de: <https://www.definicionabc.com/tecnologia/visual-basic.php>
- BigData Analytics (2022). Qué es SQL, sintaxis, conceptos básicos y características. Recopilado de: <https://bigdata-analytics.es/sql/>
- Certus (2022). Ventajas de la programación orientada a objetos. Recopilado de: <https://www.certus.edu.pe/blog/programacion-orientada-a-objetos/>
- Cevallos. (2015) *El software y la ingeniería de software.* Recopilado de: <https://ingsoftwarekarlacevallos.wordpress.com/category/el-software-y-la-ingenieria-de-software/>
- Clavijo. (2022). *Qué es el control de ventas y cómo hacerlo de manera eficaz.* Recopilado de: <https://blog.hubspot.es/sales/que-es-control-de-ventas>
- Club Terraza (2022) *Historia.* Recopilado de: <http://www.clubterrazza.com/>
- Da Silva. (2020) Sistema de control de ventas: ¿qué es y para qué sirve? Recopilado de: <https://www.zendesk.com.mx/blog/sistema-de-control-de-ventas/>
- Dangel. (2010). *Sistemas de Información.* Recopilado por: [https://www.econlink.com.ar/sistemas-informacion/definicion#:~:text=Otros%20autores%20como%20Peralta%20\(2008,de%20una%20empresa%20o%20negocio.](https://www.econlink.com.ar/sistemas-informacion/definicion#:~:text=Otros%20autores%20como%20Peralta%20(2008,de%20una%20empresa%20o%20negocio.)

- Darías. (2021). *Gestor de base de datos: Qué es, funcionalidades y ejemplos*. Recopilado de: <https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>
- EDteam. (2018). *Normalización de Bases de Datos*. Recopilado de: <https://ed.team/blog/normalizacion-de-bases-de-datos>
- Equipo editorial Etecé. (2021). *Lenguaje de programación*. Recopilado de: <https://concepto.de/lenguaje-de-programacion/>
- Gil. (2012). *Ingeniería de requerimientos*. Recopilado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2_-_Ingenier%C3%ADa_de_requerimientos.pdf?sequence=4&isAllowed=y
- Gonzales, Calero, Loaiza. (2019). *Comparación de la metodología de cascada y ágil para el aumento de la productividad en el desarrollo de software*. Recopilado de: <https://repository.usc.edu.co/bitstream/handle/20.500.12421/1208/COMPARACI%D2N%20DE%20LAS%20METODOLOG%CCAS.pdf;jsessionid=1A286F59699BC9739ECE3F116D74442A?sequence=1>
- Hurtado (2021). *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. Recopilado de: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- IBM (2021). *Programación orientada a objetos*. Recopilado de: <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=language-object-oriented-programming>
- Incubicon.(2019). *Tipos de metodologías Ágiles para la gestión de proyectos*. Recopilado de: <https://blog.incubicon.com/tipos-de-metodolog%C3%ADas-agiles-para-la-gestion-de-proyectos>
- Intelequia. (2020). *Ciclo de vida del software: Todo lo que necesitas saber*. Recopilado de: <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
- Ionos. (2020). *Modelo en espiral: el modelo para la gestión de riesgos en el desarrollo de software*. Recopilado de: <https://www.ionos.es/startupguide/productividad/modelo-en-espiral/>
- Lucidchart. (2022). *Qué es un modelo de base de datos*. Recopilado de: <https://www.lucidchart.com/pages/es/que-es-un-modelo-de-base-de-datos#:~:text=Tipos%20de%20modelos%20de%20bases%20de%20dat>

OS,-

Hay%20muchos%20tipos&text=Modelo%20de%20base%20de%20datos
%20jer%C3%A1rquico,de%20datos%20orientado%20a%20objetos

Marín. (2019). Los gestores de base de datos más usados en la actualidad.

Recopilado de: <https://www.inesem.es/revistadigital/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>

Microtech.(2021). *Implantar un sistema de control de ventas y facturación.*

Recopilado de: <https://www.microtech.es/blog/implantar-un-sistema-de-control-de-ventas-y-facturaci%C3%B3n>

Ortiz. (2017). Ingeniería de software: Modelos de desarrollo de software.

Recopilado de: <http://isw-udistrital.blogspot.com/2012/09/ingenieria-de-software-i.html>

Pressman. (2010). *Ingeniería del software: Un enfoque practico.* Recopilado de:

<http://cotana.informatica.edu.bo/downloads/ld->

[Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF](http://cotana.informatica.edu.bo/downloads/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF)

Rock Content. (2019). Lenguaje de programación. Recopilado de:

<https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/>

Sage (s.f). *Facturación.* Recopilado de: [https://www.sage.com/es-](https://www.sage.com/es-es/blog/diccionario-empresarial/facturacion/)

[es/blog/diccionario-empresarial/facturacion/](https://www.sage.com/es-es/blog/diccionario-empresarial/facturacion/)

Salazar. (2016). *Procesos de Scrum.* Recopilado de:

<http://www.prozessgroup.com/procesos-de-scrum/>

Salesforce (2017) Control de ventas. Recopilado de:

<https://www.salesforce.com/mx/products/sales-cloud/todo-sobre-ventas/control-de-ventas/>

Universidad autónoma de México. (2021). *Lenguaje de programación JAVA.*

Recopilado de: <https://docencia.tic.unam.mx/presenciales/Lenguaje-de-programacion-java.html>

Universidad Internacional de Valencia. (2022). *Los tipos de software y sus*

diferencias que debemos conocer. Recopilado de:

<https://www.universidadviu.com/es/actualidad/nuestros-expertos/los-tipos-de-software-y-sus-diferencias-que-debemos-conocer>

Universitat de Carlemany. (2020). *Características de la programación orientada*

a

objetos.

Recopilado

de:

<https://www.universitatcarlemany.com/actualidad/caracteristicas-de-la-programacion-orientada-objetos>

Universitat de Carlemany. (2020). *Metodologías del desarrollo del software*.

Recopilado de:

<https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software>

Vásquez (2020). *4 elementos y pilares básicos de la Programación Orientada a Objetos POO*. Recopilado de: <https://platzi.com/tutoriales/1474-oop/6108-4-elementos-y-pilares-basicos-de-la-programacion-orientada-a-objetos-poo/>

XIII. Anexos

Asociación Club Terraza

Encuesta realizada el día 26 del mes mayo del año 2022.

Estimado trabajador

Instrucciones: A continuación, encontraras una serie de preguntas relacionadas con el nivel de satisfacción respecto a los servicios de facturación que se realizan en el área de restaurante.

- 1. ¿Cómo valora nuestro sistema de facturación y control de ventas?**
 - a. Bueno.**
 - b. Muy bueno.**
 - c. Malo.**
- 2. ¿Considera que es de utilidad estar actualizando manual las facturas?**
 - a. Si**
 - b. No**
- 3. ¿Utilizan algún tipo de programa para actualizar los productos?**
 - a. Si**
 - b. No**
- 4. ¿Al momento de facturar que pone en la descripción de producto?**
 - a. Código**
 - b. Descripción**
 - c. Categoría**
- 5. ¿Qué utiliza cuándo calcula los totales?**
 - a. Celular**
 - b. Calculadora**
 - c. Mental**
 - d. Manual**
- 6. ¿Cree usted que al momento de facturar se lleva mucho tiempo dando un total?**
 - a. Si**
 - b. No**
- 7. ¿Se confunde con frecuencia?**
 - a. Si**
 - b. No**
- 8. Piensa que es porque:**

- a. La interrumpen demasiado
 - b. No se concentra
 - c. Son demasiados productos que sumar
9. ¿Ha manchado facturas?
- a. Si
 - b. No
10. ¿Cada cuánto dan hacer las facturas?
- a. Cada mes
 - b. Cada 2 meses
 - c. Más de 3 meses
11. ¿Ha habido clientes que le reclaman por que la facturación es lenta?
- a. Si
 - b. No
12. ¿Dónde guardan las facturas?
13. ¿Han tenido algún problema con los costos de productos? ¿Cuáles?
14. ¿Sugiere que el nuevo sistema de facturación y control de ventas sea de forma automatizada? ¿Por qué?