

**UNIVERSIDAD CENTRAL
VICERRECTORÍA ACADÉMICA**

ESCUELA DE INFORMÁTICA

**SISTEMA DE AUTOMATIZACIÓN DE PEDIDOS EN EL
RESTAURANTE FINCA LA BANDERA, UTILIZANDO
VISUAL BASIC .NET Y SQL COMO MOTOR DE BASE DE
DATOS**

**MODALIDAD DE TESIS PARA OPTAR POR EL GRADO DE
BACHILLERATO EN INGENIERÍA INFORMÁTICA**

ESTUDIANTE: BRANDON PIEDRA CHACÓN

TUTOR: ING. MAURICIO TORRES

SEDE CENTRAL SAN JOSÉ, COSTA RICA

MAYO, 2025

Tabla de contenido

Tabla de contenido.....	I
Dedicatoria y agradecimiento.....	X
Resumen.....	XI
Abastract.....	XIII
Capítulo I. Problema.....	1
Planteamiento del Problema.....	2
Objetivos.....	2
Objetivo General.....	2
Objetivos Específicos.....	2
Justificación.....	4
Antecedentes.....	6
Antecedentes Internacionales.....	6
Antecedentes Nacionales.....	7
Proyecciones.....	8
Alcances.....	8
Limitaciones.....	9
Capítulo II. Marco Teórico.....	10
Historia de la Programación.....	11
Primeros lenguajes de programación.....	11

Fundamentos de la Programación.....	16
¿Qué es un lenguaje de programación?	16
Tipos de Programación.....	16
Tipos de Lenguajes de Programación.....	18
Paradigmas de Programación	22
Algoritmos de Programación.....	26
Sistema de pedidos	27
¿Qué son los Sistemas de Pedidos?	28
¿Qué son los Sistemas de Inventarios?.....	28
¿Qué son los Métodos Para el Control de Inventarios?.....	28
Tipos de Inventarios	31
Sistemas de Información	34
Sistemas de Procesamiento de Transacciones	35
Sistemas de Información Gerencial.....	36
Sistemas de Control de Procesos de Negocio.....	37
Sistemas de Información de Marketing.....	38
Sistemas de Colaboración Empresarial	39
Sistema de Soporte de Toma de Decisiones.....	39
Sistemas de Información Ejecutiva	40
El Ciclo de Vida de un Sistema de Información	41

Planificación.....	41
Análisis de Requerimientos	41
Diseño.....	41
Desarrollo	42
Prueba	42
Integración y Ejecución	42
Operación y Mantenimiento	42
Requerimientos de Sistema	43
Requerimientos Funcionales.....	43
Requerimientos no Funcionales.....	44
Diseño Integral de Sistemas	45
Arquitectura	45
Estructura de Datos.....	46
Interface	47
Componentes	47
Diagrama UML.....	47
Casos de Uso	48
Base de Datos	49
Historia de las Bases de Datos.....	49
Que son las Bases de Código Abierto (Open Source).....	50

Metodologías de Desarrollo.....	53
Metodologías Tradicionales	53
Cascada.....	53
Incremental	53
Prototipo	54
Metodologías Agiles.....	54
SCRUM	54
Kanban.....	54
Programación Extrema (Extreme programming, XP)	55
¿Qué es el Desarrollo Web?	56
¿Qué Hace un Desarrollador Web?	57
Tipos de Desarrollo web.....	58
Front-end (interfaz).....	58
Backend (servidor)	59
Full Stack.....	60
Ventajas y Desventajas del Desarrollo web	61
Ventajas	61
Desventajas.....	61
Herramientas Para Desarrollo del Proyecto.....	62
IDE.....	62

Editores de Código	62
¿Qué es Visual Basic?	63
Capítulo III. Marco Metodológico.....	66
¿Qué es un Marco Metodológico?.....	67
Importancia de un Marco Metodológico	68
Desarrollo de un Marco Metodológico.....	70
Definición de la Investigación.....	70
Revisión de la Literatura.....	70
Diseño de Investigación:	70
Población y Muestra.....	70
Instrumentos de Recopilación de Datos	71
Procedimiento.....	71
Análisis de Datos	71
Aspectos Éticos.....	71
Plan de Trabajo.....	71
Presupuesto.....	71
Validación del Marco Metodológico.....	72
Redacción del Marco Metodológico.....	72
Revisión Continua	72
Enfoque de la Investigación	72

Tipos de Enfoques	73
Enfoque Cualitativo.....	73
Tipos de Investigación Cualitativa:	74
Enfoque Cuantitativo.....	75
Tipos de Investigación Cuantitativa:	76
Enfoque Mixto.....	76
Tipos de Investigación Mixta	78
Método de Investigación	78
Fuentes de Información	79
Fuentes Primarias	79
Fuentes Secundarias	80
Fuentes Terciarias.....	80
Otros Tipos de Fuente de Información.....	81
Local, Nacional o Internacional.....	81
Textual o Digital.....	82
General o Especializada.....	82
Formales o Informales	82
¿Qué es una Variable?	83
Requisitos Funcionales.....	85
Tipos de Requisitos Funcionales	85

Requerimientos no Funcionales.....	86
Tipos de Requisitos No Funcionales	86
Ventajas Requisitos No Funcionales	86
Desventajas de los Requisitos No Funcionales	87
Ejemplos de Requisitos No Funcionales	87
Plan de Pruebas.....	88
¿Cómo se Realiza un Plan de Pruebas?	89
Tipos de Pruebas de Software	89
Casos de uso	90
Aplicaciones de Casos de Uso.....	91
Componentes del Diagrama de Caso de uso	91
¿Cómo hacer un diagrama de caso de uso?	91
Instrumentos	92
Capítulo IV. Análisis de Resultados.....	93
Plan de Gestión de Riesgos	94
Componentes Principales	94
Tareas de la gestión de riesgos	95
Matriz de riesgos	96
Tipos de riesgos:.....	97
Estudio de factibilidad.....	98

Análisis de factibilidad.....	99
Factibilidad Técnica.	100
Factibilidad Económica.....	101
Factibilidad Operativa.	102
Metodología Utilizada.....	105
Capítulo V. Conclusiones y Recomendaciones.....	107
Conclusiones.....	108
Recomendaciones.....	109
Capítulo VI. Análisis de Requerimientos.....	110
Análisis de Requerimientos.....	111
Requerimientos de usuario.....	111
Requerimientos Funcionales.....	117
Requerimientos No Funcionales.....	121
Modelos UML.....	122
Casos de Uso.....	122
Caso de Uso 1. Login.....	123
Caso de Uso 2. Registro de Usuario.....	124
Caso de Uso 3. Registro de Productos.....	126
Caso de Uso 6. Registro de Categorías.....	127
Caso de Uso 7. Registro de Pedidos.....	129

Caso de Uso 9. Registro de Detalles Pedidos.....	130
Caso de Uso 9. Registro de Detalles Pedidos.....	132
Diagramas de flujo.....	133
Login.....	133
Menú.....	134
Mantenimiento.....	134
Modelo Entidad Relación.....	135
Diccionario de datos.....	136
Pantalla del Login.....	138
Pantalla de ingreso al sistema.....	139
Pantallas de Detalles Pedidos.....	140
Pantallas de Clientes.....	143
Pantallas de Productos.....	145
Pantallas de Facturación.....	146

Índice de Figuras

Figura 1 <i>Historia de la programación</i>	13
Figura 2 <i>Paradigmas de la programación</i>	14
Figura 3 <i>Mapa conceptual fundamentos de programación</i>	17
Figura 4 <i>Lenguaje de programación</i>	19
Figura 5 <i>Lenguajes de alto nivel vs de bajo nivel</i>	20
Figura 6 <i>Lenguajes tipados estático</i>	21
Figura 7 <i>Lenguajes tipados dinámicos</i>	22
Figura 8 <i>Lenguajes de programación</i>	23
Figura 9 <i>¿Qué son los paradigmas de programación?</i>	23
Figura 10 <i>Programación orientada a objetos</i>	24
Figura 11 <i>Programación funcional</i>	25
Figura 12 <i>Programación imperativa (Visual Basic)</i>	26
Figura 13 <i>Programación declarativa</i>	27
Figura 14.....	27
Figura 15.....	28
Figura 16 <i>Métodos para el Control de Inventarios</i>	31
Figura 17 <i>Sistemas de Información</i>	35
Figura 18 <i>Ciclo de vida de un sistema</i>	43
Figura 19 <i>Requerimientos del sistema</i>	45
Figura 20 <i>Diagrama UML</i>	49
Figura 21 <i>Casos de uso</i>	49
Figura 22 <i>Base de datos</i>	50
Figura 23 <i>Que es MySQL</i>	51
Figura 24 <i>Bases de datos open source</i>	53

Figura 25 <i>Metodologías ágiles y tradicional</i>	56
Figura 26 <i>Que Hace un Desarrollador web</i>	58
Figura 27 <i>¿Qué es frontend y backend?</i>	61
Figura 28 <i>Editor vs IDE</i>	64
Figura 29 <i>Visual Basic</i>	65
Figura 30	65
Figura 31.....	69
Figura 32 <i>Enfoques de la Investigación</i>	74
Figura 33 <i>Métodos de Investigación Cualitativa</i>	74
Figura 34.....	76
Figura 35 <i>Investigación Mixta</i>	78
Figura 36 <i>Fuentes de Información</i>	81
Figura 37.....	85
Figura 38 <i>Ejemplo Casos de Uso Registrar Cliente</i>	91
Figura 39 <i>Diagrama de Casos de Usos</i>	93
Figura 40 <i>Gestión de Riesgos</i>	95
Figura 41 <i>Matriz de Riesgos</i>	98
Figura 42 <i>Estudio de Factibilidad</i>	99
Figura 43 <i>Tipos de Factibilidad</i>	104
Figura 44 <i>Sistema Para toma de pedidos en Kanban</i>	106
Figura 45 <i>Tareas Terminadas Sistema Para la toma de pedidos Kanban</i>	107
Figura 46 <i>Diagrama de caso de uso - Login</i>	124
Figura 47 <i>Diagrama de caso de uso - Registro Usuario</i>	125
Figura 48 <i>Diagrama de caso de uso - Registro Producto</i>	127
Figura 49 <i>Diagrama de caso de uso - Registro Categorías</i>	128

Figura 50 <i>Diagrama de caso de uso - Registro Pedidos</i>	130
Figura 51 <i>Diagrama de caso de uso - Registro de Pedidos</i>	131
Figura 52 <i>Diagrama de caso de uso - Registro Detalles Pedidos</i>	133
Figura 53 <i>Diagrama de Flujo Login</i>	134
Figura 54 <i>Diagrama de Flujo Menú</i>	135
Figura 55 <i>Diagrama de Flujo Mantenimiento</i>	135
Figura 56.....	136
Figura 57 <i>Login</i>	139
Figura 58 <i>Pantalla Principal</i>	140
Figura 59 <i>Pantalla Agregar Pedidos</i>	140
Figura 60 <i>Pantalla Actualizar Pedidos</i>	141
Figura 61 <i>Pantalla confirmar pedido a cocina</i>	141
Figura 62 <i>Pantalla Listar Detalles pedidos</i>	142
Figura 63 <i>Pantalla Agregar comentarios de pedidos</i>	143
Figura 63 <i>Pantalla registrar Clientes</i>	144
Figura 64 <i>Pantalla editar Clientes</i>	145
Figura 65 <i>Pantalla Listar Productos</i>	146
Figura 66 <i>Pantalla Agregar Productos</i>	146
Figura 67 <i>Pantalla Actualizar Productos</i>	147
Figura 68.....	148

Dedicatoria y agradecimiento

Le dedico mi trabajo final de graduación a mis papás, quienes siempre me han apoyado para que estudie y llegue a ser una persona destacada. Ellos me mostraron la importancia de hacer las cosas con amor y de la mejor manera, para así obtener grandes recompensas y, sobre todo, satisfacción con uno mismo. Desde la escuela siempre me brindaron mucho apoyo para poder explotar el 100% del potencial y así llegar a grandes logros, destacar y dar un buen rendimiento académico. Me enseñaron a nunca dejar los deberes a última hora y ser muy responsable y honesto.

Como todo en la vida, estudiar es un largo proceso donde aparecen nuevas experiencias y se conocen muchas personas, algunas se quedan, otras se van. Al entrar al colegio donde estudié, conocí a una muchacha con la cual no tenía mucha comunicación porque ambos éramos muy tímidos; al pasar de los años nos fuimos acercando más hasta lograr hacernos novios y ahora ya hasta estamos planeando nuestra boda, por eso también dedico este trabajo a mi futura esposa, la cual nunca me ha dejado solo y cada día me alienta a luchar por los sueños y no dejar que la pereza o el cansancio sean un límite para llegar hasta la meta.

Y más que todo, estoy super agradecido con Dios, él me ha bendecido demasiado con una vida muy linda y la capacidad de lograr grandes cosas al lado de personas muy especiales que me apoyan y guían en cada proceso día con día.

Resumen

Este trabajo de investigación trata sobre un sistema de pedidos para el Restaurante Finca La Bandera; un restaurante pequeño, de tipo familiar, el cual está ubicado en la Zona de Los Santos, Bajo Los Ángeles de León Cortés. Este negocio busca innovarse y poder recibir los pedidos de manera automatizada, para así corregir ciertos errores y deficiencias en el proceso de recibir las órdenes, teniendo un mejor manejo de los platillos que se ofrecen y optimizando la eficiencia en la facturación de cada pedido.

Durante este proceso investigativo, se logró identificar ciertos aspectos de mejora en los procedimientos que conlleva el manejo de un restaurante, tales como rapidez para tomar la orden, *ticket* inmediato en la cocina con la orden o pedido, facturación más precisa sin errores al obtener el total de la cuenta, facturación electrónica, entre otras necesarias para administrar las ventas diarias. Por lo tanto, a fin de optimizar dichas deficiencias, se hace la propuesta de implementar un sistema para el manejo eficiente de pedidos usando Visual Basic.Net y SQL Server, los cuales tendrán ciertos formularios para un acceso seguro con contraseña, menú interactivo para agregar productos, clientes, pedidos, facturación y el debido mantenimiento para hacer cambios que desee el administrador del restaurante, con el fin de tener una mejor y eficaz gestión de pedidos.

Se investigó para contemplar puntos importantes y que el sistema sea del tipo más funcional posible. Dicho sistema podrá buscar clientes en todo el padrón electoral de Costa Rica, con solo digitar cédula o, si no tiene cédula, puede guardar un pasaporte sin ningún problema o número Dimex de ser necesario; se podrá editar o eliminar luego de guardado. Además, guardará el correo y número de celular para enviar la factura correspondiente con el formulario de facturación.

Se espera que dicho sistema sea de gran ayuda para un mejor manejo con más eficiencia y, sobre todo, llevar un orden adecuado en cuanto al tema de los pedidos y las cuentas a cancelar, a fin de evitar algún tipo de confusión o retardo en el pedido. Por este motivo, se busca que sea un sistema actualizado que cumpla con todo lo requerido e investigado y que sea muy confiable para que el negocio opere de manera óptima.

Abstract

This research work is about an ordering system for a small, family-type restaurant, which is located in the Los Santos area, Bajo los Ángeles de León Cortés, called Restaurante Finca, this business seeks to innovate and be able to receive orders in a more automated way, in order to improve certain errors and deficiencies in the process of receiving orders, having better management of the dishes offered and also improving efficiency in the billing of each order.

During this investigative process, it was possible to identify certain aspects of improvement in the procedures involved in managing a restaurant, such as: speed in taking the order, immediate receipt in the kitchen with the order or request, more accurate billing without errors when obtaining the total of the account, electronic billing, among managing daily sales. To improve these deficiencies, the proposal is made to implement a system for better order management using Visual Basic.Net and SQL Server, which will have certain forms for secure access with a password, an interactive menu to add products, customers, orders, billing and due maintenance to make changes that the restaurant administrator desires, to have better and effective order management.

A lot of research was done to consider important points and to make it the most functional type possible. This system will be able to search for clients in the entire electoral registry of Costa Rica, just by typing an ID, or if you do not have an ID, you can save a passport without any problem or Dimex if necessary, it can be edited or deleted after saving, it will also save the email and cell phone number to send the corresponding invoice with the billing form.

Capítulo I. Problema

Planteamiento del problema

Restaurante Finca La Bandera está ubicado en la Zona de Los Santos, en Bajo Los Ángeles de León Cortés. Dicho negocio enfrenta un problema muy serio, al no contar con un adecuado manejo en la toma de pedidos, sobre los platillos que ofrece, así como la lista de clientes o los que quieren obtener su factura electrónica y necesitan un registro; todos estos detalles lo hacen menos eficiente.

Al investigar en este proyecto se presenta una solución práctica que ayudará al orden y manejo de pedidos, disminuyendo pedidos incorrectos, cuentas o totales erróneos, falta de registro, inexistencia de factura electrónica y, sobre todo, lentitud en llevar la orden en papel a la cocina. Toda esta problemática se puede mejorar a través de un sistema de toma de pedidos, para que este pequeño restaurante sea más eficiente y moderno, a fin de estar a más alto nivel y competir en el mercado.

En conclusión, este proyecto tiene como fin ofrecer una solución viable para mejorar la gestión de la toma de pedidos, obtener una eficiencia operativa y un incremento en ventas a largo plazo.

Objetivos

Objetivo general

1. Desarrollar un sistema para la gestión y control de pedidos en Restaurante Finca La Bandera, enfocado en tomar las órdenes y la facturación de estas.

Objetivos específicos

1. Documentar los requisitos funcionales y no funcionales, con el fin de abarcar todas las necesidades del Restaurante Finca La Bandera.

2. Elaborar el diseño del sistema de pedidos, mediante casos de uso, diagrama de clases y diagramas UML.

3. Programar el sistema de pedidos automatizado, con las herramientas de Visual Basic.NET y SQL Server, para garantizar la toma de pedidos y emisión de facturas electrónicas.

4. Verificar pruebas del correcto funcionamiento del sistema, para asegurar que cumple los objetivos planteados y mejora la eficiencia del Restaurante Finca La Bandera.

Justificación

La investigación se basa en mejorar la eficiencia de un restaurante pequeño, el cual enfrenta problemas para realizar los pedidos. La falta del sistema hace que se retarden las órdenes y también que lo ordenado a veces no sea tan exacto con lo que pide el cliente, puede haber errores de escritura o cualquier detalle que confundan al área de cocina y contrarreste eficiencia.

Al analizar esta situación, se busca mejorar los inconvenientes que se dan al trabajar en papel, brindar mejor servicio a los clientes y manejar de manera óptima la información del menú y datos del cliente para la facturación. De esta manera, se incrementa la rapidez en preparar la comida y aumentan las ganancias, ya que se podría trabajar con menos personal porque será más eficaz y directo el proceso.

Dicho sistema será muy útil para tener unas finanzas más exactas y calcular los gastos, ya sea por semana o mensualmente, así se podrá tomar en cuenta el crecimiento del Restaurante Finca La Bandera. Además, es muy útil para el cierre de caja, así se puede llevar más control sobre si el dinero está siendo llevado en totalidad a la caja o hay faltantes por distintos motivos.

Otro aspecto importante es que muchos clientes llegan y, al momento de cancelar, necesitan factura electrónica, lo cual sin este sistema es imposible, pero, una vez se coloque en el Restaurante, se podrá brindar y así los clientes estarán más satisfechos con el servicio, lo cual demuestra que beneficia tanto al restaurante como a las personas que son clientes de este lugar.

Por último, se espera que, con estas mejoras, se pueda trabajar de forma más satisfactoria y no haya fallas en la toma de pedidos ni mucho menos en el proceso de facturación. Por supuesto, será un lugar más moderno y mejorará la reputación, ya que cerca no hay más restaurantes que cuenten con sistema de pedidos.

Antecedentes

Antecedentes internacionales

El primer antecedente corresponde a la tesis presentada por Guilbert, B. H. A. (2023), cuyo objetivo principal fue demostrar la viabilidad para desarrollar e implementar un sistema informático que fuese revolucionario para la gestión administrativa en las ventas y pedidos en el restaurante, tras la necesidad de modernizar y optimizar los procesos en el negocio. Entre sus resultados, está la elaboración de un estudio acerca de la factibilidad de introducir un sistema informático que aborde la problemática actual y que, además, mejore la eficiencia de la gestión administrativa. Se concluyó que la modernización en los procesos internos a través de la tecnología no solo mejoraría la eficiencia operativa, sino también permitiría al restaurante mantener su competitividad y proporcionar una experiencia de alta calidad.

Por otra parte, Gregorio, A. E. V. (2023), en su tesis, definió como objetivo principal implementar la eficiencia operativa, además de un estudio que se enfoca en explorar la diversidad de *softwares* disponibles para la gestión de restaurantes, con especial atención en el *software* destinado a la gestión de reservas. Se lleva a cabo un análisis exhaustivo de un *software* específico desarrollado por un *startup* con sede en Sevilla, llamado Covermanager, con el propósito de comprender su funcionalidad y su impacto en la gestión diaria de un restaurante.

Se concluye que los *softwares* de gestión son herramientas imprescindibles para el sector restaurantero, ya que posibilitan la oferta de un servicio más eficiente, ágil y adaptable a las demandas de los clientes. En un entorno tan competitivo como el del turismo en España, la adopción de estas soluciones tecnológicas se presenta como una

estrategia acertada que puede marcar la diferencia entre el éxito y el fracaso de un restaurante.

Otro antecedente es presentado por Patricio, J. C. R. (2023, 1 julio), en su tesis el objetivo principal se lleva a cabo a partir del desarrollo de una aplicación destinada a la gestión de pedidos, con el fin de mejorar la eficiencia operativa de la cadena de restaurantes *Restaurante Familiar* en Quito. Su principal objetivo es optimizar los procedimientos manuales existentes, lo que permitirá a la sucursal llevar un registro más efectivo y una administración más eficiente de los pedidos realizados por los clientes.

Entre las características clave de esta aplicación, se encuentran la gestión de clientes, la creación y seguimiento de pedidos, así como la integración de sistemas de pago y entrega. Llegó a concluir que el proyecto se logró mediante la combinación de diversas herramientas tecnológicas, siendo destacado el administrador de base de datos PostgreSQL. Esta herramienta posibilita el almacenamiento y gestión eficiente de todos los datos necesarios, así como el establecimiento de relaciones entre las diferentes tablas para garantizar el adecuado funcionamiento de la aplicación.

Finalmente, se considera el antecedente de Wilfredo, M. T. (2023), quien en su tesis se enfoca en la introducción de una aplicación web diseñada para que los usuarios, tanto chefs como meseros, puedan visualizar los pedidos realizados por los comensales en un restaurante junto con su estado actual. El principal propósito de este proyecto es implementar una herramienta web que mejore la gestión de pedidos en dicho establecimiento. A lo largo de los años de actividad comercial del restaurante, se ha observado la persistencia de procesos manuales, entre ellos, la gestión de pedidos. Por lo que se concluye que la incorporación de una aplicación web dedicada a esta tarea se espera

que optimice las operaciones del restaurante y permita un mejor manejo de los pedidos, lo que, a su vez, mejorará el tiempo dedicado a la atención al cliente.

Antecedentes nacionales

En cuanto a los antecedentes nacionales, Arias-García, J. (2013) en su tesis define como propósito exponer las características principales de la aplicación creada, según lo requerido por la empresa. Inicialmente, se partió de la propuesta de la empresa para desarrollar una aplicación web que facilitara la gestión de transacciones, permitiendo a los usuarios de la Zona Norte del país realizar pedidos de comida en línea desde los restaurantes registrados en la plataforma, así como brindar a estos últimos la oportunidad de mostrar su menú. A partir de la identificación del problema, se analizaron los requisitos necesarios para la aplicación y se procedió con el desarrollo de nuevas funcionalidades, además de mejorar una versión previa del sitio web existente. Se concluye que la aplicación fue un éxito para lograr lo solicitado por la compañía.

También se considera el proyecto de Sancho-Corrales, Nancy (TEC), quien indica que la investigación de mercados es esencial para identificar a los consumidores potenciales y garantizar el éxito comercial. En este contexto, la Cooperativa de Productores de Leche Dos Pinos R.L. realizó un estudio en noviembre de 2007 para evaluar el interés de hoteles y restaurantes en adquirir sus productos, específicamente aquellos que no son atendidos directamente por la cooperativa.

Aunque la Cooperativa ya cuenta con la preferencia de muchos consumidores, los resultados del estudio mostraron un bajo conocimiento de las presentaciones a granel en el sector de Food Service, pese al considerable interés por parte de estos clientes. Además, se destacó la importancia de los distribuidores, como Belca, en la comercialización de estos productos. Por ello, se propuso un plan de *trade marketing* para incentivar a los vendedores

de Belca a distribuir un mix de productos a granel, contribuyendo al crecimiento y competitividad de la Cooperativa en este mercado.

Proyecciones

Alcances

1. Mejorar la gestión de la toma de pedidos en el Restaurante Finca La Bandera, para evitar confusión en las cuentas y comunicación con la cocina.
2. Desarrollar un sistema para la gestión de pedidos en el Restaurante Finca La Bandera, el cual cumpla con los requerimientos de seguridad, que se pueda eliminar, editar y agregar información sobre los productos, y la información de los clientes, agregarlos, editarlos y buscarlos en el padrón para obtener la información.
3. Desarrollar un formulario de ingreso con nombre de usuario y contraseña, para que solamente tenga acceso el personal autorizado.
4. Se desarrollarán formularios para categorías, productos, pedidos, clientes, facturación y mantenimiento del sistema.
5. Se implementará un manual de usuario en caso de dudas.
6. No se realizarán estudios de factibilidad comercial, de tiempo, política o legal.

Limitaciones

1. Una de las principales limitaciones es la falta de equipo en el restaurante, se necesita conseguir varios aparatos, como impresoras, una computadora y cableado eléctrico para que todo funcione de la mejor manera.

2. Es la primera vez que el restaurante se ambienta a trabajar de esta forma, siempre se han tomado los pedidos en papel.
3. El proyecto no contempla los gastos asociados a licencias de *software* ni al equipo de *hardware* requerido para su implementación.
4. El sistema solo contará con un usuario tipo administrador que será el encargado de hacer la orden y modificar cualquier elemento, así lo ordena el dueño, ya que él es quien toma la orden y cobra en la caja, solo él tendrá acceso directo.

Capítulo II. Marco teórico

Historia de la programación

La historia de los lenguajes de programación comienza con computadores mecánicos y avanza hacia herramientas modernas de desarrollo de *software*. Inicialmente, eran especializados, con notación matemática y sintaxis compleja. La investigación en teoría de compiladores creó lenguajes de alto nivel, más accesibles. Por ejemplo, Ada Lovelace tradujo el primer programa de computadora de Charles Babbage en la década de 1840. Alonzo Church formalizó el cálculo lambda y la Máquina de Turing simplificó la operación de las máquinas. Los primeros lenguajes describían acciones en máquinas como los telares Jacquard y la máquina de diferencia de Babbage (Historia de los lenguajes de programación, 2023).

Primeros lenguajes de programación

Para 1940, se crearon los primeros ordenadores alimentados por medio de electricidad y también se desarrollaron los primeros lenguajes de programación, los cuales fueron Plankalkül y Short Code. Pero estos lenguajes fueron altamente especializados y se basaron en las notaciones matemáticas y en la sintaxis oscura. Esto también llevó al desarrollo de la programación de alto nivel, debido a lo compleja que es la programación del lenguaje de ensamblador, ya que requiere de un esfuerzo intelectual muy grande.

En 1950, se crearon los primeros lenguajes funcionales de programación. Entre ellos están:

Autocode: automatizó la traducción del código en los primeros lenguajes de alto nivel.

FORTRAN: estableció la programación científica y de alto rendimiento.

LISP: introdujo la programación simbólica y el procesamiento de listas.

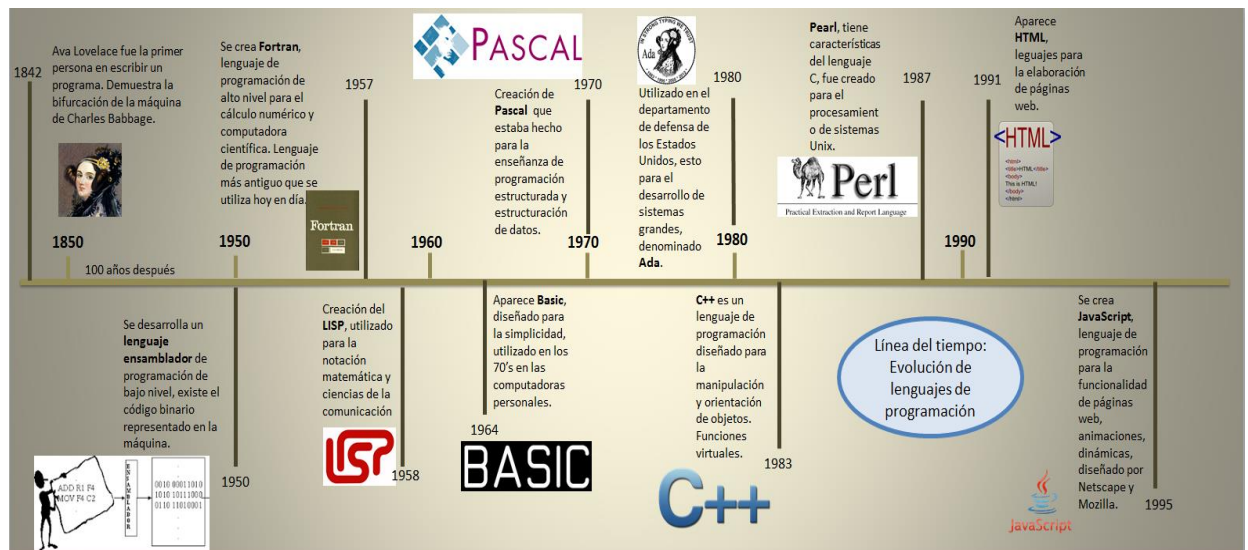
COBOL: facilitó la programación para aplicaciones comerciales y de gestión.

ALGOL: sentó las bases de la programación estructural y la sintaxis de lenguajes modernos.

Estos lenguajes fueron los que introdujeron los conceptos de estructura de bloques anidados, ámbito léxico y notación matemáticamente exacta. En particular, FORTRAN fue el primer lenguaje que tuvo una implementación funcional de alto nivel con propósito general y fue ampliamente utilizado.

Figura 1

Historia de la programación



Fuente: (Evolución de los lenguajes de programación, s.f.).

Para 1960 y 1970, se crearon los paradigmas fundamentales, entre ellos:

Simula: introdujo la programación orientada a objetos.

NORTH: se centró en la programación concatenaria.

C: estableció la programación de sistemas.

Prolog: introdujo la programación lógica.

ML: representó la programación funcional.

La mayoría de los lenguajes modernos son descendientes de estos paradigmas.

Figura 2

Paradigmas de la programación



Fuente: (*¿Qué son los paradigmas de programación?* (s/f))

En 1980, los lenguajes imperativos experimentaron una consolidación relativa con el desarrollo de lenguajes como los siguientes:

ADA: introdujo la programación para sistemas críticos y embebidos con alta facilidad.

ML: permitió el aprendizaje automático para datos.

C++: extendió C con programación orientada a objetos y de alto rendimiento.

Durante este período, se enfatizó la programación para sistemas a gran escala utilizando módulos. Esta tendencia continuó hasta la década de 1990. Además, para 1990, la llegada de la web condujo al desarrollo de lenguajes como los siguientes:

Perl: popularizó la manipulación de texto y el procesamiento de archivos.

Python: destacó su sencillez y legibilidad en programación general.

Java: introdujo el concepto de escribir una vez y ejecutar en cualquier lugar con una máquina virtual.

Ruby: promovió el desarrollo ágil y la programación orientada a objetos con una sintaxis sencilla.

Los lenguajes de programación de secuencias también se hicieron populares, como fue el caso de los siguientes:

JavaScript: se encarga de cargar en el navegador web y que se ejecute en el sistema del cliente.

Los lenguajes de programación han evolucionado para extender, mejorar y desarrollar lenguajes más seguros desde 2000. Algunos de estos lenguajes serían:

Rust: prioriza la seguridad de memoria sin sacrificar el rendimiento.

GO: simplifica el desarrollo concurrente y estable.

Swift: modernizó el desarrollo de iOS y macOS con una sintaxis segura y eficiente.

Y lenguajes específicos para la ciencia de datos y el aprendizaje automático. La evolución de la informática garantiza un futuro con variedad de lenguajes y aplicaciones únicas. Algunos de los más demandados incluyen:

JavaScript: revolucionó el desarrollo web interactuando en el navegador.

Python: destacó por ser simple y versátil en múltiples dominios.

Java: estableció la probabilidad con escribir una vez y ejecutar en cualquier lugar.

C#: potencia el desarrollo en las plataformas de Microsoft con programación orientada a objetos.

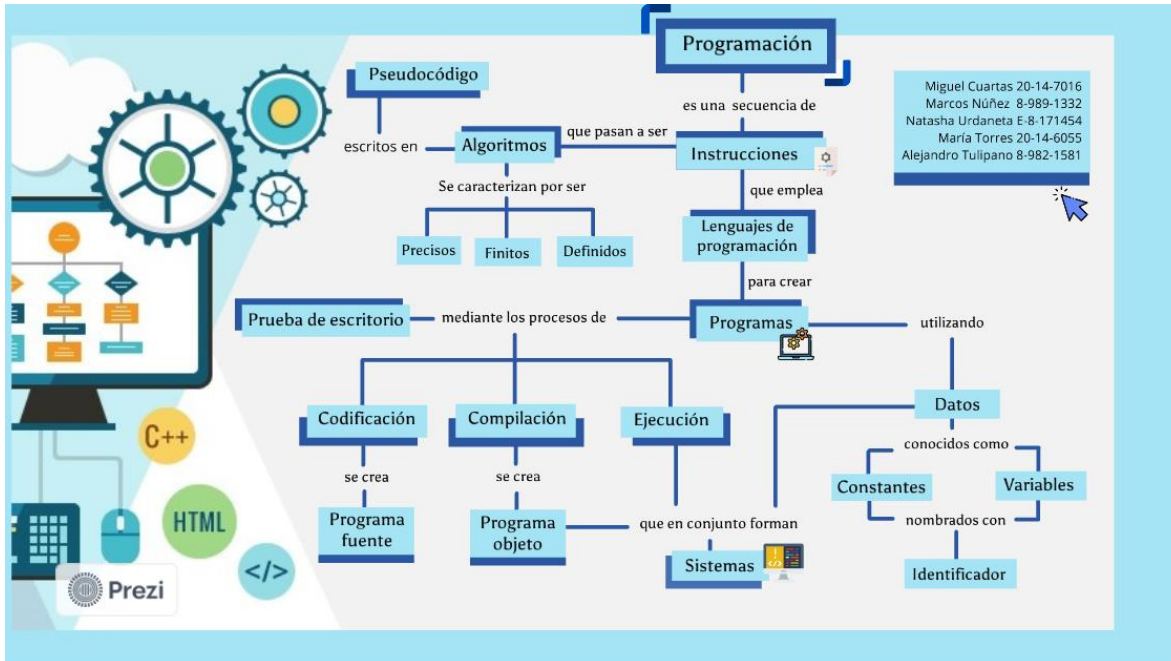
SQL: normalizó el manejo de las bases de datos relacionales por medio de consultas.

Estas se utilizan en una amplia gama de aplicaciones, desde el desarrollo web y móvil hasta la inteligencia artificial y el análisis de datos.

Fundamentos de la programación

Figura 3

Mapa conceptual fundamentos de programación



Fuente: (Introducción a la programación - Mapa conceptual. (s/f))

¿Qué es un lenguaje de programación?

Es una herramienta fundamental que habilita al usuario para dar órdenes mediante un grupo de normas y signos de programación. Los programadores pueden comunicarse con las computadoras usando estos lenguajes para solicitarles la realización de tareas concretas.

Tipos de programación

Según el artículo escrito por Tiffin University (s.f.), existen varios tipos de programación para el desarrollo de *software*, por lo que cada uno cuenta con sus características y enfoques específicos.

Los principales tipos de programación serían los siguientes:

Imperativa: este tipo de programación está diseñado para hacer un seguimiento de un conjunto de instrucciones paso a paso. Los cuales están organizados en forma de una serie de instrucciones que alteran el estado del programa mientras son ejecutados. Las instrucciones son indicaciones para la computadora que le señalan los pasos necesarios para lograr el resultado deseado (Tiffin University, s.f.).

Declarativa: en lugar de especificar cada paso que la computadora debe completar, describe aquí lo que desea lograr. Se proporcionan especificaciones para los resultados deseados y el sistema o lenguaje es responsable de encontrar la mejor manera de lograrlo (Tiffin University, s.f.).

Funcional: este tipo de programación trata a la computación como la evaluación de funciones matemáticas. Ya que, en lugar de cambiar los estados o modificar datos directamente, los programas se construyen por medio de la composición de funciones, promoviendo un enfoque más declarativo y menos centrado en la manipulación directa de los datos (Tiffin University, s.f.).

Lógica: en este modelo, los programas se desarrollan detallando los hechos y reglas, y un motor de inferencia deduce conclusiones a partir de estos elementos. Es especialmente útil en situaciones donde el razonamiento y la manipulación simbólica son clave (Tiffin University, s.f.).

Orientada a eventos: el enfoque principal es en el control de eventos, los cuales son actividades o situaciones que suceden en el sistema, por ejemplo, las interacciones de un usuario. Este estilo de codificación es típico en las aplicaciones que reaccionan a interacciones en directo, tales como juegos, interfaces visuales o sistemas de interacción (Tiffin University, s.f.).

Orientada a objetos: la POO organiza el código en torno a "objetos", los cuales son como unidades independientes que combinan los datos y funciones relacionadas. Este imita conceptos del mundo real, lo que facilita la representación de problemas complejos y promueve la reutilización del código (Tiffin University, s.f.).

Figura 4

Lenguaje de programación



Fuente: *(Los 14 lenguajes de programación más populares, según un estudio entre 100.000 desarrolladores, 2021, mayo 12, Devsoft, s.f.).*

Tipos de lenguajes de programación

En el artículo proporcionado por Latam (2023), se asegura que no hay un lenguaje de programación universal. La elección depende del propósito del proyecto. Además, los lenguajes varían según características y necesidades específicas.

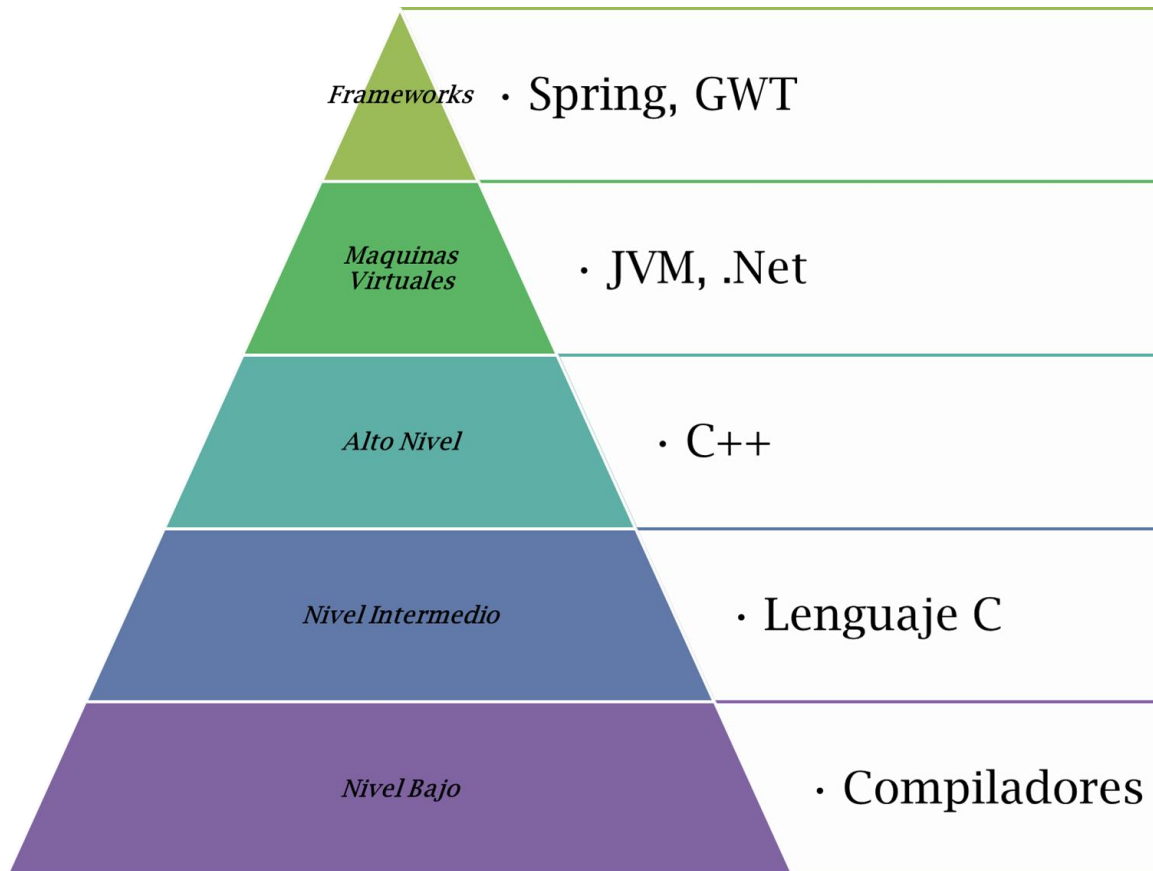
Bajo nivel: los lenguajes de bajo nivel, como ensamblador y máquina, controlan directamente el *hardware* para ejecuciones eficientes. Adecuados para aplicaciones exigentes en rendimiento, requieren conocimiento profundo, siendo menos accesibles y más complicados que los de alto nivel (Maldonado, 2024).

Alto nivel: los lenguajes de programación de alto nivel, como Python, Java y C++, son más fáciles de entender y usar que los de bajo nivel. Facilitan la escritura de código legible y eficaz, separando los detalles técnicos del *hardware*. Estas herramientas

simplifican la programación al ofrecer sintaxis clara, bibliotecas y *frameworks*, siendo ampliamente utilizadas en la industria del *software* (¿Qué es un lenguaje de programación de alto nivel? Y por qué deberías saberlo, 2024).

Figura 5

Lenguajes de alto nivel versus de bajo nivel



Fuente: (*Los lenguajes de programación*, s.f.)

Tipado estático: en ciertos lenguajes como Go, Java, C#, C, C++, TypeScript y Kotlin, las variables deben declararse con un tipo específico que no puede cambiarse más adelante. Por ejemplo, una variable diseñada para almacenar números no puede usarse para texto. Estos lenguajes verifican los tipos durante la compilación para evitar errores en tiempo de ejecución (¿*Qué son los lenguajes tipados y no tipados? (Explicación sencilla)* / EDteam, s.f.).

Figura 6

Lenguajes tipados estático



Fuente: (*Lenguajes tipados versus no tipados*, s.f.).

Tipado dinámico: Python es un ejemplo de lenguaje de programación dinámico que permite que las variables cambien de tipo durante la ejecución del programa. La verificación de tipos en estos idiomas ocurre durante la ejecución, lo que puede llevar a identificar errores relacionados con los tipos después de que el programa esté en marcha.

Otros ejemplos de lenguajes dinámicos incluyen JavaScript, PHP y Ruby (*¿Qué son los lenguajes tipados y no tipados? (Explicación sencilla) | EDteam, s.f.*).

Figura 7

Lenguajes tipados dinámicos

```
In [24]: a = 0
         print(a + 1)
1
```

```
In [25]: a = "Tipado dinámico"
         a = 4 > 3
         print(a)
True
```

```
In [26]: a = 3 - 4j
```

Fuente: (*Tipado dinámico, s.f., Interactivechaos.com*).

Compilado: de acuerdo con la información proporcionada, la compilación convierte el código fuente en código máquina para una ejecución eficiente y rápida.

Ejemplos de lenguajes compilados incluyen C#, C++ y Go, que se convierten a código máquina antes de ejecutarse (Tipos de Lenguajes de Programación | EDteam, s.f.).

Interpretado: según la investigación, los lenguajes como JavaScript, PHP, Python y Ruby se traducen en tiempo real mediante intérpretes, no compiladores. La diferencia principal es que el intérprete traduce durante la ejecución, lo que puede ralentizar el proceso (Tipos de Lenguajes de Programación | EDteam, s.f.).

Intermedio: el artículo redactado por EDteam (s.f.) menciona una etapa adicional en el proceso de desarrollo. Tras compilar el código fuente, se convierte en *Bytecode* para

la JVM, permitiendo ejecutar el código en diferentes sistemas operativos. Ejemplos de estos lenguajes son Java, Kotlin y Scala, que ofrecen flexibilidad y ejecución multiplataforma.

Figura 8

Lenguajes de programación



Fuente: (*Compilados/ Interpretados*, 2017, febrero 15).

Paradigmas de programación

Según Robert C. Martín, los paradigmas son principios que guían la estructura y redacción del código. Los principales paradigmas de programación son el imperativo, que incluye la programación orientada a objetos y el declarativo, que abarca la programación funcional y lógica. Estos influyen en el pensamiento y diseño de los programadores.

Figura 9

¿Qué son los paradigmas de programación?



Fuente: (TOMi.digital - PARADIGMAS DE LA PROGRAMACIÓN 2DO BGU, s.f.).

Orientada a objetos: una técnica que facilita notablemente la simplicidad del código es la creación de estructuras conocidas como clases. Estas clases permiten agrupar funcionalidades bien definidas y delimitadas. Esto no solo mejora la organización, sino que también acelera el desarrollo de los programas, al favorecer la reutilización del código. Lo más relevante es que esta organización más estructurada permite abordar el desarrollo de programas más complejos de manera eficiente.

Figura 10

Programación orientada a objetos



Fuente: (*Desarrollo de Aplicaciones con Software Orientado a Objetos en C*, s.f.).

Funcional: las funciones se consideran valores, permitiendo pasarse como argumentos. Incluyen tres principios importantes: invariabilidad, funciones puras y funciones de grado superior. La invariabilidad implica que los datos no deben modificarse una vez creados. Las funciones puras siempre devuelven el mismo resultado al recibir los mismos argumentos, sin efectos secundarios. Finalmente, las funciones de alto nivel reciben funciones como argumentos.

Figura 11

Programación funcional

Ámbito de la función
Parámetros de entrada
Nombre de la función

```

private Function sumar(numero1 as Integer,
numero2 as Integer) as Integer
Dim suma as Integer= numero1 + numero2
Return suma
End Function

```

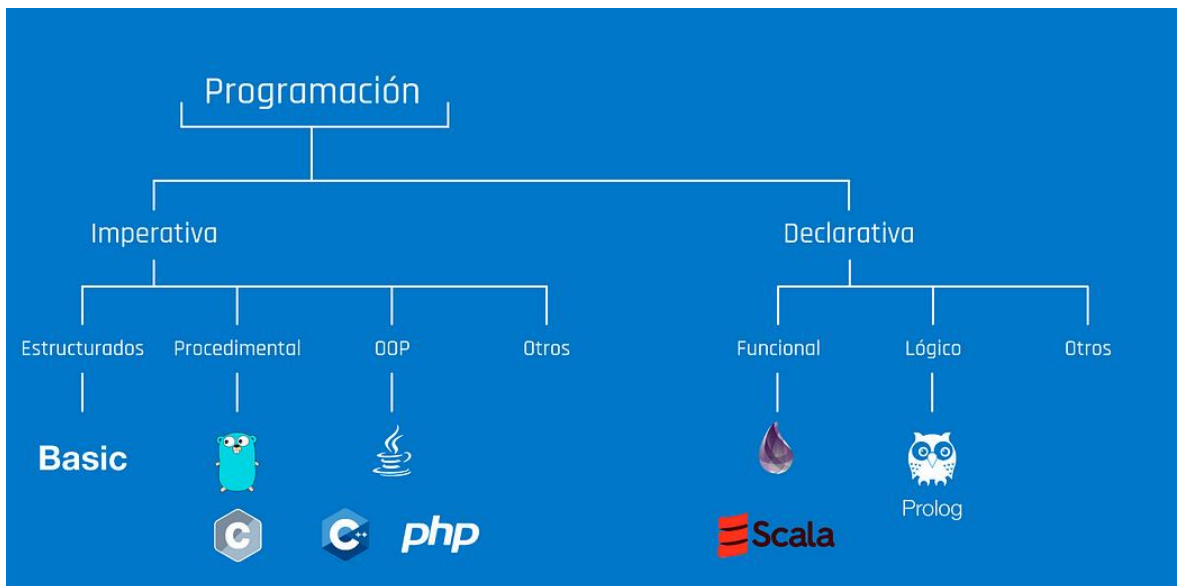
Instrucciones
Valor de retorno

Fuente: (Díaz, L. N. A *Procedimientos y Funciones I*, s.f.).

Imperativa: programación imperativa enfocada en detallar procesos para cambiar el estado del sistema con instrucciones como bucles y condicionales. Se basa en "estado" y "cambio de estado" para controlar el flujo del programa. Aunque brinda control detallado, puede generar código menos modular y complejo de mantener en comparación con enfoques más simples para el manejo del estado.

Figura 12

Programación imperativa (Visual Basic)



Fuente: (LOOPA. (2016, junio 2). *Paradigmas de Programación: Programación Imperativa y Programación Declarativa*).

Declarativa: la programación declarativa prioriza el resultado sobre los pasos, utilizando lógica y reglas para definir el comportamiento del sistema. Además, facilita la creación de aplicaciones complejas al extraer detalles de implementación.

Figura 13

Programación declarativa

```
$nombres = array_values($listaparticipantes);
```

Fuente: (*Programación imperativa: ventajas y desventajas del paradigma* - IONOS, s.f.).

Algoritmos de programación

Los algoritmos son esenciales en *software* para optimizar rendimiento, ejecutar tareas y resolver problemas matemáticos y decisiones en sistemas complejos.

Figura 14

¿Qué es un algoritmo de programación?



Fuente: (¿Cómo aprender a programar desde cero? | EDteam, s.f.).

Sistema de pedidos

Un sistema de pedidos es una herramienta tecnológica diseñada para facilitar y agilizar el proceso de toma de pedidos en un restaurante u otro negocio que involucre la venta de productos. A través de este *software*, los empleados pueden registrar rápidamente las órdenes de los clientes, ya sea mediante una interfaz en línea o a través de dispositivos móviles.

La gestión de pedidos se realiza a través del sistema, permitiendo que cada pedido sea registrado y asignado a la cocina o al área correspondiente para su preparación.

Además, estos sistemas también pueden incluir opciones adicionales como la personalización del pedido por parte del cliente (añadir ingredientes extra, elegir salsas específicas, etc.). (¿Qué es un sistema de pedidos y cómo funciona?, 2024, mayo 6, *Tecseris*).

Figura 15

Sistema de pedidos



Fuente: (*Sistema de pedidos online gratis*, 2024, octubre 17, ApperStreet, s.f.).

¿Qué son los sistemas de pedidos?

Un sistema de pedidos es capaz de ordenar los productos que necesita el cliente y cumple con todo lo necesario para recomodar lo requerido hasta brindar un servicio satisfactorio, tanto así que agiliza el trabajo y hace todos los procesos mucho más rápidos que si fuera en papel donde se toman los pedidos, agiliza el orden y disminuye los márgenes de error en cada pedido realizado.

¿Qué son los sistemas de inventarios?

Los inventarios son esenciales para el control y organización de empresas que manejan mercancías. Un sistema adecuado permite determinar cantidades, cumplir plazos y adaptarse a cambios en la demanda. Por lo que es crucial para el éxito del negocio.

Conociendo los diferentes tipos de inventarios y métodos de gestión, se logra un manejo más efectivo de los recursos (Sistema de inventario: Toma el control de tu almacén, 2023, Colombia, s.f.).

¿Qué son los métodos para el control de inventarios?

Los métodos para el control de inventarios son enfoques utilizados para gestionar de forma eficiente los niveles de inventarios que existen, ayudando a las empresas a nivelar la oferta y la demanda, optimizar recursos y minimizar costos (*Sistema de inventarios: qué es y cuáles son sus tipos, s.f.*).

Estos son algunos métodos:

Método ABC: los artículos de clase A representan el 20% del inventario, pero el 80% del valor. Los de clase B aportan el 15% del valor. Tienen ventas moderadas y requieren atención especial para una gestión financiera rigurosa.

Método PEPS: PEPS se usa en alimentos y textiles para mantener precios competitivos en épocas inflacionarias. Vende mercancía comprada a precios anteriores, evitando la obsolescencia al sacar primero los productos más antiguos del inventario.

Método EOQ: busca minimizar los costos totales del inventario, determinando la cantidad óptima de pedido para equilibrar los costos de almacenamiento, pedidos y demanda esperada.

Este método se basa en tres principales:

- Que la demanda es constante y predecible.
- Que la frecuencia de uso del inventario es estable.
- Que los pedidos se reciban exactamente cuando el inventario se agota.

Método UEPS: el método UEPS (último en entrar, primero en salir) es una estrategia de gestión de inventarios donde se retiran primero los productos más recientes, también conocido como LIFO en inglés. El conteo cíclico implica contar regularmente una parte del inventario total para garantizar que cada segmento sea contado al menos una vez en un período específico (*Método UEPS (last in, first out): usos y ejemplos, s.f.*).

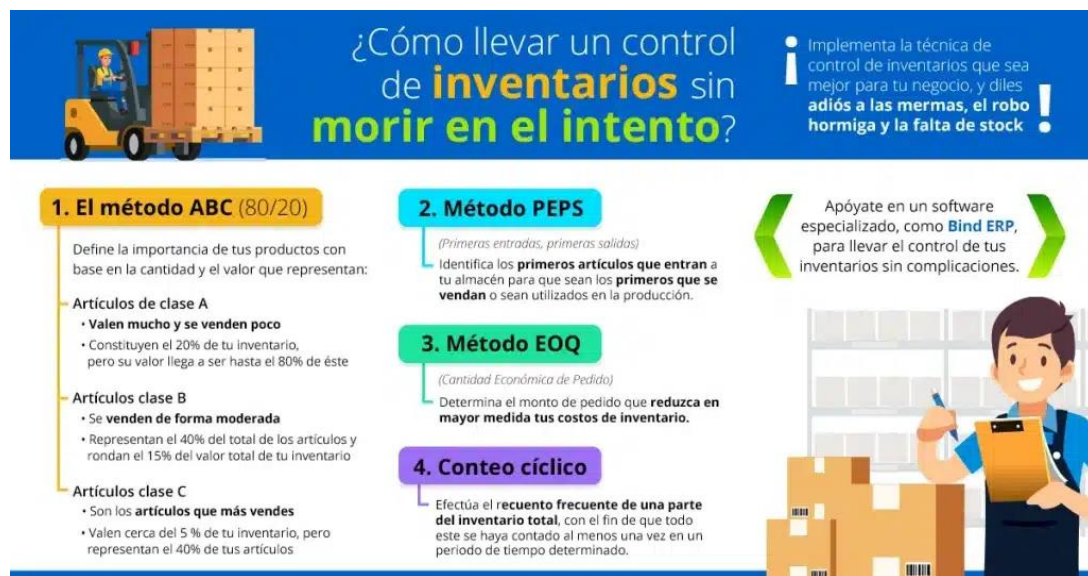
Método de conteo cíclico: el conteo cíclico implica el recuento frecuente de una parte del inventario total, asegurando que cada segmento del inventario sea contado al menos una vez en un período determinado.

Este método se complementa con el método ABC, asignando diferentes frecuencias de conteo a cada categoría. Entre sus utilidades, se encuentra la mejora en la precisión y fiabilidad del control de inventarios, permitiendo identificar y corregir discrepancias sin necesidad de un conteo total.

El conteo cíclico, en combinación con el método EOQ, es útil cuando la demanda es conocida. El utilizar un *software* especializado para el seguimiento regular del inventario puede facilitar este proceso, permitiendo programar ciclos de conteo para otras categorías de productos o ubicaciones, reduciendo la interrupción de las operaciones diarias.

Figura 15

Métodos para el control de inventarios



Fuente: (El concepto del sistema de control de inventario, el gerente profesional y el

trabajador están verificando el suministro de bienes y existencias, la gestión de inventario con la demanda de bienes 12494550 Vector en Vecteezy, s.f.)

Tipos de inventarios

De acuerdo con el artículo *Tipos y clasificación de inventarios existentes* (s.f.): "Los diversos tipos de inventarios permiten a las empresas recolectar y clasificar sus activos y productos utilizando distintos criterios". El monitoreo del inventario puede detectar errores logísticos como diferencias en referencias, defectos en etiquetas o selección incorrecta. Existen diversos tipos de inventarios que varían según el criterio de definición utilizado.

Según el momento: estos tipos de inventario son clave para manejar picos de demanda y garantizar la disponibilidad de productos para satisfacer las necesidades del mercado.

Inventario inicial: el inventario inicial, al inicio del ejercicio contable, muestra la cantidad de mercancías en cada departamento. Debe ser detallado para evitar la escasez en un área y el exceso en otra.

Inventario final: entre los tipos de inventarios de productos, el inventario final es la cantidad y el valor total de los bienes al final del año, determinado al cierre del ejercicio contable para productos específicos.

Según la periodicidad: el tipo de inventario según la frecuencia con la que se realiza. Se utiliza para verificar y actualizar el registro de *stock* de manera programada, ya sea a través de sistemas de gestión o de forma manual por parte de los empleados. Estos tipos de inventario incluyen:

Inventario intermitente: se realiza para determinar la cantidad de mercancías, materiales o materia prima disponibles. En este tipo de inventario, la fecha para la próxima

revisión es a largo plazo y las verificaciones pueden hacerse según sea necesario sin incorporarlas al inventario permanente.

Inventario perpetuo: es el sistema de inventarios más utilizado, registra las existencias de manera continua y detallada, permitiendo conocer en todo momento el monto vendido y los inventarios finales sin conteos manuales. Esto ahorra tiempo y acelera la elaboración de los estados financieros mensuales.

Inventario periódico: es un tipo de inventario cíclico programado regularmente, como cada mes, semestre o año, para controlar y planificar el *stock* de productos, aprovisionamiento y ventas. Permite conocer cantidades exactas y optimizar la gestión de inventario.

Según la fase de producción: la siguiente clasificación agrupa el inventario según la fase de producción en la que se encuentre:

Inventario materia prima. Este tipo de inventario de materias primas es el *stock* de insumos esenciales para la fabricación de productos.

Inventario de productos en etapa de fabricación: se monitorea la mercancía según su etapa de fabricación para establecer fechas de entrega. Se revisa el inventario para garantizar una producción eficiente.

Inventario de productos terminados: cuando las mercancías están terminadas, se incorporan al inventario de productos terminados para su distribución y comercialización.

Según la función por determinar: la siguiente clasificación se basa en la función del inventario para lograr una gestión óptima.

Inventario en tránsito: son bienes pagados por un proveedor en algún punto de la cadena de suministro, sin estar en almacenes. Probablemente en camino para entrega.

Inventario de ciclo: se utiliza para asegurar que la producción sea equivalente a la cantidad de productos vendidos o demandados por los clientes. Su función es evitar la sobreproducción y el exceso de *stock*.

Inventario de seguridad: se utiliza para prevenir emergencias y establecer un plan de contingencia que permita actuar rápidamente ante posibles incidentes.

Inventario de desacoplamiento: el inventario de desacoplamiento coordina procesos de inventariado para evitar solapamientos, basándose en función y ubicación en la cadena de suministro.

Según logística: según el plan logístico, se lleva a cabo para optimizar la cadena de suministro y se clasifica en diversas categorías.

Inventario de existencias para especulación: es un sistema de control que permite conocer la cantidad de productos y materias primas disponibles para determinar si es viable aumentar la producción. Este inventario se lleva a cabo para anticiparse a momentos de alta demanda.

Inventario de existencias obsoletas: permite controlar las mercancías, materias primas y productos terminados, con el objetivo de reducir las pérdidas en productos obsoletos, dañados o caducados.

Inventario de previsión o anticipación: se realiza para cubrir necesidades futuras, reduciendo peligros y asegurando el *stock* antes de periodos de alta demanda, promociones o cierres de plantas.

Inventario en consignación: la mercancía en el almacén está disponible para la venta, pero sigue siendo propiedad del proveedor, los registros controlan las fechas de ingreso y ventas.

Inventario de lote: este tipo de inventario se basa en pedir mercancías en tamaño de lote o producirlas en grandes cantidades, ya que resulta más económico anticipar el *stock* y la demanda.

Existen otras maneras de clasificar diferentes tipos de inventarios, tales como el inventario físico, de mínimo y máximo, de agregados, la línea y el de cuarentena.

Sistemas de información

Es una estructura organizada de personas, procedimientos y herramientas destinadas a la gestión eficiente de datos e información. Su objetivo es asegurar que la recuperación y el procesamiento de la información se realicen de manera ágil y sencilla.

De acuerdo con Pursell (2023): “Un sistema de información es el conjunto de técnicas, herramientas y agentes involucrados en la administración y uso de datos para la obtención de objetivos empresariales”. Estos sistemas ayudan con la gestión de información que produce y utiliza una organización para el mejoramiento de procesos y operaciones.

Los componentes de un sistema de información facilitan varios procesos clave:

1. La entrada de datos.
2. La administración.
3. El procesamiento.
4. El almacenamiento.
5. La distribución de la información a quienes necesitan acceder a ella.

Figura 16

Sistemas de información

1. Hardware.
 2. Software.
 3. Bases de datos.
 4. Telecomunicaciones.
 5. Personal.
- Procedimientos con los que se procesan las transacciones:
1. La recopilación.
 2. La edición.
 3. La corrección.
 4. La manipulación.
 5. El almacenamiento de datos.
 6. La producción de documentos.

Sistemas de información gerencial

Conocidos también como sistemas de información de gestión, apoyan la toma de decisiones estructuradas o semiestructuradas de los mandos intermedios. Su principal labor es la de sintetizar la información de rutina de una empresa, para asegurar el buen funcionamiento de los procesos; por ejemplo, los informes bimestrales, mensuales o semanales.

Existe un sistema de información gerencial para diferentes departamentos de una compañía. Entre los más utilizados se encuentran:

1. Marketing: producto, precio, competencia.
2. Producción: rendimiento, inventarios, proveedores.
3. Finanzas: inversiones, mercados, cotizaciones.
4. Recursos humanos: incentivos, reclutamiento, desempeño.

Componentes del sistema:

1. Hardware.
2. Software.
3. Datos.
4. Personas a gerenciar.
5. Métodos: las prácticas operativas, modelos e instrucciones que están en los manuales de la empresa.

Sistemas de control de procesos de negocio

Este tipo de sistemas es utilizado para monitorear y controlar procesos industriales o físicos. Se utilizan sobre todo en la industria petrolera, siderúrgica o de generación de energía. Incluyen varios equipos, programas especializados y procedimientos de operación.

Con un BPM, las empresas adoptan una serie de pasos o acciones para modificar su forma de trabajar con el fin de mejorar sus procesos y facilitar la colaboración con un enfoque centrado en el cliente.

Componentes del sistema:

1. Entorno gráfico de diseño: son los que modelan la secuencia de actividades del proceso o proyecto.
2. Middleware: son las que facilitan la comunicación del motor con las aplicaciones que se usan en el BPM.
3. Metadatos: rol, usuario, sistema, servicio.
4. Interfaz de usuario.
5. Monitorización: son para darle seguimiento puntual a procesos y actividades.
6. Reglas de negocio.

7. Simulación: las que permiten la ejecución de los procesos en un ambiente controlado.
8. Automatización.
9. Ejecución en la nube.
10. Estándares: se utilizan para que todos los involucrados conozcan y manejen los mismos códigos, gráficos y actividades.

Sistemas de información de marketing

Los SIM son un conjunto de relaciones estructuradas entre personal, máquinas y procedimientos para generar un mejor flujo de información. Estos datos provienen tanto de fuentes internas como externas y sirven para tomar decisiones más inteligentes relevantes al área de *marketing*.

Con su uso pueden determinar principalmente el grado de fiabilidad que tiene una información, el ritmo actualizado de información y un grado de detalle más específico. Por medio de este proceso, es posible hacer las modificaciones y adaptaciones pertinentes en diferentes situaciones de la empresa.

Componentes del sistema:

1. Datos internos: estas son métricas de actividades y rendimiento interno.
2. Inteligencia de marketing: es información del mercado e industria de fuentes confiables.
3. Investigación de mercados: son los datos cualitativos y cuantitativos que se recogen por medio de herramientas y actividades realizadas directamente por la empresa.
4. Soporte a las decisiones de *marketing*: en este se incluyen también las plataformas de análisis de datos y reportes.

Sistemas de colaboración empresarial

Los ERP son los más utilizados en el mercado. Tienen la función de ayudar a los directivos de una empresa para así controlar el flujo de información y proporcionar una vista integral de cada una de las áreas que conforman la compañía, para que, de esta forma, se pueda mejorar la gestión y toma de decisiones.

Su tecnología está encargada de registrar e integrar los procesos de negocio y ayuda a las empresas a optimizar sus procedimientos para tener una mayor comprensión de esta a través de la automatización. Con este tipo de programas, los líderes de los negocios pueden actuar con más rapidez con la seguridad de que estarán basando sus decisiones en información confiable y en tiempo real.

Componentes del sistema:

1. Datos.
2. *Hardware*.
3. *Software*.
4. Procedimientos.

Sistema de soporte de toma de decisiones

Conocido por sus siglas en inglés como DSS, es un sistema basado en ordenadores que es utilizado regularmente por los gerentes para tomar una decisión, con el fin de resolver un problema en la empresa. Permite la formulación de cálculos, comparar opciones y predecir escenarios para saber cuál es la mejor opción por tomar. Además, está diseñado para ayudar en situaciones complejas.

El *software* que se utiliza tiene la función de recolectar información procedente de documentos, personal interno, modelos comerciales, experiencias o análisis externos. Con

estos datos, se identifican los problemas o desafíos. De igual forma, favorecen el incremento de la efectividad de manera oportuna.

1. Componentes del sistema:
2. Gestor de datos: es la base de conocimiento.
3. Gestor de modelos.
4. Motor de conocimiento.
5. Interfaz de usuario.
6. Usuarios: cada uno tiene sus roles; recolectores de datos, asesores y tomadores de decisiones.
7. Contexto de decisión: establecido y bien definido.

Sistemas de información ejecutiva

Los EIS son los encargados de brindar acceso rápido a la información interna y externa relevante de una empresa. Estos suelen visualizarse en formatos gráficos para la presentación de datos básicos, pero detallados, con la finalidad de que los gerentes o ejecutivos puedan leer la información con facilidad.

Uno de los caracteres más sobresalientes de este tipo de sistemas es que les dan a los que toman las decisiones, los indicadores de negocio que afectan al instante. También, permiten un análisis detallado de los que no están cumpliendo con las expectativas para determinar el plan de acción más conveniente.

1. Componentes del sistema:
2. Interfaz gráfica.
3. Indicadores de negocio: estos son definidos por la empresa.
4. Integración de bases de datos: pueden ser con *datamart* o *data warehouse*.
5. Usuarios.

6. Acceso en la nube.

El ciclo de vida de un sistema de información

El ciclo de vida de un sistema de información garantiza la satisfacción del cliente, controla el presupuesto y los plazos. Incluye fases que van desde el nacimiento hasta la expiración del sistema y consiste en las siguientes fases:

Planificación

En esta etapa inicial, el analista identifica las necesidades del usuario y realiza la planificación para visualizar el proceso y definir cómo lograrlo. No hay sistema desarrollado todavía. El analista determina los datos necesarios para alcanzar el objetivo, así como los recursos disponibles y requeridos para el proyecto.

Análisis de requerimientos

La etapa de desarrollo de un sistema de información inicia luego de conocer todos los requerimientos necesarios. El objetivo es crear un diseño base rentable y eficiente desde el principio. Durante esta fase, el analista examina los requerimientos funcionales del sistema, a través de entrevistas o encuestas a los usuarios actuales. Es crucial determinar la viabilidad del plan de acción, considerando aspectos financieros, tecnológicos y organizativos. En caso necesario, se pueden realizar modificaciones para mejorar el proyecto.

Diseño

El analista ha definido los requisitos del nuevo sistema de información y comienza a trabajar en planos y diseño, integrando información clave para la empresa. Se centra en el diseño de *software* e infraestructura, incluyendo interfaces de usuario, bases de datos y seguridad.

Desarrollo

Tras finalizar el sistema, se activa un periodo de prueba para ajustar detalles y satisfacer expectativas. Este proceso implica una colaboración completa entre desarrolladores y analistas para integrar tecnología e información adecuada.

Prueba

En esta fase, se verifica el funcionamiento real del nuevo sistema de información, eliminando detalles no deseados y realizando los ajustes necesarios. Requiere inspección minuciosa para detectar y corregir problemas, garantizando el alineamiento correcto.

Integración y ejecución

Se lanza el sistema por primera vez, requiriendo seguimiento para corregir posibles imperfecciones menores. Tras solucionar los últimos detalles, estará listo para los usuarios finales.

Operación y mantenimiento

Como en todo proceso, es crucial hacer un seguimiento y mantenimiento regulares para garantizar el correcto funcionamiento y óptimas condiciones del sistema.

Figura 17

Ciclo de vida de un sistema



Fuente: (*Ciclo de Vida de un Sistema de Información*, s.f.).

Requerimientos de sistema

Al iniciar un proyecto de *software*, es fundamental comprender las necesidades del cliente mediante la creación de un documento detallado. Este documento facilita la planificación, desarrollo y pruebas del sistema, aumentando las posibilidades de éxito al garantizar que el proyecto cumpla con las expectativas iniciales del cliente.

Requerimientos funcionales

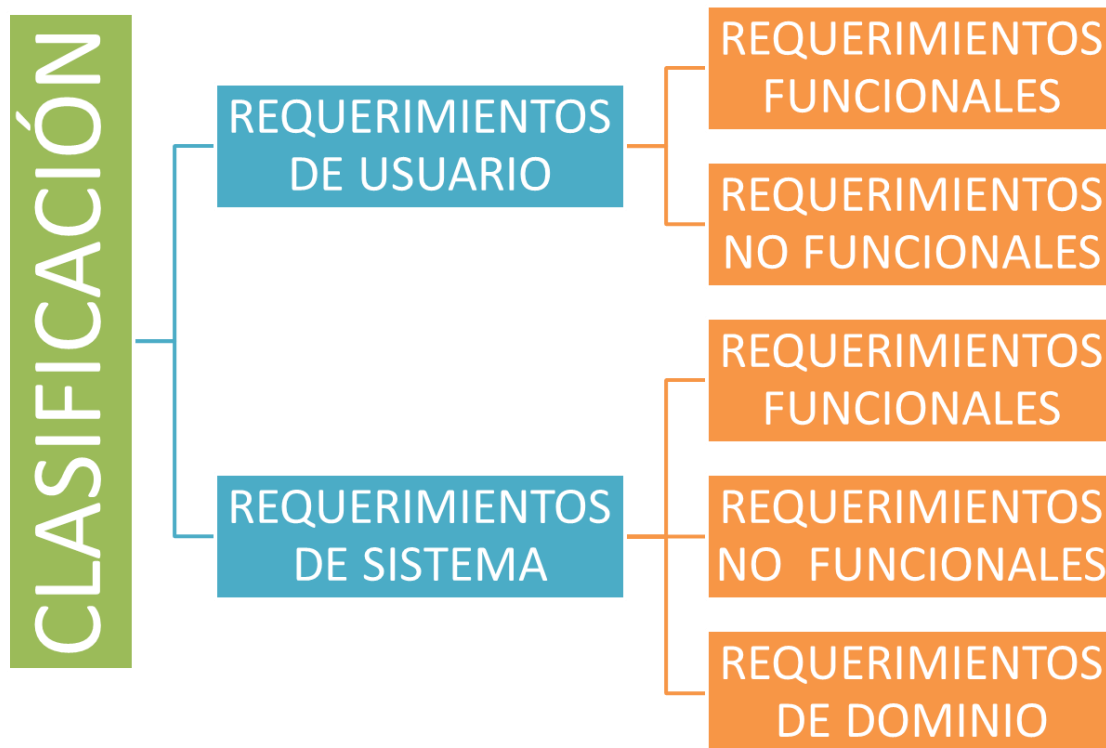
Los requisitos de un sistema especifican sus funcionalidades y comportamientos. Indican las tareas generales, procesos específicos y la interacción usuario-sistema. Además, permiten al *software* responder a diferentes entradas para satisfacer las necesidades del usuario.

Requerimientos no funcionales

Abordan aspectos que no están directamente relacionados con las funcionalidades específicas del sistema, pero que son igualmente importantes para su operatividad y calidad. Estos incluyen criterios como el rendimiento, la seguridad, la usabilidad y la compatibilidad del sistema. Los requerimientos no funcionales definen cómo debe comportarse el sistema en diversas condiciones y establecen los estándares necesarios para evaluar su eficiencia y eficacia.

Figura 18

Requerimientos del sistema



Fuente: (CLASIFICACIÓN Y TIPOS DE REQUERIMIENTOS, 2014, agosto 26, ADMINISTRACIÓN DE REQUERIMIENTOS).

Diseño integral de sistemas

El objetivo del diseño integral es crear soluciones que sean coherentes y efectivas para todos los aspectos, garantizando así que el producto o sistema cumpla con las expectativas y necesidades de los usuarios (Diseño Integral - FasterCapital, s.f.).

Arquitectura

Es la estructura base de un sistema de *software*. Además, define cómo se organizan e interactúan sus componentes. Este concepto es muy importante para garantizar que el sistema sea escalable, flexible y adaptable a las necesidades de su proyecto. Los tipos de arquitectura más conocidos son:

Monolítica: todos los componentes del sistema están integrados en un solo bloque de código. Este código es simple de implementar, pero puede volverse complejo y difícil de mantener a medida que el sistema crece.

Capas: organización del sistema en distintos niveles de importancia, cada capa cumple roles específicos y solo interactúa con los estratos cercanos. Este enfoque se utiliza combinado para dividir responsabilidades y lograr escalabilidad.

Microservicios: esta arquitectura tiene una tendencia a dividir el sistema en múltiples pequeños y completamente independientes. Cada uno ejecuta una tarea determinada, específica. Un diseño como este aporta una gran flexibilidad por la escalabilidad, pero tiene una dificultad de administración.

Eventos: este método se fundamenta en la interacción entre componentes mediante eventos, lo que posibilita un gran nivel de separación entre los diversos módulos del sistema.

Cliente-Servidor: el sistema se separa en dos partes: el cliente pide servicios y el servidor los brinda. Este tipo de sistema es muy frecuente en aplicaciones web y sistemas distribuidos.

Componentes: así, esta armoniosa estructura, que favorece la utilización de herramientas modulares, posibilita la creación de sistemas complejos mediante el uso de bloques ya existentes.

Orientada a Servicios (SOA): el sistema ofrece servicios reutilizables para crear aplicaciones grandes. Es ideal para entornos corporativos y sistemas complejos. La arquitectura adecuada depende de las especificaciones y requisitos técnicos del proyecto (Arquitectura de Software: ¿Qué Es y Qué Tipos Hay, Gluo, 2024, s.f.).

Estructura de datos

Existen varios tipos de arquitecturas de datos que se utilizan para gestionar y organizar la información dentro de una entidad, cada uno con características y aplicaciones específicas:

Arquitectura de datos centralizada: en este modelo, todos los datos se almacenan en un único repositorio central, lo que facilita su administración y control. Sin embargo, puede convertirse en un punto único de falla si no se gestiona adecuadamente.

Arquitectura de datos descentralizada: los datos se distribuyen entre varios sistemas o ubicaciones, lo que mejora la disponibilidad y reduce los riesgos de pérdida de datos, pero puede generar desafíos en la integración y consistencia de la información.

Arquitectura de datos federada: este enfoque combina aspectos de las arquitecturas centralizadas y descentralizadas, permitiendo que los datos se almacenen en múltiples ubicaciones, pero manteniendo una vista unificada y coherente de los mismos.

Arquitectura de datos orientada a servicios (SOA): en este modelo, los datos se gestionan como servicios independientes que pueden ser reutilizados y combinados para

formar aplicaciones más grandes y complejas. Es especialmente útil en entornos empresariales donde se requieren altos niveles de flexibilidad y escalabilidad.

Arquitectura de datos en la nube: la arquitectura en la nube gestiona eficientemente grandes volúmenes de datos con ventajas en escalabilidad y costos, sin embargo, enfrenta desafíos en seguridad y cumplimiento normativo. También es crucial para la integración y gobernanza de datos, alineando la información con los objetivos del negocio y los requisitos de seguridad (Arquitectura de Datos: ¿qué Es? Importancia, Tipos y Usos, Astera, 2024, s.f.).

Interfaz

En el desarrollo de *software*, las interfaces gráficas de usuario (GUI) facilitan la interacción a través de elementos visuales como botones y ventanas en lugar de comandos de texto. El diseño de una GUI es crucial para una experiencia satisfactoria, considerando aspectos como la disposición visual, coherencia de elementos y claridad en la presentación de información para garantizar eficiencia y fluidez en la interacción (*¿Qué es una interfaz?* / *KeepCoding Bootcamps*, s.f.) (*¿Qué Es Interfaz Gráfica de Usuario y Para Que Sirve?* NeoAttack, 2024, s.f.).

Componentes

Los componentes pueden ser actualizados o reemplazados de forma independiente, mejorando la flexibilidad y adaptabilidad del sistema. La arquitectura de componentes es útil en sistemas complejos con alta cohesión y bajo acoplamiento entre partes (Arquitectura de Componentes - DEV Community, Vanessa Marely. (2024), s.f.).

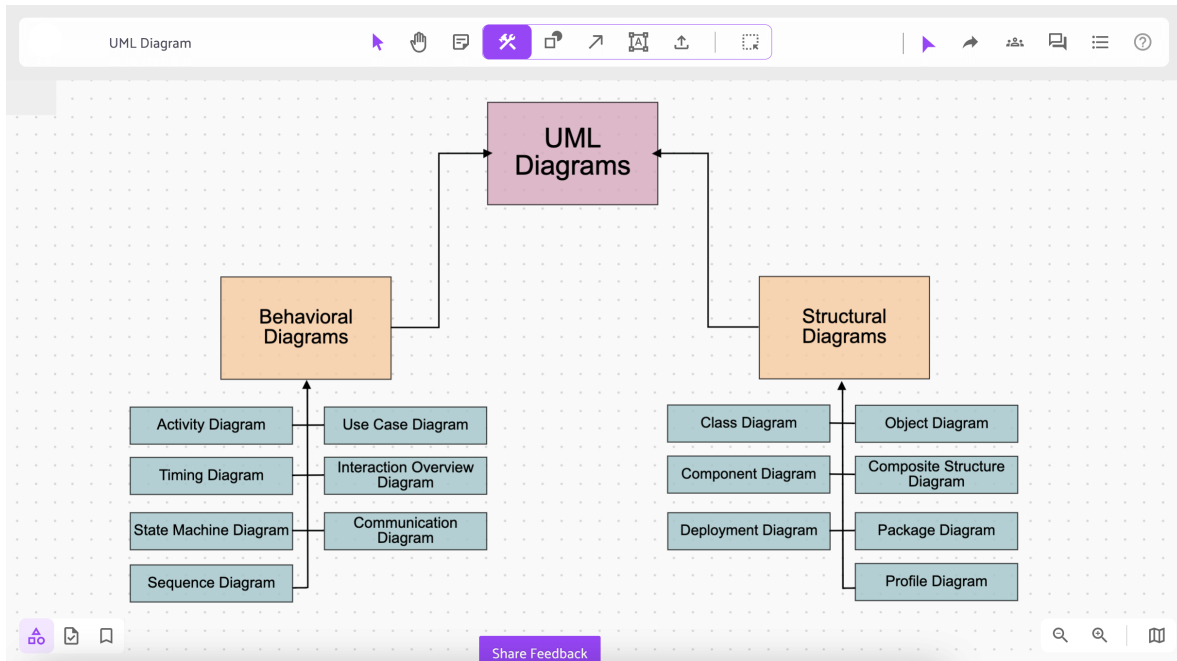
Diagrama UML

Un diagrama visual muestra la estructura y organización de los componentes en un sistema de *software*, destacando cómo se conectan y colaboran módulos y bibliotecas.

Facilita la planificación y comprensión de la arquitectura del *software*, al resaltar interfaces y dependencias, identificando mejoras y asegurando una integración efectiva (Diagrama de Componentes UML: Qué Es y Cómo Hacerlo | Miro, 2024, s.f.).

Figura 19

Diagrama UML



Fuente: (Alam, M. (2022, marzo 14). *¿Qué es un diagrama UML? Definición, casos de uso y cómo hacer.* IdeaScale, s.f.).

Casos de uso

Son una técnica de modelado que describe cómo los usuarios interactúan con un sistema para alcanzar un objetivo específico. Cada caso de uso define un conjunto de acciones que el sistema debe realizar para satisfacer las necesidades de los usuarios o actores en un escenario particular (*Definición de casos de uso - Documentación de IBM*, s.f.a).

Figura 20

Casos de uso



Fuente: (*12 conceptos básicos de la base de datos*, 2020, noviembre 13, Dream Technology).

¿Qué son las bases de código abierto (Open Source)?

Las bases de datos de código abierto proporcionan acceso gratuito a su código fuente, lo que permite a los usuarios ver, modificar y redistribuir el *software*. Además, la comunidad de desarrolladores global asegura la actualización y el cumplimiento de altos estándares de seguridad en estas bases de datos.

MySQL: es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que emplea el lenguaje SQL para administrar y recuperar datos de manera eficiente. Este sistema es ampliamente adoptado en el desarrollo de aplicaciones web, debido a su capacidad para manejar grandes volúmenes de información y su compatibilidad con diversas plataformas (*¿Qué es MySQL? Explicación y características / Blog de Arsys, s.f.*).

Figura 22

Qué es MySQL



Fuente: (Lloves, F., 2021, octubre 26, *Funciones en MySQL*. Somos PNT - Desarrollamos Software, s.f.).

PostgreSQL: también conocido como “Postgres”, es un sistema de gestión de bases de datos objeto-relacional que cumple con los estándares de SQL. Por otro lado, MySQL se destaca por su facilidad de uso, siendo ideal para proyectos más pequeños. Sin embargo, para *startups* con previsiones de crecimiento en la carga de datos, la migración de MySQL a PostgreSQL puede ser la mejor opción (*Las mejores bases de datos open source* - Codemotion, s.f.).

Redis: destaca por su capacidad para realizar operaciones de lectura y escritura a altas velocidades, lo cual es ideal para aplicaciones que requieren un rápido acceso a los datos. Redis es compatible con más de 50 lenguajes de programación y ofrece una API de módulos que permite a los desarrolladores crear extensiones personalizadas. En estos casos, es aconsejable combinar Redis con otras bases de datos para manejar cargas de trabajo adicionales y proporcionar una solución más robusta (*Las mejores bases de datos open source - Codemotion, s.f.*).

MongoDB: es una base de datos No-SQL que se caracteriza por su diseño basado en documentos, lo que contrasta con el enfoque tradicional de almacenamiento en tablas utilizado en bases de datos relacionales. Esta base de datos emplea el MongoDB Query Language (MQL) para operar con los datos, lo que permite una mayor flexibilidad en el almacenamiento y la consulta de información (*Las mejores bases de datos open source - Codemotion, s.f.*).

Neo4j: es una base de datos de grafos NoSQL diseñada para administrar, consultar y almacenar datos en forma de grafos en tiempo real. Su estructura basada en nodos y relaciones permite una gestión eficiente de componentes de datos, facilitando el análisis y la navegación de estos (*Las mejores bases de datos open source - Codemotion, s.f.*).

Figura 23

Bases de datos open source



Fuente: (*Base de datos Open Source*. (2022, marzo 2). ITD Consulting, s.f.).

Metodologías de desarrollo

Estas metodologías se dividen en varios tipos, entre los cuales destacan las metodologías tradicionales, como el modelo en cascada, y las metodologías ágiles. El modelo en cascada es secuencial, con fases bien definidas, lo que facilita la gestión de proyectos con requisitos estables.

Metodologías tradicionales

Cascada

Es un enfoque tradicional y lineal en el desarrollo de *software*, donde cada fase del proyecto se completa de manera secuencial antes de pasar a la siguiente (*Metodología en cascada: Ventajas e inconvenientes* / SafetyCulture, s.f.).

Incremental

Es una metodología en la que el sistema se construye en partes o incrementos sucesivos, cada uno de los cuales proporciona una porción funcional del producto final. La

entrega continua de incrementos funcionales también permite una validación temprana del sistema y asegura que los componentes se integren de manera efectiva a lo largo del desarrollo (*¿Qué es el desarrollo incremental? | phoenixNAP Glosario de TI, s.f.*).

Prototipo

Este enfoque permite a los desarrolladores y a las partes interesadas explorar conceptos, identificar problemas y hacer ajustes en etapas tempranas del proyecto. Al utilizar prototipos, los equipos pueden obtener retroalimentación continua, refinando y ajustando el diseño según las necesidades del usuario (*Prototipo – el papel del prototipado en el proceso de desarrollo de productos - Copymate, s.f.*).

Metodologías ágiles

Scrum

A través de esta metodología, las personas pueden abordar problemas complejos a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. Además, Scrum es una metodología que ayuda a los equipos a colaborar y realizar un trabajo de alto impacto. Por otra parte, en Scrum se trabaja con equipos pequeños multidisciplinares en ciclos iterativos centrados en el cliente y se crea un producto de forma incremental. (*Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2024] • Asana, s.f.*).

Kanban

Es una metodología ágil que utiliza un sistema de gestión de tareas. Su propósito principal es supervisar y optimizar el flujo de trabajo desde el inicio hasta el final de las tareas, asegurando un proceso continuo y eficiente. Kanban, una palabra japonesa que combina, refleja la esencia del método: el uso de tarjetas visuales para representar tareas y

su estado dentro de un proyecto (*Metodología Kanban: en qué consiste y cómo utilizarla* | APD, s.f.).

Programación extrema (Extreme programming, XP)

Es una metodología ágil de gestión de proyectos que se concentra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. Como las otras metodologías ágiles, la programación extrema es un método de desarrollo de *software* dividido en *sprints* de trabajo. Del mismo modo que otros métodos ágiles, este diseño permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real (*¿Qué es la programación extrema (XP)? [2024]* • Asana, s.f.).

Figura 24

Metodologías ágiles y tradicional.

Metodologías de desarrollo de software



Fuente: (<https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>)

¿Qué es el desarrollo web?

El desarrollo web es un proceso que abarca la creación, el diseño y el mantenimiento de un sitio web. Dado que el desarrollo web abarca muchos componentes, se refiere principalmente a la programación de sitios web. Por supuesto, la alineación social de las páginas no solo implica esto, sino que también abarca la programación, la seguridad cibernética y otras cuestiones de mantenimiento. Un desarrollador web, por lo tanto, combina habilidades técnicas y artísticas para construir las bases en las que se centran las actividades informáticas modernas. Algunos ejemplos comunes de estos sitios son tiendas en línea y servidores de redes sociales que dependen de miles de líneas de código.

¿Qué hace un desarrollador web?

El desarrollador web, también conocido como ingeniero o programador web, es un profesional multifuncional especializado en sitio y aplicación web, también es responsable del aspecto de este último. En particular, este especialista se ocupa del diseño y desarrollo de soluciones de *software* basadas en web y planea la arquitectura, desarrollo, mantenimiento y optimización posterior del sitio.

Las responsabilidades más usuales serían:

1. Reunirse con los clientes para discutir el diseño del proyecto y las necesidades técnicas.
2. Desarrollar y probar aplicaciones web, en general, escribir el código web.
3. Trabajar con un grupo mayor integrando elementos adicionales, como gráficos, audio y video, y desarrollar una gama más amplia de funcionalidades, dependiendo del cliente.
4. Supervisar la velocidad y capacidad del sitio.

Figura 25

Qué hace un desarrollador web



Fuente: (¿Qué es un desarrollador web y cuáles son sus principales funciones?, 2023, enero 13, Hackio.com; Hackio by thePower).

Tipos de desarrollo web

Front-end (interfaz)

El desarrollo *front-end* se centra en el aspecto y la experiencia que un usuario puede experimentar en un sitio web. De esta manera, los desarrolladores *front-end* deben mantenerse actualizados sobre las tendencias en diseño web y desarrollo, para asegurarse de que los sitios web estén optimizados para los consumidores y la búsqueda, así como se apegan a las mejores prácticas de seguridad informática.

Lenguajes de desarrollo

HTML: es un lenguaje de gran importancia para controlar la estructura de cualquier sitio web. HTML tiene elementos básicos como encabezados, párrafos e imágenes.

CSS: esta es la presentación visual, permite a los desarrolladores dar estilo a los elementos con diversos colores, fuentes y diseños. Garantiza que el sitio sea atractivo y funcione bien en todos los dispositivos.

JavaScript: este es el lenguaje que se utiliza para hacer interactivas las páginas web, permitiendo contenidos dinámicos como formularios y animaciones.

Funciones:

1. Crear una estructura visual.
2. Presentar datos e información.
3. Usar el diseño responsive.
4. Optimizar el rendimiento.

Back-end (servidor)

El desarrollo *back-end*, o del lado del servidor, funciona con los servidores, bases de datos y lenguajes de programación tales como Python, PHP o Java. Un desarrollador *back-end* sabe crear y mantener aplicaciones, procesar solicitudes desde el *front-end*, actualizar la base de datos e interactuar con otros sistemas.

Lenguajes de desarrollo

Python: es conocido por su claridad y eficacia, Python es el lenguaje más utilizado para el desarrollo web *back-end*, con marcos como Django y Flask que permiten un desarrollo rápido.

Java: es un lenguaje orientado a objetos utilizado para aplicaciones a gran escala, Java es valorado por su alto rendimiento y fiabilidad en diferentes plataformas.

PHP: es manejado para contenidos web dinámicos y para integrar bases de datos, PHP es compatible con muchos sistemas de gestión de contenidos.

Ruby: es mayormente usado por su elegante sintaxis, Ruby y el *framework* Rails ofrecen un método sencillo para crear aplicaciones respaldadas por bases de datos.

Node.js: es un tiempo de ejecución de JavaScript que permite a los desarrolladores utilizar JavaScript en el lado del servidor, ayudando al desarrollo de aplicaciones de red rápidas y escalables.

Funciones:

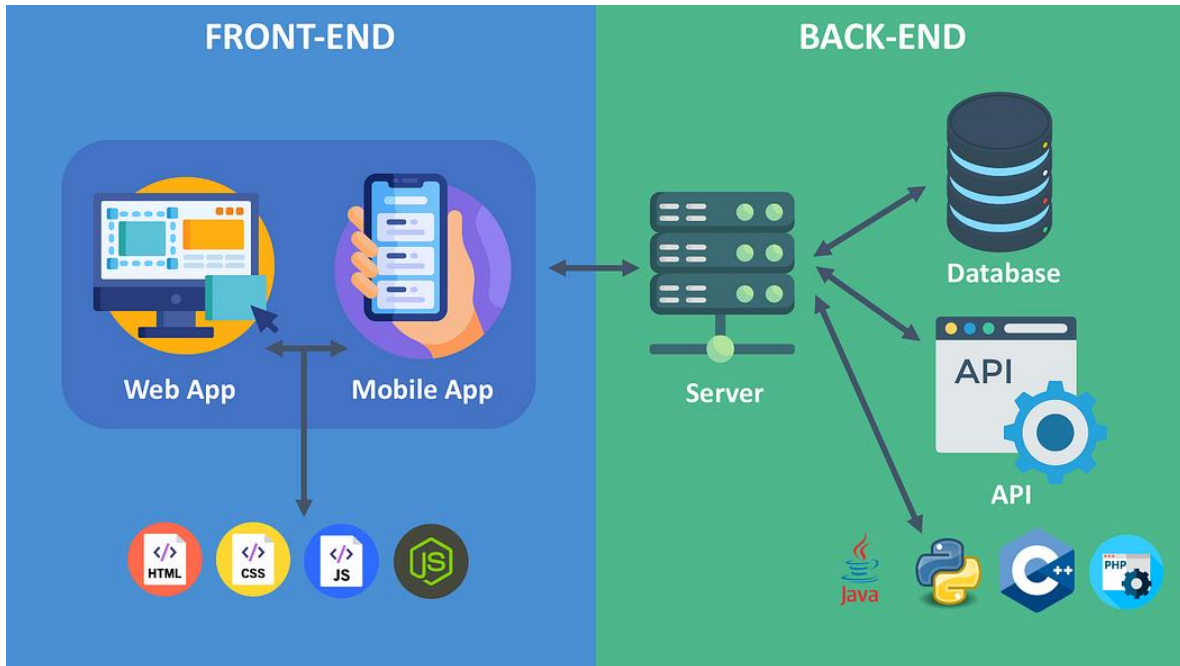
1. Gestionar la lógica de negocio.
2. Administrar el almacenaje y recuperación de datos.
3. Implementar mecanismos de seguridad.
4. Rastrear y gestionar las sesiones de los usuarios.
5. Optimizar el rendimiento.

Full stack

Los desarrolladores *full stack* tienen el conocimiento y las habilidades para ir en ambas direcciones, ya sea hacia el *front-end* o el *back-end*. Además, el desarrollo *full stack* se ocupa tanto de la parte del cliente como del servidor del sitio. Un buen desarrollador conoce los lenguajes de programación, las tecnologías y los *frameworks* más importantes, además, participa en todo el ciclo de desarrollo web.

Figura 26

¿Qué es front-end y back-end?



Fuente: (web mobileLIVE, 2023, junio 7, Back-end for front-end basics: A comprehensive guide - mobileLIVE. Medium).

Ventajas y desventajas del desarrollo web

Ventajas

1. Se tiene acceso a múltiples herramientas y marcos que les permiten diseñar soluciones únicas y flexibles.
2. Gracias a los recursos modernos como los microservicios, ahora los sitios web pueden gestionar con mayor eficacia el aumento de datos y usuarios.
3. Las técnicas de desarrollo eficaces mejoran el acceso a los sitios web y aumentan el alcance de la audiencia.
4. Las empresas que buscan progresar con sus ventas *online* pueden beneficiarse de un desarrollo web sólido.

Desventajas

1. Debido a que la creación de sitios web de alta calidad requiere conocimientos específicos, puede resultar costoso.

2. Los sitios web que tienen un mal diseño pueden no funcionar bien o cargarse con lentitud.
3. Pueden surgir proyectos complejos por la necesidad de gestionar una amplia gama de herramientas y tecnología.
4. Es necesario actualizar la seguridad de los sitios web, dado que pueden ser blanco de ataques.

Herramientas para desarrollo del proyecto

IDE

Un Entorno de Desarrollo Integrado (IDE) es una herramienta esencial para programadores y desarrolladores de *software* que facilita la escritura, depuración y administración del código. Un IDE combina varias funciones en una sola interfaz, incluyendo un editor de código, herramientas de depuración y opciones de compilación, lo que permite a los usuarios escribir código de manera más eficiente y efectiva (*IDE: ¿qué es un entorno de desarrollo integrado?*, s.f.), (*IDEs: Qué son, para qué sirven y los más populares - Blog de SW Hosting*, s.f.).

Editores de código

Los editores de código son herramientas esenciales para programadores, que facilitan la escritura y modificación del código. Una de sus ventajas es la codificación con colores según el tipo de datos y la sugerencia automática de completar líneas. Además, permiten una fácil navegación entre líneas y archivos, lo que agiliza la corrección de errores. Estas características hacen que los programadores puedan trabajar de manera más eficiente y precisa (Los 17 Mejores Editores de Código, s.f.), (*Explorar las características del editor de código - Visual Studio (Windows) / Microsoft Learn*, s.f.).

Figura 27

Editor versus IDE

EDITOR VS IDE

Con ambos puedes escribir código, pero ¿en qué se diferencian?

Editor	IDE
Software ligero con ayudas para escribir código (resaltado de sintaxis, autocompletado, etc).	Integra un editor con las herramientas que necesita un desarrollador (debugger, compilador, etc).
<ul style="list-style-type: none"> Soporta múltiples lenguajes y tecnologías. (Iconos: JS, php) Enfocado en archivos (no tienen el concepto de proyecto). Puedes agregar plugins para darle el poder de un IDE pero te toca configurar cada uno a mano. (Icono: Legos) 	<ul style="list-style-type: none"> Se especializa en un lenguaje o tecnología (Java, Python, Go, Android, etc). Enfocado en proyectos completos. Desde la primera línea hasta la salida a producción. (Icono: Maletín) Trae herramientas integradas y configuradas (ej. Android Studio trae un emulador de Android). (Icono: Teléfono Android)
EJEMPLOS DE EDITORES 	EJEMPLOS DE IDES

Domina la tecnología con EDteam y #NuncaTeDetengas
 ➔ ed.team/cursos

EDteam

Fuente: (*Editor vs IDE*, s.f.).

¿Qué es Visual Basic?

Se trata de un lenguaje de programación que revolucionó el panorama en sus inicios. Se convirtió en una opción fácil y sencilla para programar aplicaciones para Microsoft Windows.

En un principio utilizaba una sintaxis simple basada en BASIC que permitió a muchos aprender sobre programación en Visual Basic. Sin embargo, Artes Plásticas se ha ido renovando hasta adaptar su lenguaje a la arquitectura .NET de Microsoft.

Por este motivo, Visual Basic se convierte en uno de los lenguajes más utilizados para el sistema operativo Microsoft Windows. Y es que, además, integra diferentes lenguajes de programación en un mismo entorno, que es Visual Studio, como son Visual C++ y Visual C# (*¿En qué consiste la programación en Visual Basic?*, 2024, noviembre 6). Euroinnova International Online Education).

Figura 28

Visual Basic



Fuente: (Visual Basic, 2019, febrero 26, Lenguajes de Programación más populares).

Visual Basic .Net

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos desarrollado por Microsoft, que forma parte del ecosistema .NET. Lanzado en 2002 como sucesor de Visual Basic 6.0, VB.NET se creó para modernizar el popular Visual Basic clásico integrando las nuevas capacidades del .NET Framework, lo que permitió a los

desarrolladores aprovechar un entorno más robusto y flexible para la creación de aplicaciones (Frogamesformacion.com, s.f.).

Figura 30

Visual Basic.Net



Fuente: (*Tutorial online - visual basic .NET. s.f., Bibliadelprogramador.com*)

Capítulo III. Marco metodológico

¿Qué es un marco metodológico?

Según el artículo escrito por Ortega (2024), un marco metodológico es una estructura o conjunto de reglas y principios que guían el proceso de investigación. Es un mapa que ayuda a los investigadores a planificar y llevar a cabo su estudio de manera organizada y sistemática.

Estos son algunos puntos claves para el marco metodológico:

Diseño de investigación: el marco metodológico contiene el diseño general de la investigación, es decir, la manera en que se va a recopilar y analizar la información. Por ejemplo, si es un estudio experimental, observacional, de caso, etc.

Métodos de recopilación de datos: describe las herramientas y técnicas que se utilizarán para recopilar la información. Esto puede incluir lo que son las encuestas, entrevistas, experimentos, análisis de documentos, entre otros.

Población y muestra: esto indica quiénes son los participantes o elementos de estudio y el cómo se seleccionará un subconjunto relevante para realizar la investigación.

Procedimiento: esto define los pasos específicos que se seguirán para llevar a cabo la investigación. Para proporcionar ayuda, asegurar la consistencia y la transparencia del estudio.

Instrumentos de medición: si se utilizan encuestas, test u otros instrumentos, el marco metodológico describe cómo se desarrollarán y validarán para garantizar su fiabilidad.

Análisis de datos: explica cómo se procesarán y analizarán los datos recopilados. Esto puede incluir el uso de *software* estadístico, métodos cualitativos o una combinación de ambos, según el tipo de estudio.

Ética: esto considera los aspectos éticos, como la privacidad y el consentimiento de los participantes, para garantizar que la investigación se realice de manera ética y respetuosa.

Figura 31

Marco metodológico



Fuente: (Perfil, V. T. mi., s.f., *METODOLOGÍA DE LA INVESTIGACIÓN*. Blogspot.com.)

Importancia de un marco metodológico

Como se mencionó anteriormente, un marco metodológico es de vital importancia en una investigación por varias razones, no solo guía el proceso investigativo, sino que también mejora la calidad y la confiabilidad de los resultados, permitiendo a otros investigadores comprender, replicar y construir sobre el trabajo realizado.

Estas son algunas de las razones de tener esta estructura:

1. **Organización y estructura:** el marco metodológico proporciona una estructura clara y organizada para la investigación. Define los pasos a seguir,

desde la selección de métodos hasta la interpretación de los resultados, lo que ayuda a mantener la coherencia y la lógica en todo el proceso.

2. **Rigurosidad científica:** un marco metodológico sólido ayuda a la rigurosidad científica de la investigación. Establece reglas y normas que aseguran que los métodos utilizados sean apropiados para abordar las preguntas de investigación y que los resultados sean fiables y válidos.
3. **Reproducibilidad y generalización:** al seguir un marco metodológico, se permitirá la reproducción del estudio por otros investigadores. Esto es algo esencial para validar los resultados y generalizar las conclusiones a otras poblaciones o contextos.
4. **Claridad en la planificación:** esto es para ayudar a los investigadores a planificar sus estudios de manera eficiente. Al definir los métodos, las muestras y los procedimientos por adelantado, se minimiza la posibilidad de errores o malentendidos durante la implementación de la investigación.
5. **Énfasis en la pertinencia:** es para tener un buen marco metodológico y que se pueda adaptar a las características específicas de la investigación. Permite a los investigadores seleccionar métodos y técnicas que sean las más adecuadas para abordar sus preguntas de investigación, aumentando así la calidad y relevancia de los resultados.
6. **Control de variables:** esto facilita el control de las variables relevantes. Al establecer los criterios específicos para la selección de las muestras, la recopilación de datos y el análisis, el marco metodológico ayuda a minimizar la influencia de variables no deseadas en los resultados.

7. **Ética y responsabilidad:** esta parte incluye consideraciones éticas que protegen los derechos y el bienestar de los participantes. Es esencial para realizar las investigaciones de manera responsable y respetuosa.

Desarrollo de un marco metodológico

Definición de la investigación

Objetivos de investigación: se especifica claramente lo que se pretende lograr con la investigación. Define los objetivos generales y específicos.

Revisión de la literatura

Marco teórico: se realiza una revisión exhaustiva de la literatura existente sobre el tema. Aquí se identifican las teorías, conceptos y los modelos relevantes que sirvan como base para la investigación.

Diseño de investigación

Tipo de investigación: en esta parte se decide si la investigación será descriptiva, exploratoria, experimental, correlacional, entre otros.

Enfoque: se elige entre un enfoque cuantitativo, cualitativo o mixto, esto dependiendo de la naturaleza de las preguntas de investigación.

Población y muestra

Población de estudio: se identifica el grupo o población que será objeto de la investigación.

Criterios de inclusión/exclusión: se definen los criterios para seleccionar participantes.

Instrumentos de recopilación de datos

Selección de instrumentos: en esta parte se decide cuáles herramientas se usarán para la recopilación de datos, por ejemplo: encuestas, entrevistas, observaciones, pruebas, etc.

Validación y fiabilidad: si es necesario, se deben validar y verificar la confiabilidad de los instrumentos.

Procedimiento

Pasos detallados: esto describe de manera detallada los pasos que se deben seguir para llevar a cabo la investigación, desde la selección de la muestra hasta la recopilación y análisis de los datos.

Análisis de datos

Métodos de análisis: se especifican las técnicas estadísticas o cualitativas que se utilizarán para analizar los datos.

Interpretación de resultados: se muestra cómo se interpretarán los resultados obtenidos.

Aspectos éticos

Consideraciones éticas: para asegurarse de abordar aspectos éticos, como el consentimiento informado, la confidencialidad y el respeto a los participantes.

Plan de trabajo

Cronograma: se elabora un cronograma detallado que indique las etapas clave de la investigación y los plazos para cada una.

Presupuesto

Recursos necesarios: se estiman los recursos necesarios, como personal, equipo, material. Si es posible, también se incluye un presupuesto.

Validación del marco metodológico

Revisión por expertos: se considera al obtener retroalimentación de colegas o expertos en el campo para validar la capacidad del marco metodológico.

Redacción del marco metodológico

Estructura clara: redacta el marco metodológico de forma clara y organizada, siguiendo la estructura tradicional de introducción, objetivos, diseño de investigación, población y muestra.

Revisión continua

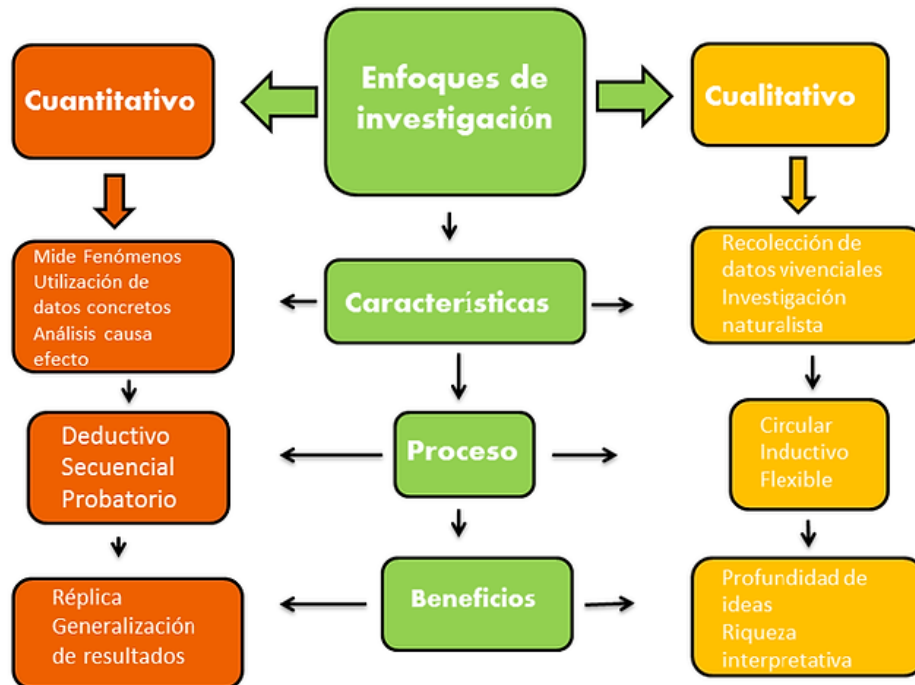
Adaptaciones: a medida que se avanza con la investigación, se necesita ser más flexible para realizar los ajustes en el marco metodológico, si son necesarios.

Enfoque de la investigación

El artículo escrito por Santos (2023) presenta que el enfoque de una investigación está definido por los intereses del investigador, el enfoque disciplinario, la base teórica y por aquellos factores que recaen dentro del problema de la investigación y que fueron desarrollados en el argumento del problema.

Por lo que el enfoque de investigación no se desarrolla hasta que se tenga por lo menos el argumento del problema y una breve base teórica relacionada con este. Se debe recordar que las cuestiones teóricas y metodológicas están estrechamente relacionadas y que todas las decisiones metodológicas están amparadas por decisiones teóricas.

La teoría sobre la metodología es aquella que define cuáles cuestiones teóricas están relacionadas con los aspectos procedimentales de la investigación, por ejemplo, la base teórica define ¿qué es una entrevista?, ¿qué es un cuestionario, ¿qué es un estudio de caso? O ¿qué es una muestra?

Figura 32*Enfoques de la investigación*

Fuente: (Compilado por: Ing. Pablo Emilio López Marquina, 2016, noviembre 21, ¿CUÁL ENFOQUE DARLE A UNA INVESTIGACIÓN? *misitio.*)

Tipos de enfoques*Enfoque cualitativo*

El enfoque cualitativo básicamente se utiliza cuando se está interesado en las cualidades del objeto investigado. Cuando el interés recae en el porqué de las cosas, las fuerzas en relación, la subjetividad de los sujetos involucrados, las relaciones entre los sujetos involucrados, así como las características de estas relaciones. Y, básicamente, cuando no se está interesado en cuantificar el proceso.

Figura 33*Métodos de investigación cualitativa*



Fuente: (*Investigación cualitativa* | *QuestionPro*, s.f.).

Características:

1. Entornos naturalistas.
2. Enfoque inductivo.
3. Perspectiva holística.
4. Subjetividad e interpretación.
5. Flexibilidad.

Tipos de investigación cualitativa

Entrevistas en grupo: las entrevistas pueden no ser individuales, también podrían ser grupales y los propósitos muy diversos.

Entrevistas estructuradas y no estructuradas: se pueden contar como una guía de preguntas abiertas o no, además, es posible aplicar las preguntas basado en lo que dice el entrevistador.

Métodos de observación cualitativa: se utiliza para recopilar información sobre los comportamientos no verbales de los sujetos. Tiene la ventaja de que permite estudiar un comportamiento de forma más efectiva al realizar esta acción de manera presencial.

Investigación etnográfica: entre las características que diferencian este tipo de investigación, están:

1. La estancia prolongada dentro de la comunidad estudiada para recoger información a través de la observación.
2. La elaboración de informes que dan prioridad a la contextualización y la relación interna de los fenómenos que se observan.
3. La presentación de interpretaciones que combinan voluntariamente la narrativa.
4. La descripción y la conceptualización teórica.

Análisis de redes sociales: el objetivo del análisis de redes es comprender a la comunidad mediante el mapeo de las relaciones que los conectan.

Enfoque cuantitativo

En el caso del enfoque cuantitativo, sí se encuentra enfocado en los números, estadísticas o en procesos que son básicamente relacionados con lo cuantificable. Es un enfoque que está relacionado con procesamiento estadístico a definiciones básicamente cuantificables.

Figura 34



Fuente: (Medina, R., s.f., *Enfoque Cuantitativo*).

Características:

1. Sirve para formular leyes y generalizaciones.
2. Se resguarda de procedimientos específicos para obtener datos.
3. Arroja información sobre una población.
4. Se trata de un método objetivo.
5. Se vale de la lógica deductiva.

Tipos de investigación cuantitativa

Investigación descriptiva: explica las propiedades, características y rasgos importantes del fenómeno estudiado mediante dinámicas objetivas de observación, análisis y demostración.

Investigación analítica: esto consiste en establecer una relación entre las variables del fenómeno estudiado En el cual se registran resultados para poder comprobar o refutar alguna hipótesis previamente establecida.

Investigación experimental: se trata de obtener información controlada mediante un experimento, de forma que se puedan comprender y eventualmente manipular las variables que determinan un fenómeno, para poder establecer cuáles son sus causas y sus efectos.

Enfoque mixto

Este tercer enfoque básicamente se trata de investigaciones en las cuales el interés recae en hacer una interpretación, por ejemplo, de la subjetividad de los sujetos involucrados en el problema, pero también se interesa cuantificar, analizar algunas estadísticas y aplicar métodos o técnicas provenientes del enfoque cuantitativo. Puede ser mixta con enfoque en lo cuantitativo.

Figura 35*Investigación mixta*

Fuente: (Pressbooks.pub., s.f.).

Características:

1. En el momento que se quiere validar o corroborar los resultados obtenidos con otros métodos.
2. Cada que se necesita utilizar un método para informar a otro.
3. Siempre que se quiera analizar continuamente una pregunta de investigación desde diferentes ángulos.

4. Cuando se quiere elaborar, clarificar o construir sobre los hallazgos de otros métodos.
5. Tan pronto como se quiera desarrollar una teoría sobre un fenómeno de interés y luego ponerla a prueba.
6. Si se quiere generalizar los resultados de la investigación cualitativa.

Tipos de investigación mixta

Diseño explicativo secuencial: este diseño implica la recopilación y análisis de datos cuantitativos, seguido de datos cualitativos, priorizando los datos cuantitativos e integrando conclusiones durante la fase de interpretación del estudio.

Diseño exploratorio secuencial: este diseño prioriza la investigación cualitativa, asegurando la recolección y análisis de datos a través de la recolección y análisis de datos cuantitativos, integra los hallazgos durante la fase de interpretación del estudio.

Estrategia concurrente de triangulación: este diseño utiliza una única fase de recolección y análisis de datos, separando la recolección y análisis de datos cuantitativos y cualitativos, integran los resultados durante la fase de interpretación del estudio.

Anidado concurrente: este diseño de investigación mixto utiliza una fase de recopilación de datos donde un método predominante se combina con un método menos prioritario, lo que da como resultado la fusión de datos durante la fase de análisis del proyecto.

Método de investigación

Como indica Compilatio (2024): Los métodos de investigación son procesos organizados y sistemáticos para obtener información y conocimiento sobre un tema en específico. Se basan en un conjunto de pasos y técnicas que permiten recolectar, analizar e interpretar datos de manera objetiva y confiable. Los métodos de investigación son

utilizados en diversas disciplinas, tanto académicas como científicas, para abordar preguntas de investigación, resolver problemas y llegar a conclusiones basadas en evidencias.

Fuentes de información

Según *Fuentes de información - Qué son, tipos y ejemplos* (s.f.): “Las fuentes de información son documentos que se consultan para obtener datos de un tema sobre el que se quiere saber más”.

En el mundo moderno, la información es amplia y accesible, pero es difícil distinguir entre fuentes de valor y erróneas. Identificar fuentes confiables y pertinentes es difícil, pero lo más importante es empleado para trabajos académicos, escolares, informes, conferencias, noticias, guías de clase, así como decisiones en empresas y organizaciones.

Fuentes primarias

Estas provienen de la experiencia directa o la observación de los hechos. Son documentos originales o primeras impresiones de un evento o investigación. A continuación, se mencionan algunos ejemplos:

- Documentos históricos.
- Entrevistas.
- Encuestas.
- Diarios de campo.
- Registros médicos.

Estas fuentes son importantes en una investigación, ya que proporcionan una comprensión directa de los fenómenos y sucesos estudiados.

Fuentes secundarias

Las fuentes secundarias son aquellas que pueden recoger, sintetizar y analizar información de las fuentes primarias. Estas son una compilación de información que otros han recogido y publicado previamente.

A continuación, se detallan ejemplos de fuentes secundarias.

- Los libros de texto.
- Los artículos de revistas.
- Los informes de investigación.

Estas fuentes son útiles para contextualizar los eventos o para obtener una visión más general sobre un tema.

Fuentes terciarias

Las fuentes terciarias sintetizan y presentan información de las fuentes secundarias.

Estas incluyen:

- Diccionarios.
- Enciclopedias.
- Directorios.
- Catálogos
- Otros recursos de referencia.

Son de gran utilidad para obtener información general sobre un tema, identificar fuentes secundarias relevantes, así como comprender términos y conceptos complejos.

Figura 36

Fuentes de información



Fuente: (3.3.1 Fuentes de información, s.f.).

Otros tipos de fuente de información

Existen otras maneras de obtener información mediante diversas fuentes. Estas se clasifican según alcance, formato y especialización.

Local, nacional o internacional

Dependiendo de la ubicación geográfica de la fuente, se podría encontrar información específica de una región o país, o de ámbito global. Es importante considerar la relevancia de la información según sea el enfoque del trabajo.

Textual o digital

Las fuentes de información pueden estar en formato impreso o digital. En la actualidad, las fuentes digitales son mayormente utilizadas debido a su facilidad de acceso y rapidez para obtener información. Sin embargo, es importante verificar la fiabilidad de la fuente y asegurarse de que no se trata de información falsa o manipulada.

General o especializada

Las fuentes de información pueden ser generales o especializadas. Las generales suelen ofrecer información amplia y básica sobre un tema; por otro lado, las fuentes especializadas brindan información más detallada y específica. Dependiendo del tema y de los objetivos de la investigación, es de vital importancia seleccionar fuentes que ofrezcan información relevante y útil.

Formales o informales

Las fuentes de información pueden ser formales o informales. Las formales son aquellas que están publicadas por instituciones o expertos reconocidos y verificadas por revisores y editores profesionales.

Algunas de las principales fuentes serían las siguientes:

1. Artículos técnicos-científicos.
2. Informes tecnológicos.
3. Informes de vigilancia tecnológica.
4. Legislación.
5. Marcas.
6. Medios de comunicación.
7. Ofertas y demandas tecnológicas.
8. Redes sociales.

9. Repertorio de dirigentes.

10. Tendencias.

11. Tesis.

Las informales, en cambio, incluyen blogs, redes sociales, foros y otros medios no profesionales donde cualquier persona puede publicar información.

Algunas de las principales fuentes serían las siguientes:

1. Clientes.
2. Comités y asociaciones empresariales.
3. Competidores.
4. Congresos, seminarios, jornadas.
5. Empresas subcontratadas.
6. Ferias, exposiciones, exhibiciones.
7. Fuentes internas de la empresa.
8. Inversores.
9. Misiones empresariales.
10. Proveedores.
11. Socios.

Es importante tener en cuenta que estas fuentes informales pueden contener información errónea.

¿Qué es una variable?

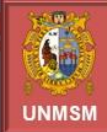
Se llama variables a todo aquello que es susceptible de asumir diferentes valores. En una metodología de investigación, se puede decir que una variable es aquello que se somete a comparación, con la finalidad de responder la pregunta de investigación.

Como se menciona en el artículo, es posible definir al término de variable como “Una característica o cualidad, magnitud o cantidad, que puede sufrir cambios, y que es objeto de análisis, medición, manipulación o control en una investigación” (Bianconi, 2024).

En el método hipotético-deductivo es fundamental, debido a que, a través de ellas, se validan los aportes al conocimiento científico. En este sentido, las variables son el eje transversal del proceso de investigación.

Figura 37

Cuadro de variables



Cuadro de Operacionalización de Variables para una investigación descriptiva					
Variables	Definición conceptual	Dimen-siones	Sub Dimensiones	OPERACIONALIZACIÓN	
				Indicadores	Tipo de variable
<u>VARIABLE DE ESTUDIO:</u> SATISFACCIÓN DEL CLIENTE	Se define la satisfacción del cliente como el nivel del estado de ánimo de una persona sobre el grado en que se han cumplido sus necesidades o expectativas establecidas, que resulta de comparar el rendimiento percibido de un producto y/o servicio con sus expectativas.	CALIDAD DEL SERVICIO	CAPACIDAD DE RESPUESTA	Capacidad de respuesta Disposición por resolver problemas del cliente	Categoría Ordinal
			EMPATÍA	Grado de comunicación cliente-empresa Dedicación de tiempo a cada cliente	Categoría Ordinal
			CONFIABILIDAD	Puntualidad Confianza	Categoría Ordinal
			SEGURIDAD	Grado de conocimiento y atención mostrados por los empleados Grado de seguridad del cliente en las transacciones con la empresa	Categoría Ordinal
			ELEMENTOS TANGIBLES	Apariencia del personal Apariencia de la instalación, equipos y del material de comunicación	Categoría Ordinal

FUENTE: Elaboración propia.

Fuente: (Ingenieroduran, P., s.f.).

Requisitos funcionales

Los requisitos funcionales en ingeniería de *software* también se llaman especificación funcional. Estas son las descripciones del servicio que un *software* debe ofrecer, a su vez, describe un sistema y sus componentes.

Una función no es más que una entrada al sistema del *software*, su comportamiento y salida. Este puede ser un cálculo, manipulación de datos, interacción del usuario, proceso de negocio o cualquier otra funcionalidad específica que defina qué función es probable que realice un sistema.

En la ingeniería de sistemas o ingeniería de *software*, un requisito funcional puede variar desde una declaración abstracta de alto nivel de la necesidad del remitente hasta especificaciones matemáticas detalladas.

Tipos de requisitos funcionales

1. Manejo de transacciones.
2. Reglas de negocios.
3. Requisitos de certificación.
4. Los requisitos de información.
5. Funciones administrativas.
6. Niveles de autorización.
7. Seguimiento de auditoría.
8. Interfaces externas.
9. Gestión de datos históricos.
10. Requisitos legales y reglamentarios.

Requerimientos no funcionales

Son también conocidos como “requisitos de calidad” o “requisitos de atributos del sistema”. Los requisitos no funcionales son criterios y características que describen cómo debe ser la seguridad, el rendimiento, la usabilidad y otros aspectos de un sistema o *software* más allá de su funcionalidad básica.

El utilizar requisitos no funcionales es una parte muy importante al momento de desarrollar un sistema, ya que hará que el *software* o aplicación sea eficaz, eficiente y cumpla con las expectativas y necesidades del usuario.

Sin embargo, se debe comprender que estos trabajan de la mano con los requisitos funcionales. Ya que los requisitos funcionales se enfocan en el *qué*, mientras que los requisitos no funcionales se enfocan en el *cómo* y en qué medida cumplen con ciertas cualidades.

Tipos de requisitos no funcionales

1. Rendimiento.
2. Seguridad.
3. Usabilidad.
4. Disponibilidad.
5. Fiabilidad.
6. Mantenibilidad.
7. Portabilidad.
8. Escalabilidad.

Ventajas de los requisitos no funcionales

1. Mejora de la calidad del sistema.

2. Experiencia del usuario mejorada.
3. Optimización del rendimiento.
4. Cumplimiento normativo.
5. Facilitación de mantenibilidad.
6. Mayor seguridad.
7. Cumplimiento normativo.
8. Planificación y gestión efectiva.
9. Competitividad y reputación.

Desventajas de los requisitos no funcionales

1. Difíciles de entender e implementar.
2. Llevan mucho tiempo.
3. Son costosas de probar.
4. Afectar la funcionalidad del sistema si no se implementan de manera correctamente.
5. Dificultades de mantenimiento.
6. Pueden llegar a generar conflictos entre requisitos.

Ejemplos de requisitos no funcionales

Redes sociales: deben ser capaces de aumentar los usuarios por medio de cuentas y los datos del usuario deben estar protegidos. Además, tiene que ser versátil para todos los usuarios, sin importar su idioma o cultura.

Aplicación de banca en línea: tanto el cliente como el servidor deben estar encriptados mediante el protocolo HTTPS y el código fuente debe estar bien documentado, así como seguir las mejores prácticas. También, la interfaz debe ser accesible y utilizable para personas con discapacidades visuales.

Sistema de gestión de inventario: el sistema debe tener la capacidad de funcionar durante al menos 6 meses continuos sin requerir reinicios a causa de fallos y, en casos de una caída del servidor, el sistema debe tener la capacidad de recuperar y restaurar los datos del inventario desde la última copia de seguridad. Además, las consultas de búsqueda y filtrado de inventario deben completarse en menos de un segundo.

Sistema de comercio electrónico: el sistema debe tener la capacidad de manejar un aumento del tráfico durante la temporada de ventas sin degradar el rendimiento, por lo que el sistema debe estar disponible las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad máximo de una hora al mes para mantenimiento del programa. Por otro lado, la página debe tener un tiempo de cargar de por lo menos 2 segundos.

Plan de pruebas

Según la definición de *Plan de pruebas de software: 8 pasos para realizarlas - icaria Technology* (s.f.): “Un plan de pruebas es un documento que describe los pasos que se deben de llevar a cabo y el enfoque de las pruebas en un proyecto de desarrollo de software”.

El objetivo del plan es garantizar el rendimiento y la calidad del *software* antes del lanzamiento. Además, el plan se ocupa de lo siguiente:

1. Establecer los objetivos y metas específicas.
2. Proporciona la estructura para las actividades de testing.
3. Determinan los criterios de aceptación de cada prueba.
4. Facilitan la evaluación objetiva en vistas a determinar si el *software* es apto para su lanzamiento.

¿Cómo se realiza un plan de pruebas?

1. Comprender los objetivos del plan de pruebas.
2. Determinar qué nuevas funcionalidades se desea testear.
3. Definir la estrategia de pruebas.
4. Establecer los entornos requeridos.
5. Definir los criterios.
6. Elección de la metodología.
7. Estimar los requisitos de personal.
8. Planificación.

Tipos de pruebas de software

1. **Prueba de aceptación:** verifica que todo funcione según lo previsto.
2. **Pruebas de integración:** se asegura de que las funciones del *software* o componentes operen juntas.
3. **Pruebas de unidad:** esto valida que cada dispositivo de *software* funcione según lo esperado.
4. **Pruebas funcionales:** verifica las funciones por medio de emulaciones de escenarios de negocio.
5. **Pruebas de rendimiento:** prueba cómo funciona el *software* bajo diferentes cargas de trabajo.
6. **Pruebas de regresión:** por medio de esto, se puede verificar si las nuevas características rompen o degradan la funcionalidad.
7. **Pruebas de estrés:** esto prueba cuánta tensión puede soportar el sistema antes de que falle.

8. **Pruebas de usabilidad:** mediante esto se valida qué tan bien un cliente puede usar un sistema o una aplicación web para completar una tarea.

Casos de uso

Según *Definición de casos de uso - Documentación de IBM* (s.f.b.): “Los casos de uso se crean para refinar un conjunto de requisitos basados en un rol o tarea”.

En lugar de la lista tradicional de requisitos que pueden no abordar directamente el uso de la solución, los casos de uso agrupan requisitos comunes basados en el tipo de rol u objetivo. Los casos de uso definen lo que los usuarios o roles están haciendo en la solución, un proceso de negocio define cómo realizan estas funciones.

Figura 38

Ejemplo de casos de uso registrar cliente

Especificación de Casos de Uso de Fachada							
No.	01						
Nombre.	Registrar Cliente						
Descripción	Cuando un vendedor inicia los contactos con un nuevo cliente, aún cuando no realice una venta, debe ingresar los datos básicos del cliente.						
Fase	Fachada						
Actores	Principal: Vendedor						
Guión	<table border="1"> <thead> <tr> <th>Actor</th> <th>Sistema</th> </tr> </thead> <tbody> <tr> <td>1. Ingresa su login</td> <td></td> </tr> <tr> <td></td> <td>2. Verifica que exista un usuario con ese login</td> </tr> </tbody> </table>	Actor	Sistema	1. Ingresa su login			2. Verifica que exista un usuario con ese login
Actor	Sistema						
1. Ingresa su login							
	2. Verifica que exista un usuario con ese login						

Fuente: *(Casos de Uso y el Proceso del Requerimiento, (s.f.).*

Aplicaciones de casos de uso

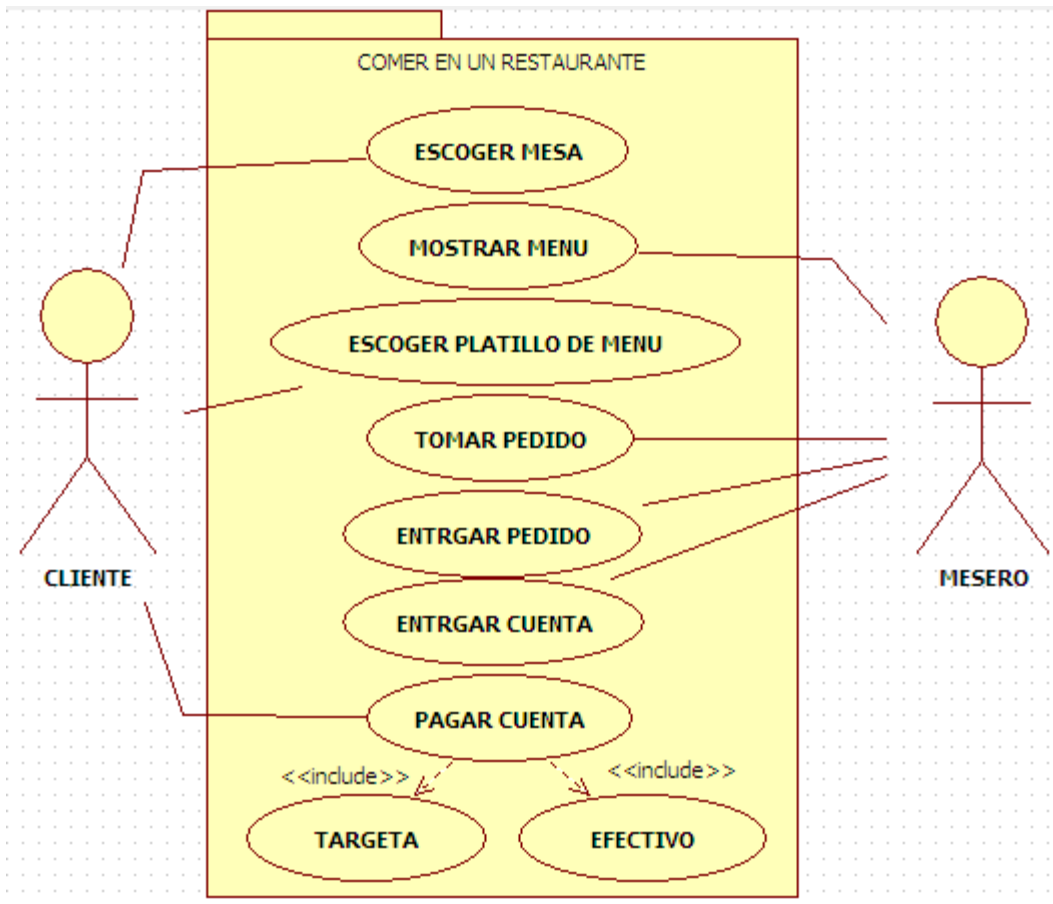
1. Diseñar interacciones claras.
2. Probar y evaluar prototipos.
3. Comunicación efectiva.
4. Documentación del sistema.
5. Evaluación de riesgos.
6. Optimización continua.

Componentes del diagrama de caso de uso

1. **Actores:** son los usuarios.
2. **Caso de uso:** representan las acciones.
3. **Relaciones:** muestra la relación entre caso de uso y actores.
4. **Sistema:** agrupan diferentes elementos.

¿Cómo hacer un diagrama de caso de uso?

1. Recopilar fuentes de información.
2. Identificar actores potenciales.
3. Identificar posibles casos de uso.
4. Conectar los casos de uso.
5. Describir actores.
6. Buscar más casos de uso.
7. Documentar casos de uso.
8. Relacionar modelos entre casos de uso empresarial.
9. Verificar la vista.

Figura 39*Diagrama de casos de usos*

Fuente: (Completo, V. mi P., s.f).

Instrumentos

En este caso, se utiliza una mezcla entre entrevistas a los usuarios implicados, así como observaciones y cuestionarios. Todo con la finalidad de obtener la información los más detalladamente posible.

Capítulo IV. Análisis de resultados

Plan de gestión de riesgos

Un plan de gestión de riesgos es un documento que describe cómo se identificarán, gestionarán, evaluarán y supervisarán estos sucesos. Este documento puede referirse a un proyecto, a un período específico o incluso una actividad o sector en especial. Además, un plan de riesgos es esencial para garantizar que los posibles problemas se aborden de forma sistemática y proactiva. De esta forma, se puede minimizar los impactos negativos, también, encontrar y aprovechar oportunidades de mejora continua.

Figura 40

Gestión de riesgos

Gestión del riesgo de desastres



Fuente: (Sepúlveda, C. A. R., 2022, julio 7). *Planes de emergencia y gestión del riesgo de desastres ¿ha llegado su fin?*

Componentes principales

Identificación de riesgos: el tener un método para enumerar e identificar todos los riesgos potenciales que podrían afectar el proyecto. Este puede ser un documento, ya sea en

una hoja de cálculo o un *software*. Lo importante es contar con un criterio bien definido y un control constante de todo esto.

Análisis del riesgo: es para evaluar las probabilidades de ocurrencia y el impacto de cada riesgo que fue identificado en el paso anterior. Ese análisis puede ser cualitativo o cuantitativo.

Plan de respuesta al riesgo: para incluir estrategias y acciones específicas que tengan como objetivo el abordar o mitigar los riesgos y así analizarlos con anticipación.

Categorías:

Evitar: se debe cambiar el plan de riesgos para eliminarlos.

Reducir: se implementan acciones para reducir la probabilidad de que ocurran.

Transferencia: este tiene como objetivo transferir el riesgo a otra parte.

Aceptar: este plan de respuesta reconoce los posibles riesgos y prepara a la empresa para enfrentarlos.

Seguimiento y revisión: sigue los riesgos y evalúa la eficiencia de las estrategias para responder. Aquí, también se puede modificar el plan de riesgos de ser necesario.

Documentación y comunicación: por último, se debe registrar toda la información relacionada con los riesgos y estrategias. Además, se debe contar con medios de comunicación que sean ágiles y fáciles para todos los que necesiten utilizarlas.

Tareas de la gestión de riesgos

Identificar:

- Considerar las amenazas potenciales, incluida la gestión avanzada del riesgo de los datos en la ingeniería de *software* y la gestión global del riesgo.
- Definir un riesgo en relación con la gestión de proyectos de *software* y evaluar su importancia.

Clasificar y priorizar:

- El clasificar los riesgos, aplicando la gestión de riesgos en la gestión de proyectos de *software*.
- Priorizar los riesgos identificados del proyecto de *software* para centrarse en los de mayor impacto.

Desarrollar un plan de acción:

- El crear una plantilla de acción para la mitigación los riesgos y aprovechando el *software* de mitigación.
- El abordar las aplicaciones específicas de la gestión de riesgos en la gestión de proyectos de *software*, se podrán adaptar las estrategias en consecuencia.

Supervisar continuamente:

- Plantar la supervisión continua en la gestión de riesgos del ciclo de vida del *software*.
- Promover la comunicación constante dentro del equipo para identificar y abordar nuevas amenazas.

Aplicar planes de acción:

- Actúe conforme al plan de acción si se materializa alguna amenaza identificada.
- Tramitar los riesgos de desarrollo de productos de forma proactiva mediante una evaluación eficaz de los riesgos del *software*.

Matriz de riesgos

Como describe *Asana* (s.f.): “Una matriz de riesgos es una herramienta que análisis los riesgos y se utiliza para evaluar la probabilidad y la gravedad del riesgo durante el proceso de planificación del proyecto”. Una vez que se haya evaluado la probabilidad y la

gravedad de cada riesgo, se pueden ubicar en la matriz para calcular la calificación del impacto de cada riesgo.

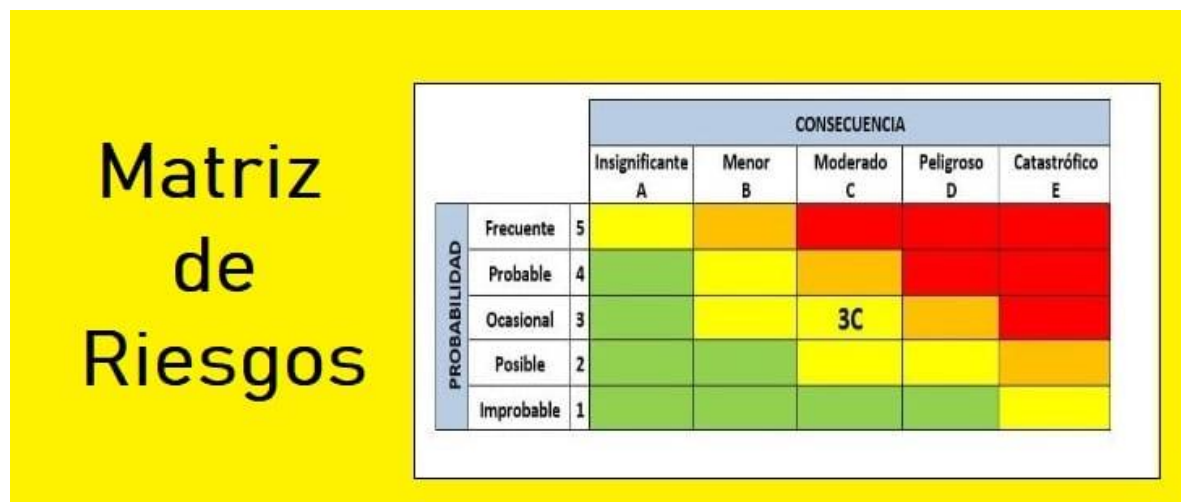
Estas calificaciones ayudarán al equipo a establecer cuál prioridad asignar a los riesgos del proyecto y a gestionarlos de manera efectiva.

Tipos de riesgos

- **Riesgo estratégico:** implican los errores de rendimiento o de decisión.
- **Riesgo operativo:** son los errores de proceso o de procedimiento.
- **Riesgo financiero:** esto puede involucrar varios eventos que causen una pérdida de ganancias para la empresa.
- **Riesgo técnico:** se puede incluir cualquier aspecto relacionado con la tecnología de la empresa.
- **Riesgo externo:** son los que están fuera del control humano.

Figura 41

Matriz de riesgos



Fuente: (Laborales, R., 2023, enero 16, *Matriz de Riesgos en la Empresa*. Riesgoslaborales.info.)

Estudio de factibilidad

Como indica el artículo de *Estudio de factibilidad - Qué es y qué tipos hay* (s.f.):

“Un estudio de factibilidad es el que hace una empresa para determinar la posibilidad de poder desarrollar un proyecto o un negocio”.

Puntos clave:

1. Ayuda a las empresas a tomar mejores decisiones.
2. Revela posibles desafíos que el proyecto podría tener, permitiendo a la empresa prepararse adecuadamente.
3. Facilitan una base para desarrollar estrategias efectivas que aumenten la probabilidad de éxito del proyecto.

Figura 42

Estudio de factibilidad



Fuente: (*Estudio de factibilidad - Qué es y qué tipos hay*, s.f.).

Análisis de factibilidad

A través de este procedimiento, se evaluarán las metas del proyecto actual a fin de asegurar que su ejecución sea factible para la microempresa, demostrando que se cuenta con el personal adecuado y la infraestructura necesaria para su implementación.

Destacar que la viabilidad es la medida de posibilidad de completar una acción, al evaluar todos los elementos del proyecto. Tal como se ha mencionado anteriormente, en este estudio, las ventajas que se podrían lograr al realizar la iniciativa son:

Organización y gestión más eficiente de los datos.

1. Control de pedidos.
2. La optimización de procesos reducirá el tiempo operativo y aumenta la eficiencia.
3. Control de acceso por roles (solo se utilizará el administrador).
4. Seguridad en el manejo de información.

Factibilidad técnica

Este aspecto evalúa si la infraestructura técnica que posee el restaurante puede responder de manera eficiente y favorable para desarrollar un proyecto o negocio que se tiene panificado. Además, se debe verificar si las personas poseen los conocimientos técnicos necesarios para utilizar el equipo y el *software* necesario.

Con respecto al *hardware* requerido para este proyecto, el servidor o terminal por utilizar debe cumplir con los requisitos mínimos, los cuales son los siguientes:

Tabla 1

Recursos del equipo

Características	Especificaciones
Sistema Operativo	Windows 10/ 11
	Windows server
Procesador	Intel Core i3 en adelante
	Ryzen 3 en adelante
Almacenamiento	128 Gb en adelante
Memoria RAM	4 Gb en adelante
Marca	Cualquiera
Impresora	Cualquiera

Al valorar el *hardware* y tener en cuenta la configuración mínima requerida, el restaurante no tiene que invertir dinero, ya que cuenta con los equipos necesarios para el correcto funcionamiento del sistema de pedidos que se va a desarrollar.

Con respecto al *software*, se dispone de todas las aplicaciones necesarias para el desarrollo del proyecto, como SQL Server y Visual Studio 2022. El sistema operativo de los dispositivos es Windows 10, lo cual indica que no se requiere gastar más dinero en la compra de otros programas.

Factibilidad económica

Se debe realizar un análisis exhaustivo de la relación costo-beneficio del negocio o proyecto y calcular ambos aspectos. Si en la valoración se observa que los costos superan a los beneficios, sería mejor no desarrollarlo. En cambio, si el beneficio supera los costos, la decisión de la realización del proyecto se vuelve menos arriesgada, aunque no implica que no existan riesgos.

Debido a que el sistema está siendo desarrollado por un pariente del dueño del restaurante, no implicará un gasto adicional. Y dado que se está desarrollando como parte de una tesis académica, significa que no generará gastos adicionales. No se cobrará por el

servicio prestado, ya que se trata de un proyecto educativo. Por lo tanto, los costos asociados al desarrollo no se reflejarán en la contabilidad del restaurante.

De acuerdo con la evaluación realizada, a continuación, se detallan los gastos financieros necesarios para llevar a cabo el proyecto de *software*.

Tabla 2

Costo de proyecto

Concepto	Descripción	Costo
Desarrollo	Por un pariente del dueño del restaurante	\$800
Horas de trabajo	Trabajo del desarrollador (250 horas x \$10/hora)	\$2500
Proyecto	Parte de una tesis académica	\$0
Contabilidad	No generará gastos en el restaurante	\$800

Tabla 3

Costos del proyecto

Concepto	Descripción	Costo
Desarrollo	Por un estudiante de ingeniería informática	\$800
Horas de trabajo	Trabajo del desarrollador (250 horas x \$10/hora)	\$2500
Proyecto	Parte de una tesis académica	\$0
Contabilidad	No generará gastos en la microempresa	\$800

Factibilidad operativa

La factibilidad operativa está relacionada con el personal que debe realizar el proyecto. Sobre todo, depende de los recursos humanos que forman parte de la organización. En consecuencia, al momento de desarrollar un proyecto o negocio, se tiene que implementar un sistema.

En resumen, la factibilidad operativa se encarga de determinar si un proyecto de desarrollo de *software* es sostenible en términos operativos, teniendo en cuenta todos los factores que puedan afectar su implementación y éxito a largo plazo.

Se debe considerar lo siguiente:

1. Que no sea muy complicado para el usuario que lo va a operar.
2. Observar si el usuario se resiste a usarlo por miedo o por cualquier otra razón.
3. Reconocer si el usuario se puede adaptar a los cambios, sobre todo si el cambio se produce de forma muy rápida.
4. Se debe valorar si no se puede volver obsoleto de forma muy rápida y buscar planes de contingencia.

Descripción del proyecto

1. Desarrollo de un sistema de toma de pedidos, utilizando Visual Studio con formularios en .Net, con estilos, módulos, clases y diferentes paquetes.
2. Funcionalidades: autenticación de usuarios, gestión de productos, categorías, clientes, pedidos, detalles de pedidos, mantenimiento y manual de usuario.
3. Parte de una tesis académica, sin costos adicionales para la microempresa.

Objetivos del proyecto

1. Facilitar la gestión de pedidos para el restaurante.
2. Mejorar la trazabilidad de las operaciones.
3. Proporcionar un sistema seguro y fácil de usar.

Recursos necesarios

Hardware:

1. Impresoras para tickets y facturas.
2. Computadoras para los usuarios.

Software:

1. Visual Basic. Net.

2. Bases de datos SQL Server.

Recursos humanos

1. Un estudiante de Ingeniería Informática con conocimiento en formularios de Visual Basic.Net

Evaluación de riesgos

1. Dependencia del desarrollador: riesgo de falta de disponibilidad del único desarrollador.
2. Tiempo de implementación: ajustar el desarrollo e implementación al cronograma académico.
3. Problemas de programación: problemas de versiones.

Evaluación de la viabilidad operativa

1. La microempresa no incurre en costos adicionales significativos.
2. La experiencia del estudiante es suficiente para cumplir con los objetivos del proyecto.

Figura 43

Tipos de factibilidad

Tipos de estudio de factibilidad

HubSpot



Factibilidad operativa

Evalúa si el proyecto puede implementarse utilizando los recursos y procesos operativos disponibles de manera efectiva.



Factibilidad técnica

Determina si la tecnología necesaria para llevar a cabo el proyecto está disponible, accesible y adecuada para su propósito.



Factibilidad económica

Analiza la viabilidad financiera del proyecto, considerando costos, ingresos potenciales, retorno de la inversión y rentabilidad a largo plazo.



Factibilidad comercial

Examina la demanda del mercado, la competencia y la viabilidad de comercialización del producto o servicio asociado al proyecto.



Factibilidad legal

Evalúa la viabilidad del proyecto en términos de cumplimiento de regulaciones, políticas gubernamentales y requisitos legales.



Factibilidad de tiempo

Determina si el proyecto puede completarse dentro de un marco de tiempo razonable, considerando plazos, recursos disponibles y posibles retrasos.



¡Recuerda!

Estos tipos de estudio de factibilidad proporcionan una visión integral que ayuda a evaluar la viabilidad global de un proyecto desde diferentes perspectivas.

Fuente: (Rodrigues, N, 2022, octubre 25, Qué es la factibilidad de un proyecto, cómo se estudia y ejemplos.)

Metodología utilizada

La metodología que se utilizó para el desarrollo del sistema de control y gestión de inventarios para el Restaurante Finca la Bandera es Kanban, ya que esta se enfoca en la visualización del trabajo, la gestión del flujo y la mejora continua. Al usar Kanban, se logró:

1. **Observar el progreso:** se implementó un tablero de Kanban para poder visualizar el estado actual de las tareas en tiempo real, lo que simplifica la gestión y orden de estas.
2. **Adaptabilidad:** debido a que es posible modificar el ritmo de trabajo según las demandas del proyecto y las actividades en desarrollo.
3. **Enfoque en eficacia:** se reduce la cantidad de trabajo en curso para prevenir la saturación y garantizar que cada tarea se finalice de forma eficiente.
4. **Mejora continua:** mediante revisiones periódicas del tablero, se detectan bloqueos y se implementan decisiones para perfeccionar el proceso de desarrollo.

Figura 44

Sistema para toma de pedidos en Kanban

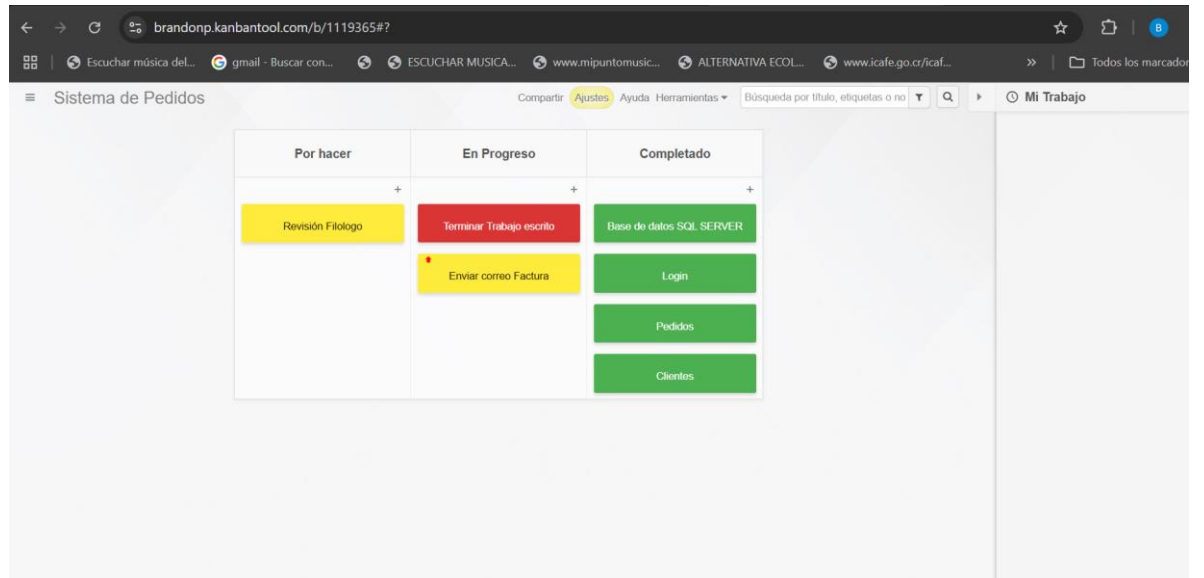
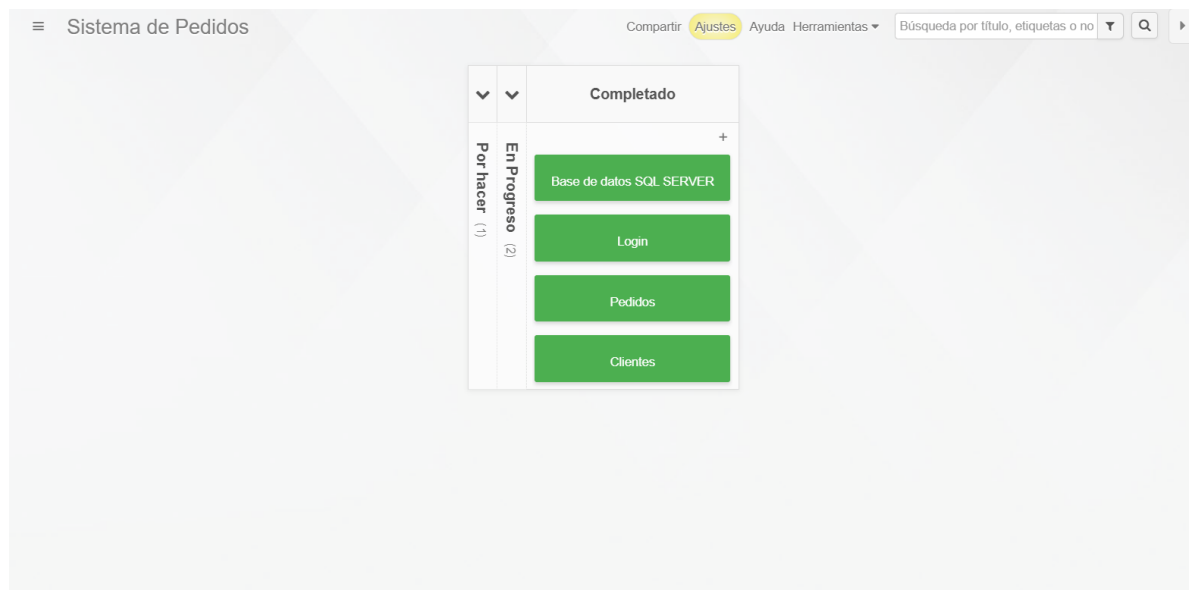


Figura 45

Tareas terminadas sistema para la toma de pedidos Kanban



Capítulo V. Conclusiones y recomendaciones

Conclusiones

1. Identificación de la limitante principal: la investigación inicial reveló que la falta de un sistema para la toma de pedidos es una limitante para el Restaurante Finca La Bandera. Debido a la falta de una solución segura, la toma de decisiones se dificultaba y esto afectaba a la eficiencia operativa.
2. Objetivo del prototipo: el prototipo fue desarrollado como parte de esta tesis y tiene como objetivo demostrar la posibilidad de un sistema eficiente que permita el control y la toma de pedidos. En él se incluyeron funciones como el registro, inclusión de usuarios, editar, agregar productos, mantenimiento, facturación y manual de usuario.
3. Estado del prototipo: actualmente, el sistema se presenta como un prototipo funcional. Aunque se ha diseñado para satisfacer las necesidades del Restaurante Finca La Bandera, su implementación aún está en discusiones y no se ha definido un plan de acción.
4. Implementación técnica: se utilizó SQL Server como base de datos, ya que garantiza un entorno confiable y seguro para el almacenamiento de datos. Lo que garantiza un crecimiento normalizado y controlado de los datos. Esta implementación proporciona una base sólida para futuras expansiones y mejoras del sistema.
5. Proyección de resultados: se espera que, si el proyecto es implementado, el sistema mejoraría de manera significativa la toma de pedidos en Restaurante Finca La Bandera. El desarrollo de un *software* propio facilitará el acceso a la información y permitirá un control más riguroso sobre los procesos internos. La implementación del sistema propuesto mejorará la gestión de

los clientes, usuario, categorías, pedidos, productos, facturas y envíos al correo del cliente para usos de contabilidad.

Recomendaciones

1. Evaluación del prototipo: se recomendó realizar una evaluación más exhaustiva del prototipo para así identificar las áreas que se deben mejorar y determinar su efectividad en el contexto real de Restaurante Finca La Bandera. Esta evaluación ayudará a tomar la decisión sobre el implemento, para asegurar que el sistema sea completamente funcional y eficiente.
2. Planificación de implementación: en caso de que se decida avanzar hacia la implementación, es importante desarrollar un plan detallado que incluya un cronograma y las fases necesarias para garantizar una implementación exitosa del sistema.
3. Documentación técnica y de usuario: es importante mantener una documentación completa del prototipo, incluyendo especificaciones técnicas. Esto será útil tanto para la posible implementación como para el mantenimiento del sistema.
4. Fomento de la retroalimentación: se sugiere crear canales para que los usuarios potenciales proporcionen retroalimentación sobre el prototipo. Esto permitirá hacer ajustes necesarios y mejorar la experiencia del usuario, aumentando así la probabilidad de éxito en una eventual implementación.

Capítulo VI. Análisis de requerimientos

Análisis de requerimientos

A continuación, se muestran los requerimientos que sirven para obtener un correcto funcionamiento del prototipo desarrollado en Restaurante Finca La Bandera.

Requerimientos de usuario.

1. Requerimientos funcionales.
2. Requerimientos no funcionales.

Requerimientos de usuario

Los requisitos de usuario son las necesidades, expectativas y deseos que los usuarios tienen respecto a un producto. Estos pueden abarcar desde la funcionalidad, usabilidad y experiencia en general, además, son esenciales para diseñar productos que satisfagan las necesidades y expectativas de los usuarios de manera efectiva (*Estrategias para gestionar y priorizar requisitos de usuario en proyectos UX*, s.f.).

Los requisitos de usuario también pueden incluir preferencias específicas, limitaciones tecnológicas o contextuales, así como expectativas emocionales y psicológicas. Comprender estos requisitos es esencial para diseñar productos que sean intuitivos, eficientes y atractivos para su público (*Estrategias para gestionar y priorizar requisitos de usuario en proyectos UX*, s.f.).

A continuación, se presentan las tablas con los requisitos del usuario de acuerdo con las solicitudes, los cuales son importantes para el correcto desarrollo del sistema.

Tabla 4

Requerimiento de usuario RUM01

Responsable: Brandon Piedra Chacón			
Id:	RUM01	Prioridad:	Alta
Descripción:			

Diseñar un sistema para la toma de pedidos en un restaurante pequeño.
Fuente:
Dueño del Restaurante
Dependencias:
Ninguna

Tabla 4, Requerimiento generado por el estudiante para Restaurante Finca La Bandera, para la toma de pedidos.

Tabla 5

Requerimiento de usuario RUM02

Responsable: Brandon Piedra Chacón			
Id:	RUM02	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para el login que reconozca el inicio de sesión.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 5, Requerimiento generado por el estudiante para Restaurante Finca La Bandera para el ingreso al sistema.

Tabla 6

Requerimiento de usuario RUM03

Responsable: Brandon Piedra Chacón			
Id:	RUM03	Prioridad:	Alta
Descripción:			

Desarrollar un formulario para registro de usuario.
Fuente:
Dueño del Restaurante
Dependencias:
Ninguna

Tabla 6, Requerimiento generado por el estudiante para Restaurante Finca La

Bandera, para registrar usuario.

Tabla 6, Requerimiento generado por el estudiante para el Restaurante Finca La Bandera para el registro de usuarios

Tabla 7

Requerimiento de usuario RUM04

Responsable: Brandon Piedra Chacón			
Id:	RUM04	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para ingreso de productos.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 7, Requerimiento generado por el estudiante para Restaurante Finca La

Bandera, para ingresar productos.

Tabla 8

Requerimiento de usuario RUM05

Responsable: Brandon Piedra Chacón			
Id:	RUM05	Prioridad:	Alta

Descripción:
Desarrollar un formulario para registro de clientes.
Fuente:
Dueño del Restaurante
Dependencias:
Ninguna

Tabla 8, Requerimiento generado por el estudiante para Restaurante Finca La Bandera, para registrar clientes.

Tabla 9

Requerimiento de usuario RUM06

Responsable: Brandon Piedra Chacón			
Id:	RUM06	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para registro de Categorías.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 9, Requerimiento generado por el estudiante para Restaurante Finca La Bandera, para registrar categorías.

Tabla 10

Requerimiento de usuario RUM07

Responsable: Brandon Piedra Chacón			
Id:	RUM07	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para registro de pedidos.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 10, Requerimiento generado por el estudiante para Restaurante Finca La Bandera, para registrar pedidos.

Tabla 11

Requerimiento de usuario RUM08

Responsable: Brandon Piedra Chacón			
Id:	RUM08	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para registro detalles de pedido.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 11, Requerimiento generado por el estudiante para Restaurante Finca La Bandera, para registrar detalles de pedido.

Tabla 12*Requerimiento de usuario RUM10*

Responsable: Brandon Piedra Chacón			
Id:	RUM14	Prioridad:	Alta
Descripción:			
Desarrollar un formulario para mostrar los datos solicitados.			
Fuente:			
Dueño del restaurante			
Dependencias:			
Ninguna			

Tabla 12, Requerimiento generado por el estudiante para controlar y ver la cantidad de datos dentro de cada formulario.

Requerimientos funcionales

A continuación, se especifica la lista de requisitos funcionales que el sistema debe cumplir para asegurar un correcto funcionamiento.

Tabla 13

Requerimiento funcional RFM01

Responsable: Brandon Piedra Chacón			
Id:	RFM01	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los datos de los clientes, siendo el administrador el responsable de estas acciones.			
Fuente:			
Dueño del restaurante			
Dependencias:			
Ninguna			

Tabla 13, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los datos de los clientes.

Tabla 14

Requerimiento funcional RFM02

Responsable: Brandon Piedra Chacón			
Id:	RFM02	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los productos, siendo el administrador el responsable de estas acciones.			
Fuente:			
Dueña del Restaurante			
Dependencias:			
Ninguna			

Tabla 14, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los productos, siendo el administrador el responsable de estas acciones.

Tabla 15

Requerimiento funcional RFM03

Responsable: Brandon Piedra Chacón			
Id:	RFM03	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los registros de pedidos, siendo el administrador el responsable de estas acciones.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 15, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los registros de pedidos, siendo el administrador el responsable de estas acciones.

Tabla 16

Requerimiento funcional RFM04

Responsable: Brandon Piedra Chacón			
Id:	RFM04	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los registros de facturas pendientes, siendo el administrador el responsable de estas acciones.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 16, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los registros de facturas pendientes, siendo el administrador el responsable de estas acciones.

Tabla 17

Requerimiento funcional RFM05

Responsable: Brandon Piedra Chacón			
Id:	RFM05	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los registros de categorías, siendo el administrador el responsable de estas acciones.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 17, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los registros de categorías, siendo el administrador el responsable de estas acciones.

Tabla 18

Requerimiento funcional RFM06

Responsable: Brandon Piedra Chacón			
Id:	RFM06	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, eliminar y modificar los registros de detalles de pedido. Esta gestión es responsabilidad del administrador, así como de los usuarios que se han habilitado para realizar estas acciones.			

Fuente:
Dueño del Restaurante
Dependencias:
Ninguna

Tabla 18, Requerimiento funcional para la gestión de agregar, actualizar, eliminar y modificar los registros de detalles de pedido. Esta gestión es responsabilidad del administrador, así como de los usuarios que se han habilitado para realizar estas acciones.

Tabla 19

Requerimiento funcional RFM07

Responsable: Brandon Piedra Chacón			
Id:	RFM07	Prioridad:	Alta
Descripción:			
El sistema permite visualizar la cantidad de datos almacenados. El <i>dashboard</i> muestra el contenido del sistema, siendo el administrador el responsable de su gestión.			
Fuente:			
Dueño del Restaurante			
Dependencias:			
Ninguna			

Tabla 19, Requerimiento funcional para la gestión del sistema permite visualizar la cantidad de datos almacenados. El *dashboard* muestra el contenido del sistema, siendo el administrador el responsable de su gestión.

Requerimientos no funcionales

Tabla 20

Requerimientos no funcionales del sistema

Id	Descripción
RNFM001	Seguridad
RNFM002	Funcionalidad
RNFM003	Accesibilidad

Tabla 29, muestra los requerimientos no funcionales del sistema.

Tabla 21

Requerimiento no funcional RNFM001

Responsable: Brandon Piedra Chacón			
Id:	RNFM001	Prioridad:	Alta
Descripción:			
El sistema valida el acceso de los usuarios que han sido registrados previamente por el administrador, quien les ha asignado un rol, un correo y una contraseña correspondiente.			
Fuente:			
Dueño del Restaurante			

Tabla 30, Requerimiento no funcional, el sistema valida el acceso de los usuarios que han sido registrados previamente por el administrador, quien les ha asignado un rol, un correo y una contraseña correspondiente.

Tabla 22*Requerimiento no funcional RNFM002*

Responsable: Brandon Piedra Chacón			
Id:	RNFM002	Prioridad:	Alta
Descripción:			
El sistema se ha diseñado para que los usuarios puedan utilizarlo y acceder a él de manera sencilla y sin dificultad.			
Fuente:			
Dueño del Restaurante			

Tabla 22, Requerimiento no funcional, el sistema ha sido diseñado para que los usuarios puedan utilizarlo y acceder a él de manera sencilla y sin dificultad.

Tabla 23*Requerimiento no funcional RNMF003*

Responsable: Brandon Piedra Chacón			
Id:	RNFM003	Prioridad:	Alta
Descripción:			
El sistema no tiene un límite de registros, lo que permite acceder a la información necesaria y realizar nuevos registros de forma indefinida.			
Fuente:			
Dueño del Restaurante			

Tabla 23, Requerimiento no funcional, el sistema no tiene un límite de registros, lo que permite acceder a la información necesaria y realizar nuevos registros de forma indefinida.

Modelos UML*Casos de uso*

A continuación, se detallan los actores involucrados en el sistema:

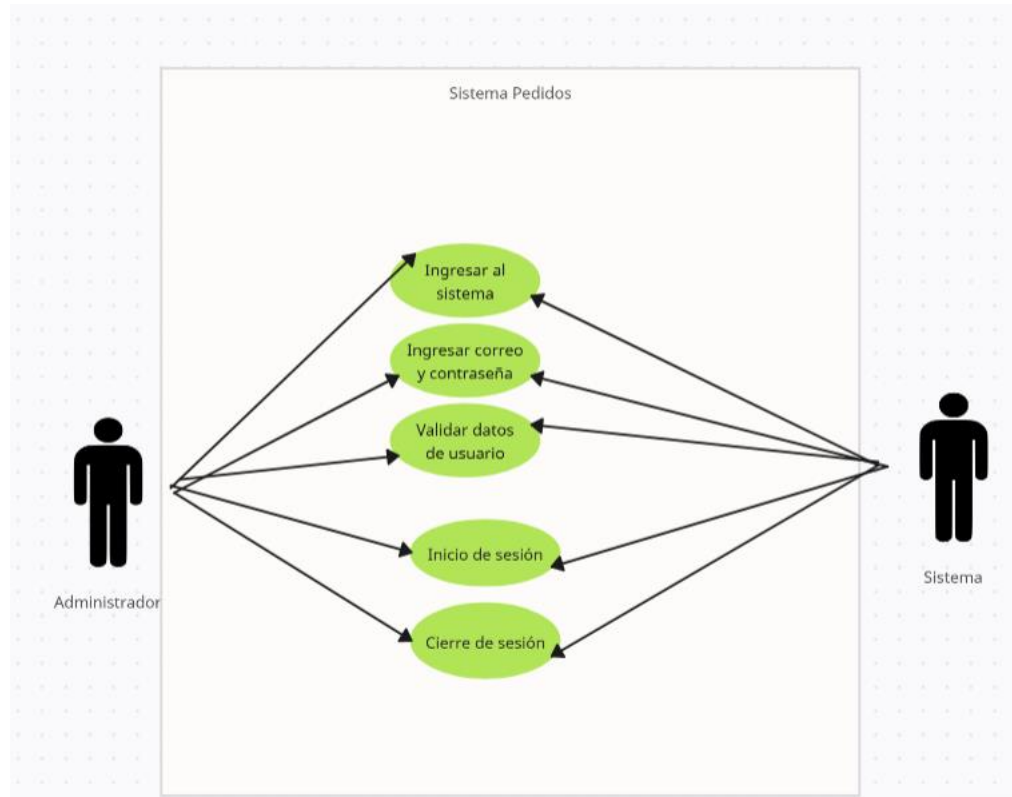
Administrador: es la personal encargado de mantener un registro exacto de los usuarios que podrán utilizar sistema.

Usuario: es la persona que puede agregar, eliminar o modificar información, aunque no tiene los permisos para eliminar usuarios o asignar roles a otros usuarios.

Caso de uso 1. Login

Figura 46

Diagrama de caso de uso - Login



En la Figura 46, el diagrama de caso de uso pertenece al login del sistema para la toma de pedidos del Restaurante Finca La Bandera.

Tabla 24

Caso de uso Login

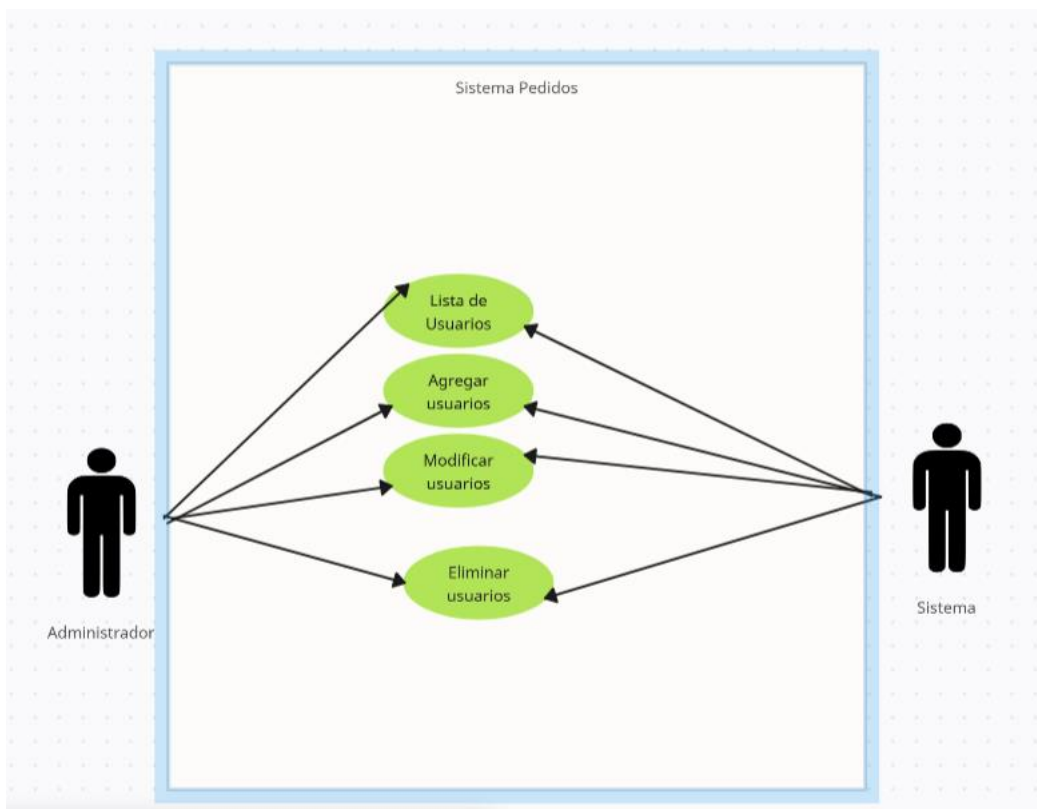
Casos de uso: Login.

Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Ingresar los datos para iniciar de sesión. 2. Cierre de sesión.
Descripción:	<ol style="list-style-type: none"> 1. Acceder al sistema. 2. Ingresar correo y contraseña en el formulario de login. 3. Darle clic al botón de “Inicio Sesión”. 4. Al validar los datos ingresará al sistema, y se visualizará el menú principal del sistema. 5. Al terminar las tareas, para salir de la página debe darle clic al botón de cerrar sesión.
Observaciones:	Los usuarios que no estén autorizados no tendrán acceso al sistema. En caso de que olvide la contraseña, se deberá solicitar al administrador que la restablezca.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Se mostrará un mensaje de error al ingresar la contraseña incorrecta. 2. Se mostrará un mensaje de error al ingresar un usuario erróneo. 3. Se mostrará un mensaje de error al ingresar un usuario no registrado.

Caso de uso 2. Registro de usuario

Figura 47

Diagrama de caso de uso - Registro Usuario



La Figura 51 muestra el diagrama de caso de uso que pertenece al registro de usuarios del sistema para la toma de pedidos en el Restaurante Finca La Bandera.

Tabla 26

Registro Usuario

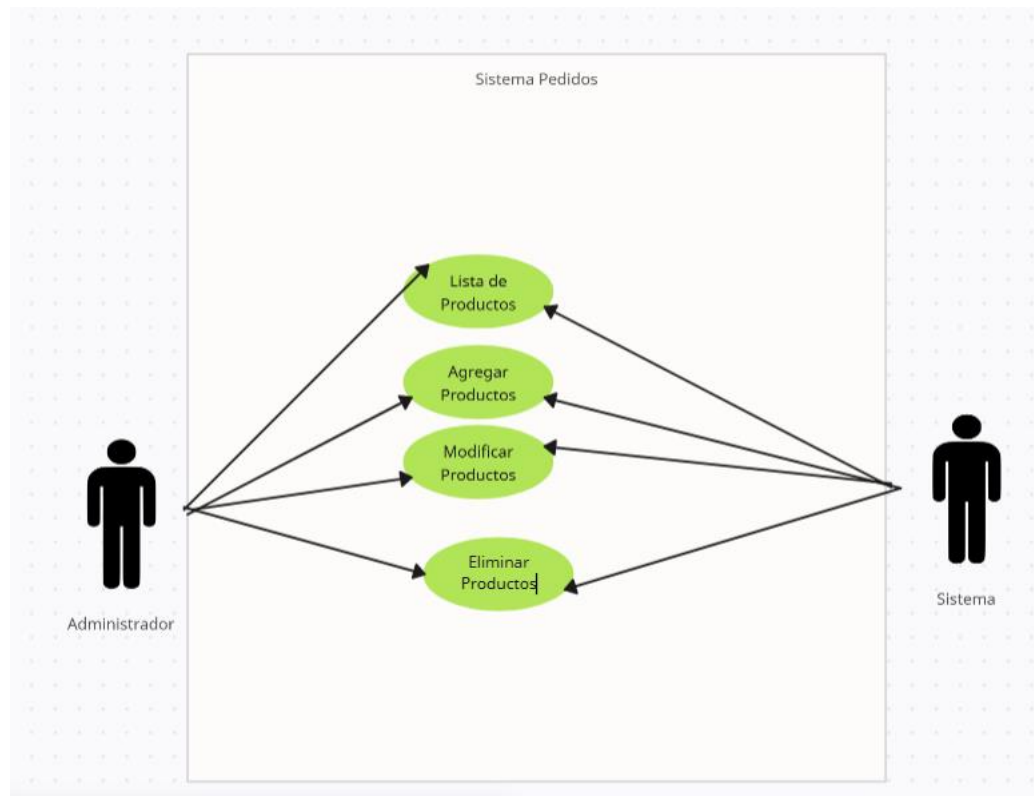
Casos de uso: Registro usuario.	
Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Lista los datos del usuario. 2. Agrega nuevos usuarios con un rol asignado. 3. Modifica los datos del usuario. 4. Elimina los datos del usuario.
Descripción:	<ol style="list-style-type: none"> 1. Acceder a la página de registro. 2. Ingresar los datos requeridos del nuevo usuario. 3. Darle clic al botón de guardar. 4. El sistema valida que los campos estén completos y no exista un usuario registrado con el mismo correo electrónico. 5. El sistema cifra la contraseña y guarda al nuevo usuario en la base de datos.

Observaciones:	Solo los administradores pueden agregar, modificar o eliminar a un usuario.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Si los datos ingresados son incompletos o incorrectos, el sistema muestra un mensaje de error indicando los campos que faltan o están mal. 2. El administrador corrige los datos y vuelve a intentar el registro. 3. Si el correo electrónico ya está registrado, el sistema muestra un mensaje de error. 4. El administrador debe ingresar un nuevo correo electrónico.

Caso de uso 3. Registro de productos

Figura 48

Diagrama de caso de uso - Registro Producto

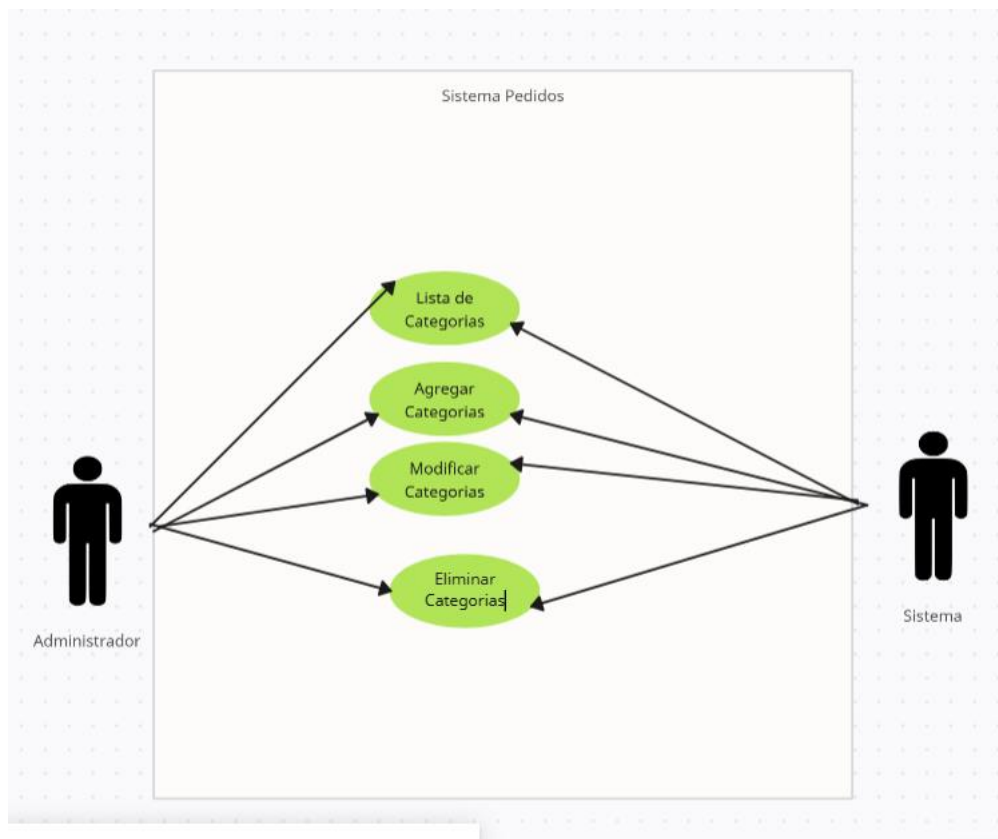


La Figura 48 muestra el diagrama de caso de uso que pertenece al registro de productos del sistema para la toma de pedidos del Restaurante Finca La Bandera.

Tabla 26*Registro Productos*

Casos de uso: Registro productos.	
Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Lista los datos de los productos. 2. Agrega nuevos productos. 3. Modifica los datos de los productos. 4. Elimina los datos de los productos.
Descripción:	<ol style="list-style-type: none"> 1. Acceder a la página de productos. 2. Ingresar los datos requeridos del nuevo producto. 3. Darle clic al botón de guardar. 4. El sistema valida que todos los campos obligatorios estén completos. 5. El sistema guarda el nuevo producto en la base de datos.
Observaciones:	Solo los administradores pueden agregar, modificar o eliminar a un producto.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Si el registro es exitoso, el nuevo producto queda registrado en la base de datos y aparece en la lista de productos.

*Caso de uso 6. Registro de categorías***Figura 49***Diagrama de caso de uso - Registro Categorías*



La Figura 49 describe el diagrama de caso de uso que pertenece al registro de categorías del sistema para toma de pedidos en Restaurante Finca La Bandera.

Tabla 27

Registro Categorías

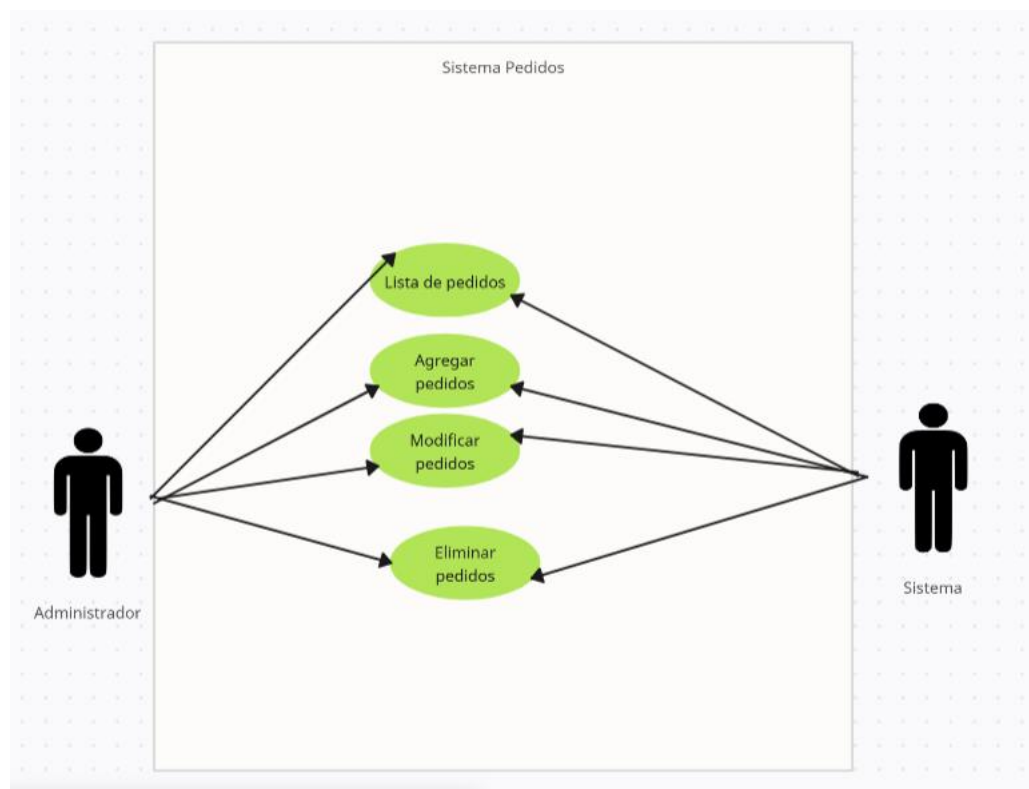
Casos de uso:	Registro categorías.
Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Lista los datos de las categorías. 2. Agrega nuevas categorías. 3. Modifica los datos de las categorías.

	4. Elimina los datos de las categorías.
Descripción:	<ol style="list-style-type: none"> 1. Acceder a la página de categorías. 2. Ingresar los datos requeridos del nuevo categorías 3. Darle clic al botón de guardar. 4. El sistema valida que todos los campos obligatorios estén completos. 5. El sistema guarda la nueva categoría en la base de datos.
Observaciones:	Solo los administradores pueden agregar, modificar o eliminar a unas categorías.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Si el registro es exitoso, la nueva categoría queda registrada en la base de datos y aparece en la lista de categorías.

Caso de uso 7. Registro de pedidos

Figura 50

Diagrama de caso de uso - Registro Pedidos

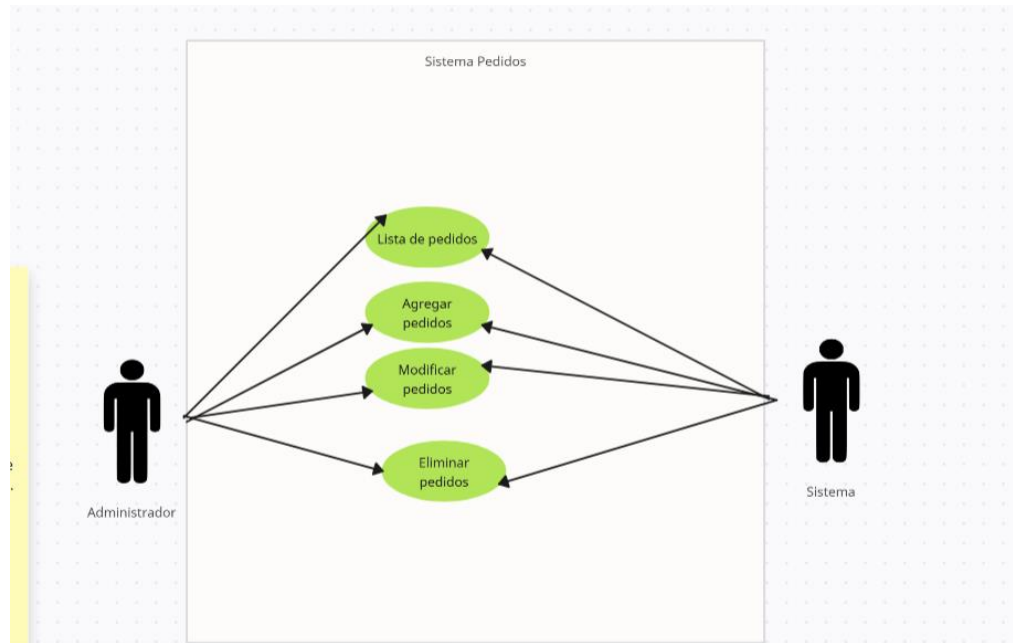


La Figura 50 muestra el diagrama de caso de uso que pertenece al registro de pedidos del sistema para toma de pedidos en Restaurante Finca La Bandera.

Tabla 28*Registro Pedidos*

Casos de uso:	Registros pedidos.
Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Usuario. 3. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Lista los datos de los pedidos. 2. Agrega nuevos pedidos. 3. Modifica los datos de los pedidos. 4. Elimina los datos de los pedidos.
Descripción:	<ol style="list-style-type: none"> 1. Acceder a la página de pedidos. 2. Ingresar los datos requeridos del nuevo pedido. 3. Darle clic al botón de guardar. 4. El sistema valida que todos los campos obligatorios estén completos. 5. El sistema guarda el nuevo pedido en la base de datos.
Observaciones:	Tanto los administradores como los usuarios pueden agregar, modificar o eliminar un pedido.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Si el registro es exitoso, el nuevo pedido queda registrado en la base de datos y aparece en la lista de pedidos.

*Caso de uso 9. Registro de detalles pedidos***Figura 51***Diagrama de caso de uso - Registro de Pedidos*



La Figura 51 muestra el diagrama de caso de uso que pertenece al registro de detalles pedidos del sistema para la toma de pedidos Restaurante Finca La Bandera.

Tabla 29

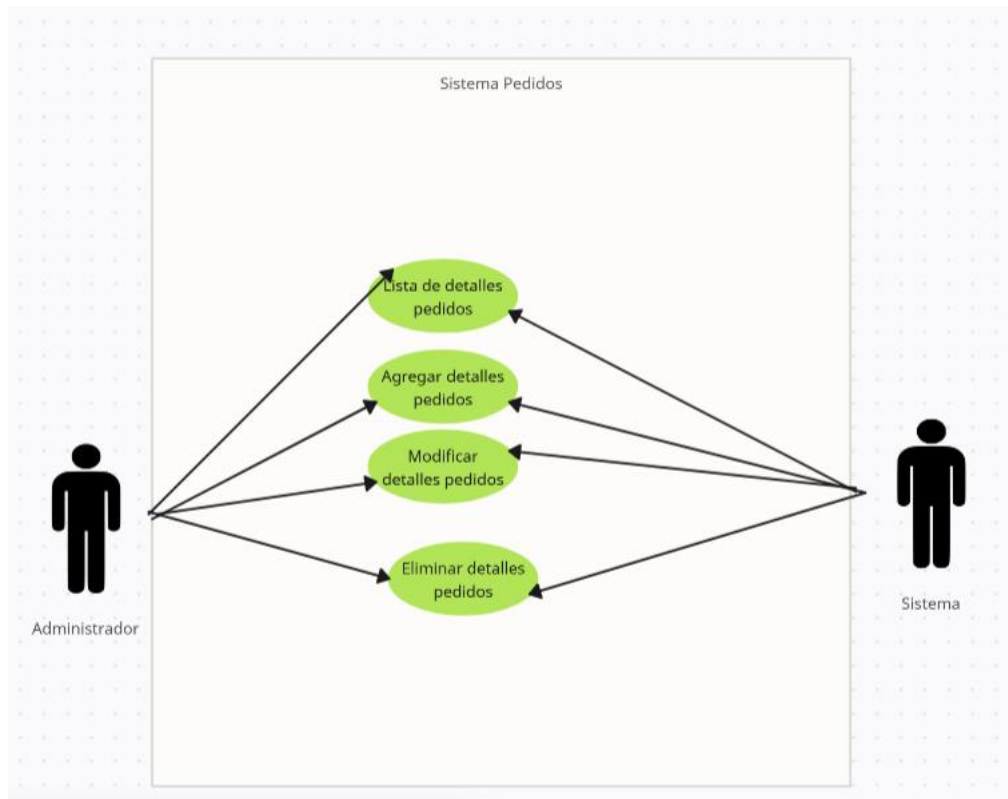
Registro Detalles Pedidos

Casos de uso: Registros detalles pedidos.	
Autora:	Brandon Piedra Chacón
Actores:	<ol style="list-style-type: none"> 1. Administrador. 2. Usuario. 3. Sistema.
Observaciones:	<ol style="list-style-type: none"> 1. Lista los datos de los detalles de pedido. 2. Agrega nuevos detalles de pedido. 3. Modifica los datos de los detalles de pedido. 4. Elimina los datos de los detalles de pedido.
Descripción:	<ol style="list-style-type: none"> 1. Acceder a la página de detalles de pedido. 2. Ingresar los datos requeridos del nuevo detalle de pedido. 3. Darle clic al botón de guardar. 4. El sistema valida que todos los campos obligatorios estén completos. 5. El sistema guarda el nuevo detalle de pedido en la base.
Observaciones:	Tanto los administradores como los usuarios pueden agregar, modificar o eliminar un detalle de pedido.
Mensajes de errores:	<ol style="list-style-type: none"> 1. Si el registro es exitoso, el nuevo detalle de pedido queda registrado en la base de datos y aparece en la lista de detalles de pedido.

Caso de uso 9. Registro de detalles pedidos

Figura 52

Diagrama de caso de uso - Registro Detalles Pedidos



La Figura 52 muestra el diagrama de caso de uso que pertenece al registro de detalles pedidos del sistema para la toma de pedidos en Restaurante Finca La Bandera.

Tabla 30

Registro Detalles Pedidos

Casos de uso:	Registros detalles pedidos.
Autora:	Brandon Piedra Chacón

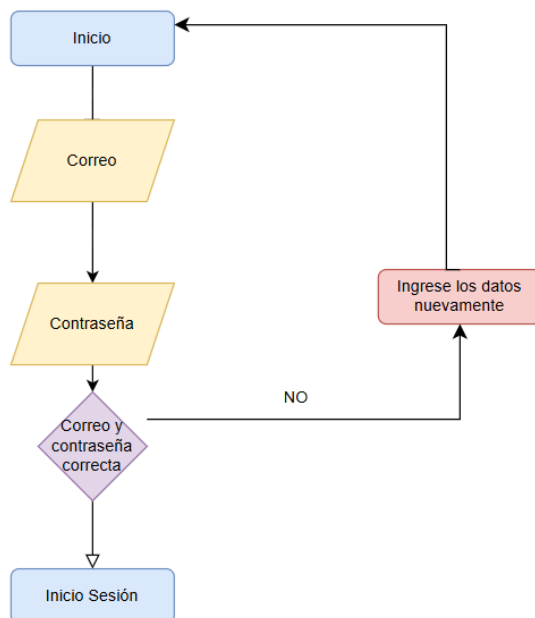
Actores:	<ol style="list-style-type: none"> 4. Administrador. 5. Usuario. 6. Sistema.
Observaciones:	<ol style="list-style-type: none"> 5. Lista los datos de los detalles de pedido. 6. Agrega nuevos detalles de pedido. 7. Modifica los datos de los detalles de pedido. 8. Elimina los datos de los detalles de pedido.
Descripción:	<ol style="list-style-type: none"> 6. Acceder a la página de detalles de pedido. 7. Ingresar los datos requeridos del nuevo detalle de pedido. 8. Darle clic al botón de guardar. 9. El sistema valida que todos los campos obligatorios estén completos. 10. El sistema guarda el nuevo detalle de pedido en la base.
Observaciones:	Tanto los administradores como los usuarios pueden agregar, modificar o eliminar un detalle de pedido.
Mensajes de errores:	<ol style="list-style-type: none"> 2. Si el registro es exitoso, el nuevo detalle de pedido queda registrado en la base de datos y aparece en la lista de detalles de pedido.

Diagramas de flujo

Login

Figura 53

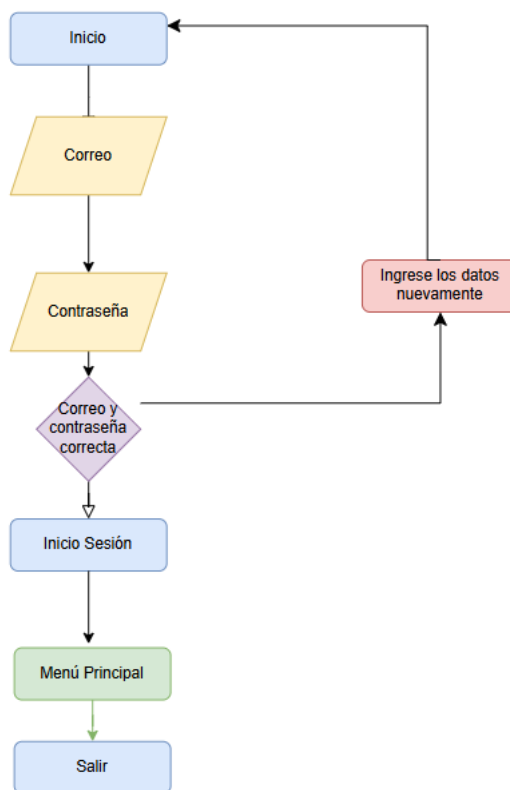
Diagrama de flujo Login



Menú

Figura 54

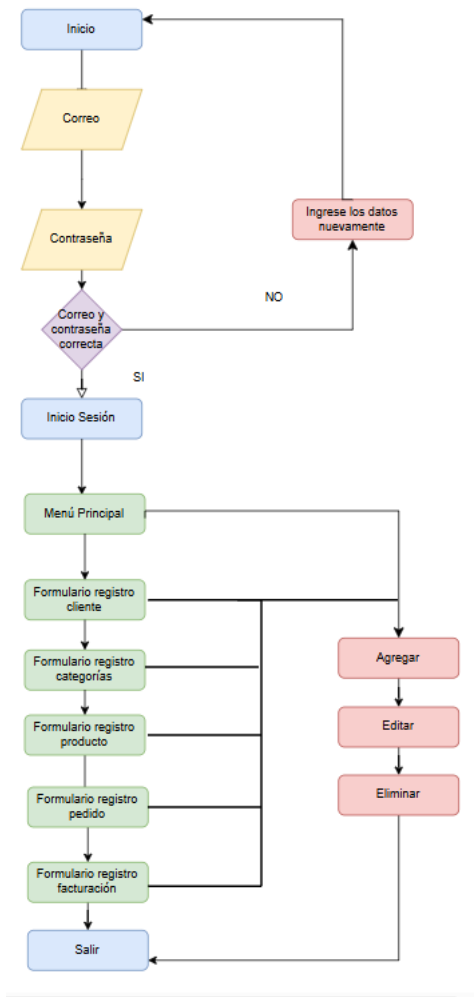
Diagrama de flujo Menú



Mantenimiento

Figura 29

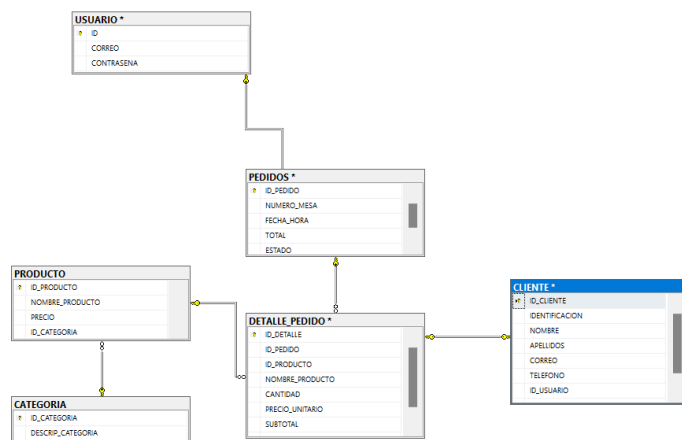
Diagrama de flujo Mantenimiento



Modelo entidad relación

Figura 56

Modelo Relacional



Diccionario de datos

Tabla 31

Tabla Categorías

Columna	Tipo	Nulo
ID_CATEGORIA(Primaria)	int	No
DESCRIP_CATEGORIA	varchar (100)	No

Tabla 32

Tabla Usuario

Columna	Tipo	Nulo
ID(Primaria)	int	No
CORREO	nvarchar (100)	No
CONTRASEÑA	nvarchar (100)	NO

Tabla 33

Tabla Cliente

Columna	Tipo	Nulo
---------	------	------

ID_CLIENTE(Primaria)	int	No
IDENTIFICACION	varchar(20)	
NOMBRE	varchar (100)	No
APELLIDOS	varchar (100)	No
CORREO	varchar (100)	No
TELEFONO	varchar (15)	NO
ID_USUARIO	int	NO
TIPO_IDENTIFICACION	varchar (20)	NO

Tabla 34*Tabla Detalle Pedido*

Columna	Tipo	Nulo
ID_DETALLE(Primaria)	int	No
ID_PEDIDO	int	No
ID_PRODUCTO	int	No
NOMBRE_PRODUCTO	varchar (100)	No
CANTIDAD	int	No
PRECIO_UNITARIO	decimal(10, 2)	NO
SUBTOTAL	decimal(10, 2)	NO
COMENTARIO	nvarchar(255)	Si

Tabla 35*Tabla Pedidos*

Columna	Tipo	Nulo
ID_PEDIDO	int	No
NUMERO_MESA	int	No
FECHA_HORA	datetime	No
TOTAL	decimal(10, 2)	No
ESTADO	varchar(10)	NO

Tabla 36*Tabla Producto*

Columna	Tipo	Nulo
ID_PRODUCTO	int	No
NOMBRE_PRODUCTO	varchar(100)	No
PRECIO	decimal(10, 2)	No
ID_CATEGORIA	int	NO

Desarrollo del sistema

Pantalla del login

Pantalla para logearse

Figura 57

Login



RESTAURANTE FINCA LA BANDERA

Usuario

Contraseña

Acceder

Pantalla de ingreso al sistema

Pantalla que ve el usuario al ingresar al sistema.

Figura 58

Pantalla Principal

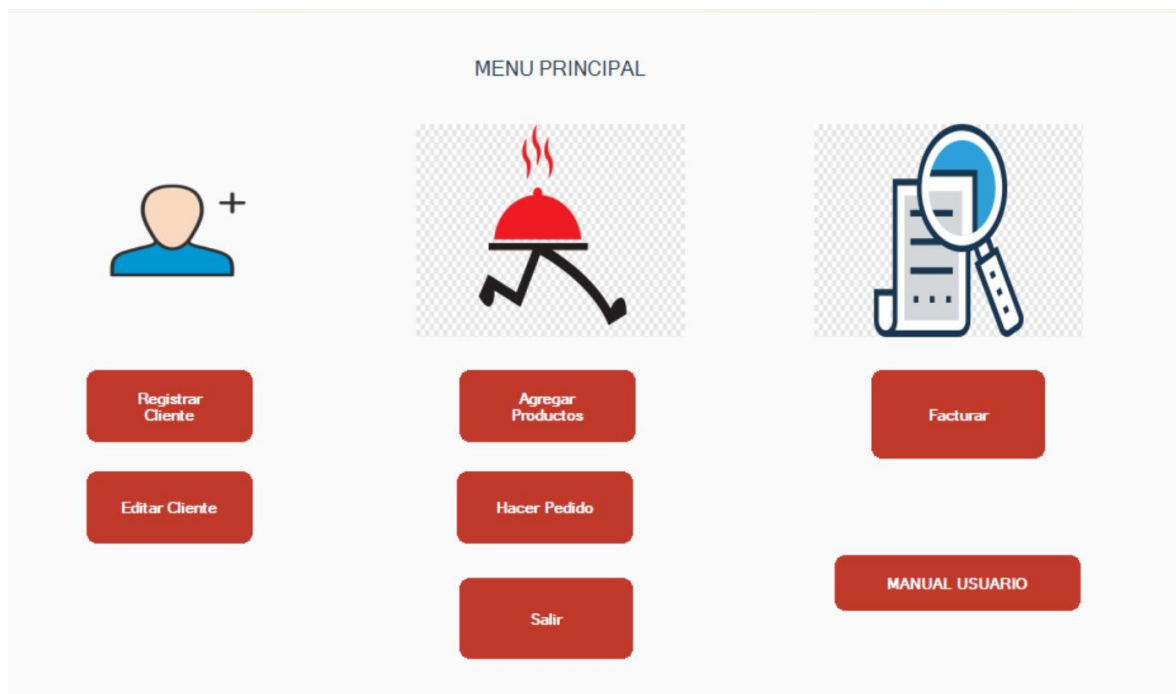


Figura 59

Pantalla Agregar Pedidos

CREAR PEDIDOS

MESA

Categorías

Productos

ID_PRODUCTO	NOMBRE_PRODUCTO	PRECIO
3	ARROZ CON POLLO	4000,00

Cantidad

Comentario

ELIMINAR AGREGAR GUARDAR SALIR

Figura 60*Pantalla Actualizar Pedidos*

CREAR PEDIDOS

MESA

Categorías

Productos

ID_PRODUCTO	NOMBRE_PRODUCTO	PRECIO
3	ARROZ CON POLLO	4000,00

Cantidad

Comentario

ID Producto	Nombre	Cantidad	Precio Unitario	Total	C
3	ARROZ C...	2	4000,00	8000,00	

ELIMINAR AGREGAR GUARDAR SALIR

Pantallas de detalles pedidos

Pantalla donde se muestra el mantenimiento de detalles pedidos.

Figura 61*Pantalla confirmar pedido a cocina*

CREAR PEDIDOS

MESA: Categorías:

Productos: Cantidad: Comentario:

ID_PRODUCTO	NOMBRE_PRODUCTO	PRECIO	Nombre	Cantidad	Precio Unitario	Total	C
2	TROPICAL FRUTAS	1000,00	PAPAZ	1	1000,00	1000,00	
*			TROPICA...	1	1000,00	1000,00	

Ticket Cocina

TICKET COCINA
Pedido #: 25

1x PAPA...
1x TROPICAL FRUTAS

Total: €2 000,00
Fecha: 04/03/2025 21:32

Aceptar

ELIMINAR AGREGAR GUARDAR SALIR

Figura 62

Pantalla Listar Detalles pedidos

FACTURACIÓN

NOMBRE DIGITE IDENTIFICACION MESA

CEDULA

APELLIDO **BUSCAR CLIENTE**

CORREO

TELEFONO

IMPRIMIR FACTURA

ENVIAR A CORREO

VER FACTURA

	ID_PEDID	NUMERO_	NOMBRE_	CANTIDA/	PRECIO_U	SUBTOTA
▶	20	3	ARROZ CON POLLO	2	4000,00	8000,00
	20	3	ARROZ CON POLLO	2	4000,00	8000,00
	22	3	ARROZ CON POLLO	1	4000,00	4000,00
	24	3	TROPICAL FRUTAS	3	1000,00	3000,00
	24	3	PAPAZ	3	1000,00	3000,00
*						

TOTAL:

CANCELA CON:

SU CAMBIO:

PAGAR

Figura 63

Pantalla Agregar comentarios de pedidos

Categorías

Platos Fuertes

Cantidad

Comentario

	Nombre	Cantidad	Precio Unitario	Total	Comentario
▶	ARROZ C...	1	4000,00	4000,00	SIN ENSA...
*					

GUARDAR

SALIR

Pantallas de clientes

Pantalla donde se muestra el mantenimiento de clientes.

Figura 63

Pantalla registrar clientes

REGISTRO DE CLIENTES

TIPO IDENTIFICACIÓN	<input type="text" value="Cédula"/>	<input type="text"/>
Cédula	<input type="text"/>	<input type="text"/>
NOMBRE	<input type="text"/>	<input type="text"/>
PRIMER APELLIDO	<input type="text"/>	<input type="button" value="BUSCAR"/>
SEGUNDO APELLIDO	<input type="text"/>	
CORREO ELECTRÓNICO	<input type="text"/>	
TELÉFONO CELULAR	<input type="text"/>	
<input type="button" value="REGISTRAR CLIENTE"/>		

Figura 64

Pantalla editar clientes

TIPO IDENTIFICACION	<input type="text" value="Cédula"/>	<input type="button" value="BUSCAR"/>
Cédula:	<input type="text" value="1-1895-0668"/>	<input type="button" value="ACTUALIZAR DATOS"/>
NOMBRE	<input type="text" value="BRANDON VIANNEY"/>	
APELLIDOS	<input type="text" value="PIEDRA CHACON"/>	
CORREO	<input type="text" value="FACTURASE"/>	
TELEFONO	<input type="text" value="86253262"/>	

✕

Datos actualizados correctamente.

Pantallas de productos

Pantalla donde se muestra el mantenimiento de productos.

Figura 65

Pantalla Listar Productos



The screenshot displays a table with five columns: ID_PRODUC, NOMBRE_P, PRECIO, and DESCRIP_C. The first row is highlighted in blue. Below the table are two red buttons labeled 'EDITAR' and 'ELIMINAR'.

	ID_PRODUC	NOMBRE_P	PRECIO	DESCRIP_C
▶	2	TROPICA...	1000,00	Bebidas
	3	ARROZ C...	4000,00	Platos Fue...
	4	PAPAZ	1000,00	Comida R...
	5	TRUCHA	5500,00	Platos Fue...
*				

Figura 66

Pantalla Agregar Productos



Figura 67

Pantalla Actualizar Productos



Pantallas de facturación

Pantalla donde se muestra la facturación.

Figura 68

FACTURACIÓN

NOMBRE: DIGITE IDENTIFICACION:

CEDULA:

APELLIDO:

CORREO:

TELEFONO:

MESA:

TOTAL:

CANCELA CON:

SU CAMBIO:

ID_PEDID	NUMERO	NOMBRE			
20	3	ARROZ CON POLLO			
20	3	ARROZ CON POLLO	2	4000,00	8000,00
22	3	ARROZ CON POLLO	1	4000,00	4000,00
24	3	TROPICAL FRUTAS	3	1000,00	3000,00
24	3	PAPAZ	3	1000,00	3000,00

×

Éxito

Factura marcada como pagada.

Referencias

¿Qué es la Programación orientada a objetos? | EDteam. (s/f). Recuperado el 20 de agosto de 2024, de <https://ed.team/comunidad/que-es-la-programacion-orientada-a-objetos>

Editor vs IDE | EDteam. (s/f). Recuperado el 26 de agosto de 2024, de <https://ed.team/comunidad/editor-vs-ide>

3.1 Requerimientos funcionales y no funcionales. (s/f). Recuperado el 22 de agosto de 2024, de <https://view.genially.com/659329ab9c403800147f96f2/interactive-content-31-requerimientos-funcionales-y-no-funcionales>

8 ejemplos de diagramas de casos de uso para inspirarse | Boardmix. (s/f). Recuperado el 22 de agosto de 2024, de <https://boardmix.com/es/articles/use-case-examples/>

- (25) *Explorando las Similitudes entre Metodologías Ágiles y Desarrollo Tradicional.* | LinkedIn. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.linkedin.com/pulse/explorando-las-similitudes-entre-metodolog%C3%ADas-%C3%A1giles-y-perea-r/>
- ¡Aprende a distinguir entre fuentes primarias, secundarias y terciarias!* (s/f). Recuperado el 25 de septiembre de 2024, de <https://tesisdeceroa100.com/aprende-a-distinguir-entre-fuentes-primarias-secundarias-y-terciarias/>
- Arquitectura de Componentes - DEV Community.* (s/f). Recuperado el 22 de agosto de 2024, de <https://dev.to/vanessamarely/arquitectura-de-componentes-283p>
- Arquitectura de datos: ¿qué es? Importancia, tipos y usos.* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.astera.com/es/type/blog/data-architecture/>
- Arquitectura de software: ¿Qué es y qué tipos hay?* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay>
- Bases de datos y su importancia en las empresas-Gestión de base de datos.* (s/f). Recuperado el 26 de agosto de 2024, de <https://visualizamarketingonline.com.ar/blog/bases-datos-importancia-las-empresas/>
- Casos Uso Farmacia | PDF | Farmacia | Caso de uso.* (s/f). Recuperado el 26 de septiembre de 2024, de <https://es.scribd.com/doc/64216512/Casos-uso-farmacia>
- ¿Cómo aprender a programar desde cero?* | EDteam. (s/f). Recuperado el 22 de agosto de 2024, de <https://ed.team/blog/como-aprender-programar-desde-cero>
- ¿Cómo funciona un sistema de inventario?* | Defontana Chile. (s/f). Recuperado el 4 de agosto de 2024, de <https://www.defontana.com/blog/cl/como-funciona-un-sistema-de-inventario>
- Computacionales, S. (s/f). *UNIVERSIDAD ESTATAL DE MILAGRO FACULTAD CIENCIAS E INGENIERÍA TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN.*
- ¿Qué son los paradigmas de programación?* (s/f). Ed.team. Recuperado el 20 de febrero de 2025, de <https://ed.team/blog/que-son-los-paradigmas-de-programacion-f57de068-6426-4332-b5dd-dbdcff1aaaec>
- Perfil, V. (2017, junio 13). 06/03/17 ⇒Evolución de los lenguajes de programación⇐. *Blogspot.com.* <https://lpportafolioambrizg.blogspot.com/2017/06/evolucion-de-los-lenguajes-de.html>
- Introducción a la programación - Mapa conceptual.* (s/f). Prezi.com. Recuperado el 20 de febrero de 2025, de <https://prezi.com/p/fys0jjkc0qff/introduccion-a-la-programacion-mapa-conceptual/>
- Los 14 lenguajes de programación más populares, según un estudio entre 100.000 desarrolladores.* (2021, mayo 12). Devsoft. <https://devsoft.io/los-14-lenguajes-de-programacion-mas-populares-segun-un-estudio-entre-100-000-desarrolladores/>

Los lenguajes de programación. (s/f). Espino.Info. Recuperado el 20 de febrero de 2025, de <https://desarrollo.espino.info/la-alegria-de-programar/los-lenguajes-de-programacion.html>

Lenguajes tipados vs no tipados. (s/f). Ed.team. Recuperado el 20 de febrero de 2025, de <https://ed.team/comunidad/lenguajes-tipados-vs-no-tipados>

Tipado dinámico. (s/f). Interactivechaos.com. Recuperado el 22 de febrero de 2025, de <https://interactivechaos.com/es/manual/tutorial-de-python/tipado-dinamico>

Compilados/ Interpretados. (2017, febrero 15). Introducción a la ingeniería en sistemas. <https://rafaramoneblog.wordpress.com/2017/02/15/compiladosinterpretados/>

Mapa Conceptual Sobre Paradigmas de Programación. (s/f). Scribd. Recuperado el 22 de febrero de 2025, de <https://es.scribd.com/document/652778206/Mapa-Conceptual-sobre-Paradigmas-de-Programacion>

TOMi.digital - PARADIGMAS DE LA PROGRAMACIÓN 2DO BGU. (s/f). TOMi.digital. Recuperado el 22 de febrero de 2025, de https://tomi.digital/es/es/60415/paradigmas-de-la-programacion-2do-bgu?utm_source=google&utm_medium=seo

Desarrollo de Aplicaciones con Software Orientado a Objetos en C. (s/f). desarrollodesoftware. Recuperado el 22 de febrero de 2025, de <https://desarrollodesoftware.dev/c-software-para-el-desarrollo-de-aplicaciones-orientada-a-objetos>

Diaz, L. N. A. (s/f). *Procedimientos y Funciones I - Programación en Visual Basic Net con Visual Studio 2013 (21-25).* www.incanatoit.com - Desarrollando Software. Recuperado el 22 de febrero de 2025, de <https://www.incanatoit.com/2017/03/procedimientos-y-funciones-i.html>

LOOPA. (2016, junio 2). *Paradigmas de Programación: Programación Imperativa y Programación Declarativa.* Medium. <https://medium.com/@Loopa/paradigmas-de-programaci%C3%B3n-programaci%C3%B3n-imperativa-y-programaci%C3%B3n-declarativa-4c4a4182fd87>

¿Qué es un sistema de pedidos y cómo funciona? (2024, mayo 6). Tecseris. <https://tecseris.com/blog/que-es-un-sistema-de-pedidos-y-como-funciona/>

Sistema de pedidos online gratis. (2024, octubre 17). ApperStreet. <https://apperstreet.com/sistema-de-pedidos-online-gratis/>

GamerPC. (2021, mayo 19). *¿Qué son los sistemas de información en el mundo de la informática?* GamerPC. <https://gamerpc.es/sistemas-informacion>

Ciclo de Vida de un Sistema de Información. (s/f). Tecnologias-informacion.com. Recuperado el 25 de febrero de 2025, de <https://www.tecnologias-informacion.com/ciclodevida.html>

CLASIFICACIÓN Y TIPOS DE REQUERIMIENTOS. (2014, agosto 26).

ADMINISTRACIÓN DE REQUERIMIENTOS.

<https://administracionderequerimientos.wordpress.com/2014/08/26/clasificacion-y-tipos-de-requerimientos/>

Diferentes diagramas UML – Propósito y uso. (s/f). Edrawsoft.com. Recuperado el 25 de febrero de 2025, de <https://www.edrawsoft.com/es/uml-introduction.html>

Alam, M. (2022, marzo 14). *¿Qué es un diagrama UML? Definición, casos de uso y cómo hacer.* IdeaScale. <https://ideascale.com/es/blogs/uml-diagram-definition/>

12 conceptos básicos de la base de datos. (2020, noviembre 13). Dream Technology. <https://tecnologiasdelainformacion.over-blog.com/2020/11/12-conceptos-basicos-de-la-base-de-datos.html>

Lloves, F. (2021, octubre 26). *Funciones en MySQL.* Somos PNT - Desarrollamos Software. <https://somospnt.com/blog/230-funciones-en-mysql>

Base de datos Open Source. (2022, marzo 2). ITD Consulting. <https://itdconsulting.com/base-de-datos/mysql-maria-db/base-de-datos-open-source/>

Metodologías de desarrollo de software: ¿qué son? (s/f). Santanderopenacademy.com. Recuperado el 25 de febrero de 2025, de <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>

¿Qué es un desarrollador web y cuáles son sus principales funciones? (2023, enero 13). Hackio.com; Hackio by thePower. <https://www.hackio.com/blog/que-es-un-desarrollador-web>

mobileLIVE. (2023, junio 7). *Backend for frontend basics: A comprehensive guide - mobileLIVE.* Medium. <https://mobilelive.medium.com/backend-for-frontend-basics-a-comprehensive-guide-37768062e55a>

Editor vs IDE. (s/f). Ed.team. Recuperado el 26 de febrero de 2025, de <https://ed.team/comunidad/editor-vs-ide>

¿En qué consiste la programación en Visual Basic? (2024, noviembre 6). Euroinnova International Online Education. <https://www.euroinnova.com/blog/programacion-visual-basic>

Visual Basic. (2019, febrero 26). Lenguajes de Programación más populares. <https://lenguajesdeprogramacionmaspopulares.wordpress.com/visual-basic/>

(S/f). Frogamesformacion.com. Recuperado el 26 de febrero de 2025, de <https://cursos.frogamesformacion.com/pages/blog/visual-basic-net>

Tutorial online - visual basic .NET. (s/f). Bibliadelprogramador.com. Recuperado el 26 de febrero de 2025, de <https://www.bibliadelprogramador.com/2013/08/tutorial-online-visual-basic-net.html>

Perfil, V. T. mi. (s/f). *METODOLOGIA DE LA INVESTIGACION*. Blogspot.com. Recuperado el 27 de febrero de 2025, de <https://armandoperezeste.blogspot.com/2015/11/marco-metodologico.html>

Compilado por: Ing Pablo Emilio López Marquina. (2016, noviembre 21). ¿CUÁL ENFOQUE DARLE A UNA INVESTIGACIÓN? *misitio*. <https://pablolopezsistemas.wixsite.com/misitio/single-post/2016/11/21/-cu%C3%A1l-enfoque-darle-a-una-investigaci%C3%B3n>

Medina, R. (s/f). *Enfoque Cuantitativo*. GoConqr. Recuperado el 27 de febrero de 2025, de <https://www.goconqr.com/diapositiva/9290938/enfoque-cuantitativo>

(S/f). Pressbooks.pub. Recuperado el 27 de febrero de 2025, de <https://scienceetbiencommun.pressbooks.pub/evalsalud/chapter/revuesmixtes/>

3.3.1 *Fuentes de información*. (s/f). Edu.mx. Recuperado el 27 de febrero de 2025, de https://maestrias.clavijero.edu.mx/cursos/MCDEMS/T6/618SI/modulo2/contenidos/331_fuentes_de_informacin.html

Ingenieroduran, P. (s/f). *CUADRO DE OPERACIONALIZACIÓN DE VARIABLES – ASESORÍA*. Automatisoft.pe. Recuperado el 27 de febrero de 2025, de <https://www.darwinduran.automatisoft.pe/2018/03/29/cuadro-de-operacionalizacion-de-variables/>

Casos de Uso y el Proceso del Requerimiento. (s/f). Slideplayer.es. Recuperado el 27 de febrero de 2025, de <https://slideplayer.es/slide/1685876/>

Completo, V. mi P. (s/f). *INFORMÁTICA IV (DIAGRAMAS STAR UML)*. Blogspot.com. Recuperado el 27 de febrero de 2025, de <https://octaviotorres25.blogspot.com/p/ejemplo-de.html>

Aldrin Velázquez. (2023). Questionpro. Investigación de mercado. Estudio de factibilidad del mercado: Qué es, ventajas y cómo realizarlo. <https://www.questionpro.com/blog/es/factibilidad-delmercado/#:~:text=Un%20estudio%20de%20factibilidad%20del%20mercado%20es%20una%20herramienta%20que,realizar%20o%20no%20la%20inversi%C3%B3n>. Alejandro Núñez. (2022). CORPORATIVORIBA. CONCEPTOS-ECONOMICOS | F. ¿Qué es la Factibilidad Operativa? Explicado de forma sencilla. https://corporativoriba.com/factibilidad-operativa/#%C2%BFQue_es_la_factibilidad_operativa Arias-García, J. (2013). Quefilo, plataforma de pedidos de comida por internet, Zona Norte, Avantica San Carlos. <https://repositoriotec.tec.ac.cr/handle/2238/3315> Atlassian. (2023) Desarrollo de software. Agile. El consejeroagil. <https://www.atlassian.com/es/agile> Daniella Tudela (2023). Geal.cloud. Agilidad. <https://blog.gael.cloud/metodologias-agiles-ventajas/#:~:text=Existen%20varios%20tipos%20de%20metodolog%C3%ADas,Lean%20%20Kanban%20y%20Design%20Sprint>. Guilbert, B. H. A. (2023). Estudio de factibilidad para el desarrollo de un sistema informático en la gestión administrativa de venta y pedidos

- a los proveedores en el Restaurant «El Rincón Vegetariano» de la ciudad de Babahoyo. <http://dspace.utb.edu.ec/handle/49000/14782> Gregorio, A. E. V. (2023). Programas informáticos de gestión de restaurantes: análisis de un software de gestión de reservas y pedidos. idUS - Depósito de Investigación Universidad de Sevilla. <https://idus.us.es/handle/11441/148296> Gestión de alimentos y bebidas para hoteles, bares y restaurantes. (s. f.). Google Books. 75 https://books.google.com.mx/books?hl=es&lr=&id=FpA0qUy1NjcC&oi=fnd&pg=PA1&dq=Origenes+de+los+restaurantes&ots=fg1_Tyn9dp&sig=wnT4_wS0CuZnnnsWm0Ylw6Ir8Uo#v=onepage&q=Origenes%20de%20los%20restaurantes&f=false Patricio, J. C. R. (2023, 1 julio). Desarrollo de una aplicación móvil para la gestión de pedidos para la cadena de restaurante “Restaurante Familiar”. <http://dspace.istvidanueva.edu.ec/handle/123456789/338> Julia Martins (2022). GESTIÓN DE PROYECTOS. ¿Qué es la metodología Kanban y cómo funciona? <https://asana.com/es/resources/what-is-kanban> Martin Barten (2023). Sistema de pedidos de restaurantes: consejos para los restauradores a la hora de elegir uno. Revfine/tecnología. <https://www.revfine.com/es/sistema-de-pedidos-de-restaurante/>
- Sepúlveda, C. A. R. (2022, julio 7). *Planes de emergencia y gestión del riesgo de desastres ¿ha llegado su fin?* LinkedIn.com. <https://es.linkedin.com/pulse/planes-de-emergencia-y-gesti%C3%B3n-del-riesgo-desastres-cristian-andr%C3%A9s>
- Laborales, R. (2023, enero 16). Matriz de Riesgos en la Empresa. *Riesgoslaborales.info*. <https://riesgoslaborales.info/matriz-de-riesgos/>
- Rodrigues, N. (2022, octubre 25). *Qué es la factibilidad de un proyecto, cómo se estudia y ejemplifica*. Hubspot.es. <https://blog.hubspot.es/sales/que-es-factibilidad>
- Comunicación, I. Y. (2023). *Propuesta de mejora en el sistema de gestión de inventarios de roles en bodega central de suministro, por medio del uso de herramientas de la calidad que permitan apoyar la reducción de costos operativos del proceso, en departamento y área seleccionada en Ingenio Taboga, año 2022*. <https://repositorio.ulatina.ac.cr/handle/20.500.12411/2321>
- Conceptos básicos de HTML - Aprende desarrollo web | MDN*. (s/f). Recuperado el 26 de agosto de 2024, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics
- Conditionals in Thymeleaf | Baeldung*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.baeldung.com/spring-thymeleaf-conditionals>
- Corella-Parra, L. M., Olea-Miranda, J., Corella-Parra, L. M., & Olea-Miranda, J. (2023). Desarrollo de un sistema de control de inventario para una empresa comercializadora de sistemas de riego. *Ingeniería, investigación y tecnología*, 24(1), 1–10. <https://doi.org/10.22201/FI.25940732E.2023.24.1.006>

- Crear un sitio web html usando html, css y javascript.* (s/f). Recuperado el 26 de agosto de 2024, de https://de.fiverr.com/umar_daraz_/create-html-website-using-html-css-and-javascript
- Cuadro de Operacionalización de La Variable | PDF | Logística | Inventario.* (s/f). Recuperado el 25 de septiembre de 2024, de <https://es.scribd.com/doc/127011876/CUADRO-DE-OPERACIONALIZACION-DE-LA-VARIABLE-xlsx>
- Curso: MySQL (Comienza gratis) | EDteam.* (s/f). Recuperado el 26 de agosto de 2024, de <https://ed.team/cursos/mysql>
- Definición de casos de uso - Documentación de IBM.* (s/f-a). Recuperado el 22 de agosto de 2024, de <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Definición de casos de uso - Documentación de IBM.* (s/f-b). Recuperado el 26 de septiembre de 2024, de <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Desarrollo de un software web para la generación de planes de gestión de riesgos de software.* (s/f). Recuperado el 27 de septiembre de 2024, de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000300135
- Diagrama de casos de usos | PPT.* (s/f). Recuperado el 26 de septiembre de 2024, de <https://es.slideshare.net/slideshow/diagrama-de-casos-de-usos/13589232#1>
- Diagrama de componentes UML: Qué es y cómo hacerlo | Miro.* (s/f). Recuperado el 22 de agosto de 2024, de <https://miro.com/es/diagrama/que-es-diagrama-componentes-uml/>
- Diseño Integral - FasterCapital.* (s/f). Recuperado el 22 de agosto de 2024, de <https://fastercapital.com/es/palabra-clave/dise%C3%B1o-integral.html>
- Dos siglos de historia del lenguaje de programación.* (s/f). Recuperado el 22 de julio de 2024, de <https://www.revistaonoff.es/lenguaje-de-programacion-una-historia-con-mas-de-200-anos/>
- El ciclo SDLC en 7 fases - Viewnext.* (s/f). Recuperado el 18 de septiembre de 2024, de <https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>
- el concepto del sistema de control de inventario, el gerente profesional y el trabajador están verificando el suministro de bienes y existencias, la gestión de inventario con la demanda de bienes 12494550 Vector en Vecteezy.* (s/f). Recuperado el 4 de agosto de 2024, de <https://es.vecteezy.com/arte-vectorial/12494550-el-concepto-del-sistema-de-control-de-inventario-el-gerente-profesional-y-el-trabajador-estan-verificando-el-suministro-de-bienes-y-existencias-la-gestion-de-inventario-con-la-demanda-de-bienes>
- El portal de la tesis.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://recursos.uco.mx/tesis/investigacion.php>

- En, M., De Proyectos, G., Aguilar, A. C., Alajuela, P., & Del, J. (2021). *Plan de gestión del proyecto Inventario Forestal Nacional de Costa Rica*.
<https://repositoriotec.tec.ac.cr/handle/2238/13426>
- Estrategias para gestionar y priorizar requisitos de usuario en proyectos UX*. (s/f). Recuperado el 6 de octubre de 2024, de <https://keepcoding.io/blog/requisitos-de-usuario-en-proyectos-ux/>
- Estudio de factibilidad - Qué es y qué tipos hay*. (s/f). Recuperado el 27 de septiembre de 2024, de <https://economipedia.com/definiciones/estudio-de-factibilidad.html>
- Explorar las características del editor de código - Visual Studio (Windows) | Microsoft Learn*. (s/f). Recuperado el 26 de agosto de 2024, de <https://learn.microsoft.com/es-es/visualstudio/ide/writing-code-in-the-code-and-text-editor?view=vs-2022>
- Fundamentos de JavaScript - Aprende desarrollo web | MDN*. (s/f). Recuperado el 26 de agosto de 2024, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- García-Pacheco, M. C., & Andrés-Laz, E. M. S. (2021). DISEÑO DE UN SISTEMA DE GESTIÓN POR PROCESOS PARA EL MANEJO DE INVENTARIOS. CASO: FERRETERÍA QUIROZ. *REVISTA CIENTÍFICA MULTIDISCIPLINARIA ARBITRADA YACHASUN - ISSN: 2697-3456*, 5(9 Ed. esp.), 180–203.
<https://doi.org/10.46296/YC.V5I9EDESPSOCT.0118>
- Historia de Bases de Datos: Evolución y Orígenes - jhonmosquera.com*. (s/f). Recuperado el 26 de agosto de 2024, de <https://jhonmosquera.com/bases-de-datos-historia/>
- Historia de los lenguajes de programación - Administración de Sistemas*. (s/f). Recuperado el 19 de julio de 2024, de <https://administraciondesistemas.com/historia-de-los-lenguajes-de-programacion/>
- IDE: ¿qué es un entorno de desarrollo integrado?* (s/f). Recuperado el 26 de agosto de 2024, de <https://datascientest.com/es/ide>
- IDEs: Qué son, para qué sirven y los más populares - Blog de SW Hosting*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.swhosting.com/es/blog/ides-que-son-para-que-sirven-y-los-mas-populares>
- Introducción a la programación funcional con Java | by Johan Del Valle | Medium*. (s/f). Recuperado el 20 de agosto de 2024, de <https://johandvv.medium.com/introducci%C3%B3n-a-la-programaci%C3%B3n-funcional-con-java-89c58e0972bc>
- Investigación cualitativa | QuestionPro*. (s/f). Recuperado el 24 de septiembre de 2024, de <https://www.questionpro.com/es/investigacion-cualitativa.html>

- Java: ¿Qué es y para qué sirve?* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.godaddy.com/resources/latam/stories/java-que-es-y-para-que-sirve>
- Las mejores bases de datos open source - Codemotion.* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.codemotion.com/magazine/es/backend-es/las-5-mejores-bases-de-datos-open-source/>
- Lenguajes de Alto nivel vs de Bajo nivel | EDteam.* (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/lenguajes-de-alto-nivel-vs-de-bajo-nivel-b8806b09-73ac-4d83-9047-b487dae2cab2>
- Lenguajes de bajo nivel: aprende qué y cuáles son.* (s/f). Recuperado el 19 de agosto de 2024, de <https://keepcoding.io/blog/que-son-lenguajes-de-bajo-nivel-cuales-hay/>
- Lenguajes de programación: una breve introducción contextualizada | Alura Cursos Online.* (s/f). Recuperado el 19 de agosto de 2024, de <https://www.aluracursos.com/blog/lenguajes-de-programacion>
- Los 17 mejores editores de código.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.hostinger.es/tutoriales/editores-de-codigo>
- Los enfoques cuantitativo y cualitativo en la investigación | PPT.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://es.slideshare.net/slideshow/los-enfoques-cuantitativo-y-cualitativo-en-la-investigacin/30112318#6>
- Los sistemas de información de una empresa.* (s/f). Recuperado el 18 de septiembre de 2024, de <https://informatecdigital.com/productividad/los-sistemas-de-informacion-de-una-empresa/>
- Mapa conceptual fundamentos de programación by Christopher Diego Jiménez Almaraz on Prezi.* (s/f). Recuperado el 19 de agosto de 2024, de <https://prezi.com/bqomkgmholfj/mapa-conceptual-fundamentos-de-programacion/>
- Matriz de riesgo: Importancia y ejemplos | SafetyCulture.* (s/f). Recuperado el 27 de septiembre de 2024, de <https://safetyculture.com/es/temas/evaluacion-de-riesgos/matriz-de-riesgo/>
- Matriz de riesgos: cómo evaluar los riesgos para lograr el éxito del proyecto [2024] • Asana.* (s/f). Recuperado el 27 de septiembre de 2024, de <https://asana.com/es/resources/risk-matrix-template>
- Método UEPS (last in, first out): usos y ejemplos.* (s/f). Recuperado el 4 de agosto de 2024, de <https://www.mecalux.com.mx/blog/metodo-ueps>
- Metodología de la Investigación: MARCO METODOLOGICO.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://uftmetodologia.blogspot.com/2019/09/marco-metodologico.html>
- Metodología en cascada: Ventajas e inconvenientes | SafetyCulture.* (s/f). Recuperado el 26 de agosto de 2024, de <https://safetyculture.com/es/temas/metodologia-en-cascada/>

- Metodología Kanban: en qué consiste y cómo utilizarla | APD.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.apd.es/metodologia-kanban/>
- Perfil y Salario de un Perfil de Desarrollador Web en México en 2024 - Hireline México.* (s/f). Recuperado el 19 de septiembre de 2024, de <https://hireline.io/mx/enciclopedia-de-perfiles-de-tecnologia/desarrollador-web>
- Plan de pruebas de software: 8 pasos para realizarlas - icaria Technology.* (s/f). Recuperado el 26 de septiembre de 2024, de <https://icariatechnology.com/plan-de-pruebas-de-software/>
- Programación imperativa: ventajas y desventajas del paradigma - IONOS.* (s/f). Recuperado el 20 de agosto de 2024, de <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/>
- Prototipo – el papel del prototipado en el proceso de desarrollo de productos - Copymate.* (s/f). Recuperado el 26 de agosto de 2024, de <https://copymate.app/es/blog/multi/prototipo-el-papel-del-prototipado-en-el-proceso-de-desarrollo-de-productos/>
- Publicaes.* (s/f). Recuperado el 24 de septiembre de 2024, de https://www.publicaes.com/?fbclid=IwY2xjawFhs5RleHRuA2FlbQIxMAABHW9ksVxMxQ0Z0RqdzjYUF9_XfQUV-CstGjDKRgpQiWr2oMGKDco-o_RNJQ_aem__o-l3yxTNupy11q76mSoZw
- ¿Qué es CSS y para qué sirve? - Webempresa.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.webempresa.com/blog/que-es-css-y-para-que-sirve.html>
- ¿Qué es el desarrollo incremental? | phoenixNAP Glosario de TI.* (s/f). Recuperado el 26 de agosto de 2024, de <https://phoenixnap.mx/glosario/desarrollo-incremental>
- ¿Qué es frontend y backend? - Ecdisis Estudio.* (s/f). Recuperado el 19 de septiembre de 2024, de <https://ecdisis.com/que-es-frontend-y-backend/>
- ¿Qué es Interfaz Gráfica de Usuario y para qué sirve? - Neo Wiki.* (s/f). Recuperado el 22 de agosto de 2024, de <https://neoattack.com/neowiki/interfaz-grafica-de-usuario/>
- ¿Qué es la programación extrema (XP)? [2024] • Asana.* (s/f). Recuperado el 26 de agosto de 2024, de <https://asana.com/es/resources/extreme-programming-xp>
- ¿Qué es la programación, sus lenguajes y tipos de enfoque? - Tiffin University.* (s/f-a). Recuperado el 19 de agosto de 2024, de <https://global.tiffin.edu/blog/que-es-programacion>
- ¿Qué es la programación, sus lenguajes y tipos de enfoque? - Tiffin University.* (s/f-b). Recuperado el 19 de agosto de 2024, de <https://global.tiffin.edu/blog/que-es-programacion>
- ¿Qué es MySQL? Explicación y características | Blog de Arsys.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.arsys.es/blog/mysql>
- ¿Qué es Spring Boot y cómo se desarrollan aplicaciones? (s/f).* Recuperado el 26 de agosto de 2024, de <https://keepcoding.io/blog/que-es-spring-boot/>

- Qué es Thymeleaf* | *OpenWebinars*. (s/f). Recuperado el 26 de agosto de 2024, de <https://openwebinars.net/blog/que-es-thymeleaf/>
- ¿*Qué es un lenguaje de programación de alto nivel? Y porque saberlo.* (s/f). Recuperado el 19 de agosto de 2024, de <https://impulso06.com/que-es-un-lenguaje-de-programacion-de-alto-nivel-y-porque-deberias-saberlo/>
- ¿*Qué es una interfaz?* | *KeepCoding Bootcamps*. (s/f). Recuperado el 22 de agosto de 2024, de <https://keepcoding.io/blog/que-es-una-interfaz/>
- ¿*Qué es Visual Studio Code y cuáles son sus ventajas?* | *Blog de Arsys*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>
- ¿*Qué son los lenguajes tipados y no tipados? (Explicación sencilla)* | *EDteam*. (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/que-son-los-lenguajes-tipados-y-no-tipados-explicacion-sencilla>
- ¿*Qué son los paradigmas de programación?* | *EDteam*. (s/f). Recuperado el 20 de agosto de 2024, de <https://ed.team/blog/que-son-los-paradigmas-de-programacion-f57de068-6426-4332-b5dd-dbdcff1aaec>
- ¿*Qué son los paradigmas de programación?* - *Platzi*. (s/f). Recuperado el 22 de julio de 2024, de <https://platzi.com/clases/2211-historia-programacion/35072-que-son-los-paradigmas-de-programacion/>
- Rodríguez, Á., Martínez, L., Carolina, M., López, R., & Ramón, M. (2023). *Propuesta de un sistema de gestión y control de inventarios en la bodega de almacenamiento a partir de un estudio de rotación y trazabilidad de los productos comercializados por la Distribuidora La Ruta Costa Rica durante el periodo 2021-2022*. Puntarenas, Costa Rica. <https://hdl.handle.net/20.500.13077/838>
- Rueda Vera, G., Avendaño Castro, W. R., & Parada Trujillo, A. E. (2022). Sistemas de información y control de inventarios en Micro Pequeñas y Medianas Empresas - Mipymes de la ciudad de Cúcuta, Colombia. *Saber, Ciencia y Libertad*, 17(2), 328–351. <https://doi.org/10.18041/2382-3240/saber.2022v17n2.9295>
- Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2024]* • *Asana*. (s/f). Recuperado el 26 de agosto de 2024, de <https://asana.com/es/resources/what-is-scrum>
- Silva Polo, C. R., & Ramírez García, V. J. (2021). Desarrollo de un sistema de gestión de activos para mejorar su control en el departamento de Tecnología de Información de la empresa Industrias Plásticas Reunidas S.A.C. *Repositorio Institucional – UCS*. <https://doi.org/10.21142/TB.2020.2126>
- sistema de inventario: Toma el control de tu almacén (2023)* | *Colombia*. (s/f). Recuperado el 4 de agosto de 2024, de <https://blog.alegra.com/colombia/sistema-de-inventarios/>

Sistema de inventarios: qué es y cuáles son sus tipos. (s/f). Recuperado el 4 de agosto de 2024, de <https://www.doofinder.com/es/blog/sistema-de-inventarios>

Sistemas Computacionales, I. E., Milagro, E. DE, & Izquierdo Albarracín Miguel Rolando Sr Shigla Morocho Alex Paul, S. (2021). Desarrollo de un Sistema WEB para el Control y Gestión de Reactivos e Insumos de los Laboratorios de Ingeniería de la Universidad Estatal de Milagro. *Repositorio de la Universidad Estatal de Milagro*. <http://repositorio.unemi.edu.ec/xmlui/handle/123456789/6025>

Sistemas de información en empresas: definición, tipos y ejemplos. (s/f). Recuperado el 18 de septiembre de 2024, de <https://blog.hubspot.es/marketing/sistema-informacion#que>

spring-boot-logo | yes. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.yessinfotech.com/spring-boot/spring-boot-logo/>


Tipos de lenguajes de programación | EDteam. (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/tipos-de-lenguajes-de-programacion>


Tipos y clasificación de inventarios existentes. (s/f). Recuperado el 14 de agosto de 2024, de <https://www.xamai.com/blog/clasificacion-de-inventarios>

Tipos y niveles de lenguajes de programación ejemplos 2024 | AAU. (s/f). Recuperado el 19 de agosto de 2024, de <https://www.auniv.com/s/blog/lenguajes-de-programacion/>

Un poco de historia sobre las bases de datos - ADR Formación. (s/f). Recuperado el 26 de agosto de 2024, de https://www.adrformacion.com/knowledge/programacion/un_poco_de_historia_sobre_las_bases_de_datos.html

Visual Studio Code - Code Editing. Redefined. (s/f). Recuperado el 26 de agosto de 2024, de <https://code.visualstudio.com/>

 *¿Qué es la Programación orientada a objetos? | EDteam.* (s/f). Recuperado el 20 de agosto de 2024, de <https://ed.team/comunidad/que-es-la-programacion-orientada-a-objetos>

 *Editor vs IDE | EDteam.* (s/f). Recuperado el 26 de agosto de 2024, de <https://ed.team/comunidad/editor-vs-ide>

3.1 Requerimientos funcionales y no funcionales. (s/f). Recuperado el 22 de agosto de 2024, de <https://view.genially.com/659329ab9c403800147f96f2/interactive-content-31-requerimientos-funcionales-y-no-funcionales>

8 ejemplos de diagramas de casos de uso para inspirarse | Boardmix. (s/f). Recuperado el 22 de agosto de 2024, de <https://boardmix.com/es/articles/use-case-examples/>

(25) Explorando las Similitudes entre Metodologías Ágiles y Desarrollo Tradicional. | LinkedIn. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.linkedin.com/pulse/explorando-las-similitudes-entre-metodolog%C3%ADas-%C3%A1giles-y-perea-r/>

- ¡Aprende a distinguir entre fuentes primarias, secundarias y terciarias!* (s/f). Recuperado el 25 de septiembre de 2024, de <https://tesisdeceroa100.com/aprende-a-distinguir-entre-fuentes-primarias-secundarias-y-terciarias/>
- Arquitectura de Componentes - DEV Community.* (s/f). Recuperado el 22 de agosto de 2024, de <https://dev.to/vanessamarely/arquitectura-de-componentes-283p>
- Arquitectura de datos: ¿qué es? Importancia, tipos y usos.* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.astera.com/es/type/blog/data-architecture/>
- Arquitectura de software: ¿Qué es y qué tipos hay?* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.gluo.mx/blog/arquitectura-de-software-que-es-y-que-tipos-hay>
- Bases de datos y su importancia en las empresas-Gestión de base de datos.* (s/f). Recuperado el 26 de agosto de 2024, de <https://visualizamarketingonline.com.ar/blog/bases-datos-importancia-las-empresas/>
- Bootstrap Blog · Official blog for the Bootstrap framework.* (s/f). Recuperado el 26 de agosto de 2024, de <https://blog.getbootstrap.com/>
- Bootstrap: qué es, para qué sirve y cómo usarlo | Blog de Arsys.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.arsys.es/blog/guia-completa-sobre-bootstrap>
- Casos Uso Farmacia | PDF | Farmacia | Caso de uso.* (s/f). Recuperado el 26 de septiembre de 2024, de <https://es.scribd.com/doc/64216512/Casos-uso-farmacia>
- ¿Cómo aprender a programar desde cero? | EDteam.* (s/f). Recuperado el 22 de agosto de 2024, de <https://ed.team/blog/como-aprender-programar-desde-cero>
- ¿Cómo funciona un sistema de inventario? | Defontana Chile.* (s/f). Recuperado el 4 de agosto de 2024, de <https://www.defontana.com/blog/cl/como-funciona-un-sistema-de-inventario>
- Computacionales, S. (s/f). *UNIVERSIDAD ESTATAL DE MILAGRO FACULTAD CIENCIAS E INGENIERÍA TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN.*
- Comunicación, I. Y. (2023). *Propuesta de mejora en el sistema de gestión de inventarios de roles en bodega central de suministro, por medio del uso de herramientas de la calidad que permitan apoyar la reducción de costos operativos del proceso, en departamento y área seleccionada en Ingenio Taboga, año 2022.*
<https://repositorio.ulatina.ac.cr/handle/20.500.12411/2321>
- Conceptos básicos de HTML - Aprende desarrollo web | MDN.* (s/f). Recuperado el 26 de agosto de 2024, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics
- Conditionals in Thymeleaf | Baeldung.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.baeldung.com/spring-thymeleaf-conditionals>

- Corella-Parra, L. M., Olea-Miranda, J., Corella-Parra, L. M., & Olea-Miranda, J. (2023). Desarrollo de un sistema de control de inventario para una empresa comercializadora de sistemas de riego. *Ingeniería, investigación y tecnología*, 24(1), 1–10. <https://doi.org/10.22201/FI.25940732E.2023.24.1.006>
- Crear un sitio web html usando html, css y javascript.* (s/f). Recuperado el 26 de agosto de 2024, de https://de.fiverr.com/umar_daraz_/create-html-website-using-html-css-and-javascript
- Cuadro de Operacionalización de La Variable | PDF | Logística | Inventario.* (s/f). Recuperado el 25 de septiembre de 2024, de <https://es.scribd.com/doc/127011876/CUADRO-DE-OPERACIONALIZACION-DE-LA-VARIABLE-xlsx>
- Curso: MySQL (Comienza gratis) | EDteam.* (s/f). Recuperado el 26 de agosto de 2024, de <https://ed.team/cursos/mysql>
- Definición de casos de uso - Documentación de IBM.* (s/f-a). Recuperado el 22 de agosto de 2024, de <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Definición de casos de uso - Documentación de IBM.* (s/f-b). Recuperado el 26 de septiembre de 2024, de <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Desarrollo de un software web para la generación de planes de gestión de riesgos de software.* (s/f). Recuperado el 27 de septiembre de 2024, de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000300135
- Diagrama de casos de usos | PPT.* (s/f). Recuperado el 26 de septiembre de 2024, de <https://es.slideshare.net/slideshow/diagrama-de-casos-de-usos/13589232#1>
- Diagrama de componentes UML: Qué es y cómo hacerlo | Miro.* (s/f). Recuperado el 22 de agosto de 2024, de <https://miro.com/es/diagrama/que-es-diagrama-componentes-uml/>
- Diseño Integral - FasterCapital.* (s/f). Recuperado el 22 de agosto de 2024, de <https://fastercapital.com/es/palabra-clave/dise%C3%B1o-integral.html>
- Dos siglos de historia del lenguaje de programación.* (s/f). Recuperado el 22 de julio de 2024, de <https://www.revistaonoff.es/lenguaje-de-programacion-una-historia-con-mas-de-200-anos/>
- El ciclo SDLC en 7 fases - Viewnext.* (s/f). Recuperado el 18 de septiembre de 2024, de <https://www.viewnext.com/el-ciclo-sdlc-en-7-fases/>
- el concepto del sistema de control de inventario, el gerente profesional y el trabajador están verificando el suministro de bienes y existencias, la gestión de inventario con la demanda de bienes 12494550 Vector en Vecteezy.* (s/f). Recuperado el 4 de agosto de 2024, de <https://es.vecteezy.com/arte-vectorial/12494550-el-concepto-del-sistema-de-control-de-inventario-el-gerente-profesional-y-el-trabajador-estan-verificando-el-suministro-de-bienes-y-existencias-la-gestion-de-inventario-con-la-demanda-de-bienes>

- El portal de la tesis.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://recursos.ucol.mx/tesis/investigacion.php>
- En, M., De Proyectos, G., Aguilar, A. C., Alajuela, P., & Del, J. (2021). *Plan de gestión del proyecto Inventario Forestal Nacional de Costa Rica.* <https://repositoriotec.tec.ac.cr/handle/2238/13426>
- Estrategias para gestionar y priorizar requisitos de usuario en proyectos UX.* (s/f). Recuperado el 6 de octubre de 2024, de <https://keepcoding.io/blog/requisitos-de-usuario-en-proyectos-ux/>
- Estudio de factibilidad - Qué es y qué tipos hay.* (s/f). Recuperado el 27 de septiembre de 2024, de <https://economipedia.com/definiciones/estudio-de-factibilidad.html>
- Explorar las características del editor de código - Visual Studio (Windows) | Microsoft Learn.* (s/f). Recuperado el 26 de agosto de 2024, de <https://learn.microsoft.com/es-es/visualstudio/ide/writing-code-in-the-code-and-text-editor?view=vs-2022>
- Fundamentos de JavaScript - Aprende desarrollo web | MDN.* (s/f). Recuperado el 26 de agosto de 2024, de https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- García-Pacheco, M. C., & Andrés-Laz, E. M. S. (2021). DISEÑO DE UN SISTEMA DE GESTIÓN POR PROCESOS PARA EL MANEJO DE INVENTARIOS. CASO: FERRETERÍA QUIROZ. *REVISTA CIENTÍFICA MULTIDISCIPLINARIA ARBITRADA YACHASUN - ISSN: 2697-3456*, 5(9 Ed. esp.), 180–203. <https://doi.org/10.46296/YC.V5I9EDESPSOCT.0118>
- Historia de Bases de Datos: Evolución y Orígenes - jhonmosquera.com.* (s/f). Recuperado el 26 de agosto de 2024, de <https://jhonmosquera.com/bases-de-datos-historia/>
- Historia de los lenguajes de programación - Administración de Sistemas.* (s/f). Recuperado el 19 de julio de 2024, de <https://administraciondesistemas.com/historia-de-los-lenguajes-de-programacion/>
- IDE: ¿qué es un entorno de desarrollo integrado?* (s/f). Recuperado el 26 de agosto de 2024, de <https://datascientest.com/es/ide>
- IDEs: Qué son, para qué sirven y los más populares - Blog de SW Hosting.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.swhosting.com/es/blog/ides-que-son-para-que-sirven-y-los-mas-populares>
- Introducción a la programación funcional con Java | by Johan Del Valle | Medium.* (s/f). Recuperado el 20 de agosto de 2024, de <https://johandvv.medium.com/introducci%C3%B3n-a-la-programaci%C3%B3n-funcional-con-java-89c58e0972bc>

- Investigación cualitativa | QuestionPro.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://www.questionpro.com/es/investigacion-cualitativa.html>
- Java: ¿Qué es y para qué sirve?* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.godaddy.com/resources/latam/stories/java-que-es-y-para-que-sirve>
- Las mejores bases de datos open source - Codemotion.* (s/f). Recuperado el 22 de agosto de 2024, de <https://www.codemotion.com/magazine/es/backend-es/las-5-mejores-bases-de-datos-open-source/>
- Lenguajes de Alto nivel vs de Bajo nivel | EDteam.* (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/lenguajes-de-alto-nivel-vs-de-bajo-nivel-b8806b09-73ac-4d83-9047-b487dae2cab2>
- Lenguajes de bajo nivel: aprende qué y cuáles son.* (s/f). Recuperado el 19 de agosto de 2024, de <https://keepcoding.io/blog/que-son-lenguajes-de-bajo-nivel-cuales-hay/>
- Lenguajes de programación: una breve introducción contextualizada | Alura Cursos Online.* (s/f). Recuperado el 19 de agosto de 2024, de <https://www.aluracursos.com/blog/lenguajes-de-programacion>
- Los 17 mejores editores de código.* (s/f). Recuperado el 26 de agosto de 2024, de <https://www.hostinger.es/tutoriales/editores-de-codigo>
- Los enfoques cuantitativo y cualitativo en la investigación | PPT.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://es.slideshare.net/slideshow/los-enfoques-cuantitativo-y-cualitativo-en-la-investigacin/30112318#6>
- Los sistemas de información de una empresa.* (s/f). Recuperado el 18 de septiembre de 2024, de <https://informatecdigital.com/productividad/los-sistemas-de-informacion-de-una-empresa/>
- Mapa conceptual fundamentos de programación by Christopher Diego Jiménez Almaraz on Prezi.* (s/f). Recuperado el 19 de agosto de 2024, de <https://prezi.com/bqomkgmholfj/mapa-conceptual-fundamentos-de-programacion/>
- Matriz de riesgo: Importancia y ejemplos | SafetyCulture.* (s/f). Recuperado el 27 de septiembre de 2024, de <https://safetyculture.com/es/temas/evaluacion-de-riesgos/matriz-de-riesgo/>
- Matriz de riesgos: cómo evaluar los riesgos para lograr el éxito del proyecto [2024] • Asana.* (s/f). Recuperado el 27 de septiembre de 2024, de <https://asana.com/es/resources/risk-matrix-template>
- Método UEPS (last in, first out): usos y ejemplos.* (s/f). Recuperado el 4 de agosto de 2024, de <https://www.mecalux.com.mx/blog/metodo-ueps>
- Metodología de la Investigación: MARCO METODOLOGICO.* (s/f). Recuperado el 24 de septiembre de 2024, de <https://uftmetodologia.blogspot.com/2019/09/marco-metodologico.html>

- Metodología en cascada: Ventajas e inconvenientes* | *SafetyCulture*. (s/f). Recuperado el 26 de agosto de 2024, de <https://safetyculture.com/es/temas/metodologia-en-cascada/>
- Metodología Kanban: en qué consiste y cómo utilizarla* | *APD*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.apd.es/metodologia-kanban/>
- Perfil y Salario de un Perfil de Desarrollador Web en México en 2024 - Hireline México*. (s/f). Recuperado el 19 de septiembre de 2024, de <https://hireline.io/mx/enciclopedia-de-perfiles-de-tecnologia/desarrollador-web>
- Plan de pruebas de software: 8 pasos para realizarlas - icaria Technology*. (s/f). Recuperado el 26 de septiembre de 2024, de <https://icariatechnology.com/plan-de-pruebas-de-software/>
- Programación imperativa: ventajas y desventajas del paradigma - IONOS*. (s/f). Recuperado el 20 de agosto de 2024, de <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/programacion-imperativa/>
- Prototipo – el papel del prototipado en el proceso de desarrollo de productos - Copymate*. (s/f). Recuperado el 26 de agosto de 2024, de <https://copymate.app/es/blog/multi/prototipo-el-papel-del-prototipado-en-el-proceso-de-desarrollo-de-productos/>
- Publicaes*. (s/f). Recuperado el 24 de septiembre de 2024, de https://www.publicaes.com/?fbclid=IwY2xjawFhs5RleHRuA2F1bQIxMAABHW9ksVxMxQ0Z0RqdZjYUF9_XfQUV-CstGjDKRgpQiWr2oMGKDco-o_RNJQ_aem__o-l3yxTNupy11q76mSoZw
- ¿Qué es CSS y para qué sirve?* - *Webempresa*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.webempresa.com/blog/que-es-css-y-para-que-sirve.html>
- ¿Qué es el desarrollo incremental?* | *phoenixNAP Glosario de TI*. (s/f). Recuperado el 26 de agosto de 2024, de <https://phoenixnap.mx/glosario/desarrollo-incremental>
- ¿Qué es frontend y backend?* - *Ecdisis Estudio*. (s/f). Recuperado el 19 de septiembre de 2024, de <https://ecdisis.com/que-es-frontend-y-backend/>
- ¿Qué es Interfaz Gráfica de Usuario y para qué sirve?* - *Neo Wiki*. (s/f). Recuperado el 22 de agosto de 2024, de <https://neoattack.com/neowiki/interfaz-grafica-de-usuario/>
- ¿Qué es la programación extrema (XP)? [2024]* • *Asana*. (s/f). Recuperado el 26 de agosto de 2024, de <https://asana.com/es/resources/extreme-programming-xp>
- ¿Qué es la programación, sus lenguajes y tipos de enfoque?* - *Tiffin University*. (s/f-a). Recuperado el 19 de agosto de 2024, de <https://global.tiffin.edu/blog/que-es-programacion>
- ¿Qué es la programación, sus lenguajes y tipos de enfoque?* - *Tiffin University*. (s/f-b). Recuperado el 19 de agosto de 2024, de <https://global.tiffin.edu/blog/que-es-programacion>
- ¿Qué es MySQL? Explicación y características* | *Blog de Arsys*. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.arsys.es/blog/mysql>

- ¿Qué es Spring Boot y cómo se desarrollan aplicaciones? (s/f). Recuperado el 26 de agosto de 2024, de <https://keepcoding.io/blog/que-es-spring-boot/>
- Qué es Thymeleaf | OpenWebinars. (s/f). Recuperado el 26 de agosto de 2024, de <https://openwebinars.net/blog/que-es-thymeleaf/>
- ¿Qué es un lenguaje de programación de alto nivel? Y porque saberlo. (s/f). Recuperado el 19 de agosto de 2024, de <https://impulso06.com/que-es-un-lenguaje-de-programacion-de-alto-nivel-y-porque-deberias-saberlo/>
- ¿Qué es una interfaz? | KeepCoding Bootcamps. (s/f). Recuperado el 22 de agosto de 2024, de <https://keepcoding.io/blog/que-es-una-interfaz/>
- ¿Qué es Visual Studio Code y cuáles son sus ventajas? | Blog de Arsys. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>
- ¿Qué son los lenguajes tipados y no tipados? (Explicación sencilla) | EDteam. (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/que-son-los-lenguajes-tipados-y-no-tipados-explicacion-sencilla>
- ¿Qué son los paradigmas de programación? | EDteam. (s/f). Recuperado el 20 de agosto de 2024, de <https://ed.team/blog/que-son-los-paradigmas-de-programacion-f57de068-6426-4332-b5dd-dbdceff1aaec>
- ¿Qué son los paradigmas de programación? - Platzi. (s/f). Recuperado el 22 de julio de 2024, de <https://platzi.com/clases/2211-historia-programacion/35072-que-son-los-paradigmas-de-programacion/>
- Rodríguez, Á., Martínez, L., Carolina, M., López, R., & Ramón, M. (2023). *Propuesta de un sistema de gestión y control de inventarios en la bodega de almacenamiento a partir de un estudio de rotación y trazabilidad de los productos comercializados por la Distribuidora La Ruta Costa Rica durante el periodo 2021-2022*. Puntarenas, Costa Rica. <https://hdl.handle.net/20.500.13077/838>
- Rueda Vera, G., Avendaño Castro, W. R., & Parada Trujillo, A. E. (2022). Sistemas de información y control de inventarios en Micro Pequeñas y Medianas Empresas - Mipymes de la ciudad de Cúcuta, Colombia. *Saber, Ciencia y Libertad*, 17(2), 328–351. <https://doi.org/10.18041/2382-3240/saber.2022v17n2.9295>
- Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2024] • Asana. (s/f). Recuperado el 26 de agosto de 2024, de <https://asana.com/es/resources/what-is-scrum>
- Silva Polo, C. R., & Ramírez García, V. J. (2021). Desarrollo de un sistema de gestión de activos para mejorar su control en el departamento de Tecnología de Información de la empresa Industrias Plásticas Reunidas S.A.C. *Repositorio Institucional – UCS*. <https://doi.org/10.21142/TB.2020.2126>

sistema de inventario: Toma el control de tu almacén (2023) | Colombia. (s/f). Recuperado el 4 de agosto de 2024, de <https://blog.alegra.com/colombia/sistema-de-inventarios/>

Sistema de inventarios: qué es y cuáles son sus tipos. (s/f). Recuperado el 4 de agosto de 2024, de <https://www.doofinder.com/es/blog/sistema-de-inventarios>

Sistemas Computacionales, I. E., Milagro, E. DE, & Izquierdo Albarracín Miguel Rolando Sr Shigla Morocho Alex Paul, S. (2021). Desarrollo de un Sistema WEB para el Control y Gestión de Reactivos e Insumos de los Laboratorios de Ingeniería de la Universidad Estatal de Milagro. *Repositorio de la Universidad Estatal de Milagro*. <http://repositorio.unemi.edu.ec/xmlui/handle/123456789/6025>

Sistemas de información en empresas: definición, tipos y ejemplos. (s/f). Recuperado el 18 de septiembre de 2024, de <https://blog.hubspot.es/marketing/sistema-informacion#que>

spring-boot-logo | yes. (s/f). Recuperado el 26 de agosto de 2024, de <https://www.yessinfotech.com/spring-boot/spring-boot-logo/>

Tipos de lenguajes de programación | EDteam. (s/f). Recuperado el 19 de agosto de 2024, de <https://ed.team/blog/tipos-de-lenguajes-de-programacion>

Tipos y clasificación de inventarios existentes. (s/f). Recuperado el 14 de agosto de 2024, de <https://www.xamai.com/blog/clasificacion-de-inventarios>

Tipos y niveles de lenguajes de programación ejemplos 2024 | AAU. (s/f). Recuperado el 19 de agosto de 2024, de <https://www.auniv.com/s/blog/lenguajes-de-programacion/>

Un poco de historia sobre las bases de datos - ADR Formación. (s/f). Recuperado el 26 de agosto de 2024, de https://www.adrformacion.com/knowledge/programacion/un_poco_de_historia_sobre_las_bases_de_datos.html