

UNIVERSIDAD BOLIVIANA DE INFORMÁTICA

SUB SEDE CIUDAD DE LA PAZ

CARRERA INGENIERÍA DE SISTEMAS



**UNIVERSIDAD BOLIVIANA
DE INFORMÁTICA**

PROYECTO DE GRADO

**Sistema Web y Aplicación Móvil para la gestión de la experiencia turística en la agencia
"JIWAKY"**

**PROYECTO DE GRADO PARA OBTENCIÓN
EL GRADO ACADÉMICO DE LICENCIATURA
EN INGENIERÍA DE SISTEMAS.**

Postulante: MIGUEL ANGEL HUAYCHO HERMOZO

Tutor: VICTOR RAMIRO TELLEZ PEÑARANDA

La Paz – Bolivia

INDICE

| | |
|---|----|
| CAPÍTULO 1: INTRODUCCIÓN | 1 |
| 1.1. ANTECEDENTES | 1 |
| 1.2. PROBLEMÁTICA | 2 |
| 1.2.1. PLANTEAMIENTO DEL PROBLEMA | 2 |
| 1.2.2. FORMULACIÓN DEL PROBLEMA | 3 |
| 1.3. OBJETIVOS | 4 |
| 1.3.1. OBJETIVO GENERAL | 4 |
| 1.3.2. OBJETIVOS ESPECÍFICOS | 4 |
| 1.4. JUSTIFICACIÓN | 5 |
| 1.4.1. ECONÓMICA | 5 |
| 1.4.2. SOCIAL | 6 |
| 1.4.3. TÉCNICA | 7 |
| 1.5. LÍMITES Y ALCANCES | 8 |
| 1.5.1. LÍMITES | 9 |
| 1.5.2. ALCANCES | 9 |
| 1.6. NOVEDADES TECNOLÓGICAS | 9 |
| CAPÍTULO 2: MARCO TEÓRICO | 11 |
| 2.1. SISTEMA DE INFORMACIÓN | 11 |
| 2.1.1. ELEMENTOS DE UN SISTEMA DE INFORMACIÓN | 11 |
| 2.1.2. TIPOS DE SISTEMAS DE INFORMACIÓN | 12 |
| 2.1.2.1. SISTEMA DE PROCESAMIENTO DE TRANSACCIONES | 12 |
| 2.1.2.2. SISTEMA DE APOYO A LA TOMA DE DECISIONES | 13 |
| 2.2. METODOLOGÍA SCRUM | 13 |
| 2.2.1. VISIÓN GENERAL DE SCRUM | 13 |
| 2.2.2. EQUIPO SCRUM | 14 |
| 2.2.2.1. DUEÑO DE PRODUCTO (PRODUCT OWNER) | 14 |
| 2.2.2.2. MAESTRO SCRUM (SCRUM MASTER) | 14 |
| 2.2.2.3. EL EQUIPO DE DESARROLLO (DEVELOPMENT TEAM) | 15 |
| 2.2.3. PROCESOS DE SCRUM | 15 |
| 2.3. LENGUAJE UNIFICADO DE MODELADO (UML) | 15 |
| 2.3.1. MODELOS | 16 |

| | |
|--|----|
| 2.3.2. DIAGRAMAS | 17 |
| 2.3.2.1. DIAGRAMA DE CASOS DE USO | 17 |
| 2.3.2.2. DIAGRAMAS DE CLASES | 18 |
| 2.3.2.3. DIAGRAMAS DE SECUENCIA | 19 |
| 2.3.2.4. DIAGRAMA DE ESTADOS | 20 |
| 2.3.2.5. DIAGRAMA DE ACTIVIDADES | 21 |
| 2.4. TÉCNICA DE MODELADO DE OBJETOS (OMT) | 22 |
| 2.4.1. ANÁLISIS DE OBJETOS | 23 |
| 2.4.1.1. MODELO DE OBJETOS | 23 |
| 2.4.1.2. MODELO DINÁMICO | 23 |
| 2.4.1.3. MODELO FUNCIONAL | 24 |
| 2.5. ARQUITECTURA DEL SOFTWARE | 25 |
| 2.5.1. PATRÓN MODELO VISTA CONTROLADOR | 25 |
| 2.5.1.1. EL MODELO | 25 |
| 2.5.1.2. LA VISTA | 25 |
| 2.5.1.3. CONTROLADOR | 26 |
| 2.5.2. ARQUITECTURA DE 3 CAPAS | 26 |
| 2.5.2.1. CAPA DE PRESENTACIÓN | 26 |
| 2.5.2.2. CAPA DE REGLAS DE NEGOCIO (EMPRESARIAL) | 27 |
| 2.5.2.3. CAPA DE DATOS | 27 |
| 2.6. SEGURIDAD | 27 |
| 2.6.1. PROPIEDADES DE UN SISTEMA DE INFORMACIÓN SEGURO | 28 |
| 2.6.1.1. INTEGRIDAD | 28 |
| 2.6.1.2. CONFIDENCIALIDAD | 28 |
| 2.6.1.3. DISPONIBILIDAD | 28 |
| 2.6.2. MÉTODOS | 29 |
| 2.6.2.1. CORTAFUEGOS (FIREWALL) | 29 |
| 2.6.2.2. AUTENTICACIÓN Y AUTORIZACIÓN | 29 |
| 2.6.2.3. CIFRADO DE DATOS | 29 |
| 2.6.2.4. SESIÓN | 29 |
| 2.7. MODELO DE ESTIMACIÓN DE COSTOS COCOMO | 30 |
| 2.8. MÉTRICAS DE CALIDAD DEL SOFTWARE | 30 |
| 2.8.1. EL ESTÁNDAR ISO/IEC 25010 | 31 |
| 2.8.1.1. ADECUACIÓN FUNCIONAL | 31 |

| | |
|---|----|
| 2.8.1.2. EFICIENCIA DE DESEMPEÑO | 32 |
| 2.8.1.3. COMPATIBILIDAD | 32 |
| 2.8.1.4. USABILIDAD | 33 |
| 2.8.1.5. FIABILIDAD | 33 |
| 2.8.1.6. SEGURIDAD | 34 |
| 2.8.1.7. MANTENIBILIDAD | 34 |
| 2.8.1.8. PORTABILIDAD | 35 |
| 2.9. HERRAMIENTAS DE DESARROLLO WEB | 35 |
| 2.9.1. SISTEMA GESTOR DE BASE DE DATOS MYSQL | 35 |
| 2.9.2. FUNCIONAMIENTO | 35 |
| 2.9.2.1. BASE DE DATOS RELACIONAL | 36 |
| 2.9.2.2. LENGUAJE DE PROGRAMACIÓN PHP ("PREPROCESADOR DE HIPERTEXTO") | 36 |
| 2.9.2.3. MARCO DE DESARROLLO YIIFRAMEWORK II | 36 |
| 2.9.2.4. CARACTERÍSTICAS GENERALES DE ANGULAR | 36 |
| 2.9.2.5. CARACTERÍSTICAS GENERALES DE IONIC | 37 |
| 2.9.2.6. CARACTERÍSTICAS GENERALES DE CAPACITOR | 37 |
| 2.9.2.7. HTML (LENGUAJE DE MARCADO DE HIPERTEXTO) | 37 |
| 2.9.2.8. CASCADING STYLE SHEETS - HOJAS DE ESTILO EN CASCADA) | 37 |
| 2.9.2.9. JAVASCRIPT | 38 |
| 2.9.2.10. JQUERY | 38 |
| 2.9.2.11. AJAX (ASYNCHRONOUS JAVASCRIPT AND XML) | 38 |
| 2.9.2.12. JQUERY UI | 38 |
| 2.9.2.13. BOOTSTRAP | 38 |
| CAPÍTULO 3: MARCO METODOLOGICO | |
| TIPO DE INVESTIGACIÓN | 39 |
| 3.1. DISEÑO DE LA INVESTIGACIÓN | 39 |
| 3.2. SELECCIÓN DE LA POBLACIÓN | 40 |
| 3.2.1. UNIVERSO | 40 |
| 3.2.2. MUESTRA | 40 |
| 3.3. INSTRUMENTOS DE RECOLECCIÓN DE DATOS | 41 |
| 3.3.1. ENTREVISTA | 41 |
| 3.3.2. OBSERVACIÓN | 41 |
| 3.3.3. ENCUESTA | 42 |

CAPÍTULO 4: MARCO APLICATIVO

| | |
|---|----|
| 4.1. RESULTADOS DEL RELEVAMIENTO DE INFORMACIÓN | 43 |
| 4.1.1. ENTREVISTA | 43 |
| 4.1.2. OBSERVACIÓN | 43 |
| 4.1.3. ENCUESTA | 43 |
| 4.2. IDENTIFICACIÓN DE LA METODOLOGÍA DE LA PROPUESTA | 44 |
| 4.3. DESARROLLO DE LA PROPUESTA | 44 |
| 4.4. ESTIMACIÓN DE COSTOS DE DESARROLLO (ESTUDIO DE COSTOS, ANÁLISIS VAN Y TIR) | 45 |
| 4.5. CÁLCULO DEL VAN Y TIR..... | 45 |
| 4.5.1. VALOR ACTUAL NETO (VAN) | 46 |
| 4.5.2. TASA INTERNA DE RETORNO (TIR) | 47 |
| 4.6. CONTROL DE CALIDAD (APLICACIÓN DE NORMAS DE CALIDAD) | 48 |
| 4.7. PRUEBAS DE SOFTWARE | 48 |
| 4.7.1. PRUEBAS DE CAJA NEGRA | 49 |
| 4.7.2. PRUEBAS DE CAJA BLANCA..... | 50 |
| 4.8. SEGURIDAD | 51 |

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

| | |
|----------------------------|----|
| 5.1. CONCLUSIONES..... | 55 |
| 5.2. RECOMENDACIONES | 56 |

CAPITULO I

1-1 INTRODUCCIÓN

En el contexto de un entorno económico altamente competitivo, las empresas buscan sistemas de información eficaces para optimizar su eficiencia y obtener una visión completa de sus operaciones. Este desafío se manifiesta en diversos sectores, y el turismo destaca como uno de los más dinámicos en Bolivia, con un crecimiento positivo notable.

Dentro de la vasta industria turística, que abarca desde establecimientos hoteleros hasta parques temáticos, este trabajo se enfoca específicamente en el análisis detallado de las agencias de viaje, con JIWAKY como el caso de estudio central.

JIWAKY, en su calidad de agencia de viajes, se especializa en una amplia gama de servicios para el viajero, que incluyen reservas de boletos, tours, alquiler de autos, alquiler de bicicletas y más. La agencia se distingue por acercar de manera personalizada el producto turístico a sus clientes. La estructura organizativa de JIWAKY comprende agencias mayoristas, encargadas de diseñar paquetes turísticos, y agencias minoristas, que representan el contacto directo con el cliente y son responsables de las ventas.

Las funciones clave de JIWAKY abarcan desde proporcionar información detallada a los clientes hasta actuar como intermediarios en la gestión de reservas, desempeñando además un papel productor al confeccionar productos y servicios únicos. Los paquetes turísticos de JIWAKY incluyen desde excursiones hasta opciones que incorporan comidas, todo adaptado a las características específicas de cada paquete. Este trabajo explorará en profundidad el funcionamiento y las dinámicas de JIWAKY como representante destacado en el sector de agencias de viaje.

1-2 ANTECEDENTES

Los sistemas de información en agencias de turismo son herramientas fundamentales diseñadas para respaldar las actividades de reserva y gestión de servicios turísticos. En este contexto, la agencia de turismo JIWAKY busca aprovechar estas soluciones informáticas para ofrecer un servicio eficiente y competitivo a sus clientes.

Es importante destacar que, a diferencia de otras agencias, JIWAKY aún no cuenta con un sistema moderno y sigue utilizando tecnología más antigua, lo que puede limitar la eficiencia de sus operaciones y retrasar sus movimientos en el dinámico mercado turístico actual.

Estos sistemas suelen integrar una base de datos completa que abarca destinos turísticos, opciones de alojamiento, vuelos, alquiler de vehículos y otros medios de transporte. Además, cuentan con módulos dedicados a las reservas y la gestión de clientes, permitiendo una administración eficaz de grandes volúmenes de reservas y clientes.

A lo largo de las décadas, estos sistemas han evolucionado en paralelo al crecimiento del sector turístico y la creciente demanda de servicios de reservas en línea. Inicialmente utilizados por grandes agencias de viajes en la década de 1970, han pasado a ser herramientas sofisticadas y accesibles para agencias de todos los tamaños. La disponibilidad en línea de muchos de estos sistemas ha permitido que las agencias ofrezcan servicios de reservas y gestión de viajes en línea a sus clientes.

En la actualidad, los sistemas de información en agencias de turismo abarcan una amplia variedad de funcionalidades esenciales, como la reserva de vuelos, hoteles y alquiler de coches, la gestión de clientes, y la generación de informes y estadísticas. JIWAKY reconoce la importancia de estas herramientas para mantenerse competitiva y eficiente en el dinámico mercado turístico actual, y se propone modernizar sus operaciones para estar a la vanguardia del sector.

1-3 PROBLEMÁTICA

1-3.1 PLANTEAMIENTO DEL PROBLEMA

La agencia de turismo JIWAKY se encuentra frecuentemente enfrentando diversos desafíos en su gestión administrativa, los cuales pueden incidir negativamente en su eficiencia operativa y en la prestación de servicios de calidad. Entre los problemas más recurrentes se encuentran:

- **Registro y Gestión de Clientes:** La agencia JIWAKY enfrenta la carencia de un sistema eficiente para almacenar y gestionar información detallada sobre sus clientes.

- **Registro y Administración de Paquetes Turísticos:** La falta de un sistema organizado para registrar y actualizar información sobre los paquetes turísticos ofrecidos constituye un desafío.
- **Seguimiento y Planificación de Viajes Turísticos:** La agencia carece de una herramienta que permita llevar un seguimiento adecuado de los viajes realizados y planificados.
- **Gestión de Personal en JIWAKY:** Existe una necesidad imperante de una plataforma que facilite la gestión de información sobre el personal de la agencia, incluyendo roles, responsabilidades y datos relevantes.
- **Apoyo a el Marketing:** La carencia de estrategias efectivas de apoyo al marketing y promoción puede limitar significativamente la adquisición y retención de clientes.
- **Modernización Tecnológica:** El uso de sistemas obsoletos de gestión representa un obstáculo para la automatización de procesos y la adaptación a las tendencias tecnológicas actuales.
- **Gestión Ineficiente de Reservas:** Las dificultades para gestionar eficazmente las reservas de viajes, tours y otros servicios pueden resultar en errores significativos en las fechas.
- **Registro de Comprobantes de Pago:** Pérdida de comprobantes de pago representa un desafío significativo en la gestión financiera de la agencia.
- **Generación de Reportes e Informes:** Se enfrenta un problema recurrente en la generación de informes, lo cual conlleva pérdida de datos y retrasos notables.
- **Demora en Transferir Multimedia del Viaje a los Clientes:** Experimenta demoras significativas en la entrega de imágenes y videos a los clientes.

1-3.2 FORMULACIÓN DEL PROBLEMA

¿Cómo podemos abordar los desafíos fundamentales que enfrenta la agencia de turismo JIWAKY en la gestión de clientes, paquetes turísticos, seguimiento de viajes y gestión de personal, así como apoyar al marketing, superar la tecnología obsoleta y resolver problemas en la gestión financiera, como la pérdida de comprobantes de pago, generación de reportes, evitar retrasos en informes y agilizar la entrega de multimedia a los clientes para mejorar su competitividad y eficiencia operativa?

1-4 OBJETIVOS.

1-4.1 OBJETIVO GENERAL

Desarrollar e implementar un sistema integral en la agencia de turismo JIWAKY que aborde de manera efectiva los desafíos en la gestión de clientes, paquetes turísticos, seguimiento de viajes y gestión de personal. El objetivo es mejorar significativamente la eficiencia operativa de la agencia al proporcionar soluciones tecnológicas modernas, apoyo a el marketing, y mejorar la gestión financiera para optimizar procesos clave como la generación de reportes, evitar retrasos en informes, y agilizar la entrega de multimedia a los clientes.

1-4.2 OBJETIVOS ESPECÍFICOS

- **Registro y Gestión de Clientes:** Implementar un sistema eficiente de gestión de clientes que permita almacenar y gestionar de manera detallada la información de los clientes de JIWAKY.
- **Registro y Administración de Paquetes Turísticos:** Desarrollar un sistema organizado para el registro y actualización de información sobre los paquetes turísticos ofrecidos por la agencia.
- **Seguimiento y Planificación de Viajes Turísticos:** Implementar un sistema que facilite el seguimiento preciso de los viajes realizados y planificados.
- **Gestión de Personal en JIWAKY:** Establecer un modulo que simplifique la gestión de información sobre el personal, incluyendo roles (*Administrador, subadministrador y guías*) y datos relevantes para mejorar la eficiencia operativa.
- **Apoyo al Marketing:** Desarrollar post en el sistema en apoyo a el marketing y promoción que impulsen la adquisición y retención de clientes, fortaleciendo la posición competitiva de JIWAKY.
- **Modernización Tecnológica:** Actualizar los sistemas obsoletos de gestión para permitir la automatización de procesos y asegurar la adaptación a las tendencias tecnológicas actuales.
- **Gestión Eficiente de Reservas:** Establecer un módulo para gestionar eficazmente las reservas de viajes, reduciendo errores.

- **Registro de Comprobantes de Pago:** Establecer un módulo seguro y eficiente para el registro y almacenamiento de comprobantes de pago.
- **Generación de Reportes e Informes:** Resolver los problemas recurrentes en la generación de informes, asegurando la integridad de los datos y eliminando retrasos notables en el proceso.
- **Agilización en la Transferencia de Multimedia:** Desarrollar e implementar una aplicación móvil que permita a los guías fotografiar y cargar instantáneamente imágenes al sistema web. Esta solución agilizará el proceso de compartir multimedia con los clientes que hayan realizado el viaje, reduciendo de manera significativa las demoras en la entrega de imágenes y videos. El objetivo es mejorar la experiencia del cliente al proporcionar un acceso más rápido y eficiente a contenido visual relevante.

1-5 JUSTIFICACIÓN

1-5.1 ECONÓMICA

La justificación económica de implementar el sistema web de información y la aplicación móvil para guías turísticos y turistas en la agencia de turismo JIWAKI se sustenta en la perspectiva de que esta inversión tecnológica resultará en un retorno de inversión ROI (Retorno de la Inversión) favorable. A continuación, se detallan los aspectos fundamentales de esta justificación económica:

- **Ahorro de costos operativos:** La automatización de procesos reducirá significativamente los costos operativos relacionados con la gestión manual de reservas, facturación y otros procesos administrativos. Se espera una disminución en los costos de personal y recursos, lo que contribuirá a un flujo de efectivo más positivo.
- **Mejora de la eficiencia:** El sistema mejorará la eficiencia en la planificación y ejecución de viajes, lo que permitirá atender a más clientes con los mismos recursos o incluso menos. Esto conlleva una reducción de costos y un aumento potencial en la rentabilidad.
- **Incremento de ingresos:** La capacidad de atraer nuevos clientes y mejorar la retención de clientes existentes a través de una oferta de servicios más eficiente y una mejor experiencia del cliente puede aumentar los ingresos de la agencia.

- **Reducción de errores:** La disminución de errores en la gestión de información evitará costos asociados con la corrección de errores y la posible insatisfacción del cliente, lo que podría resultar en pérdida de ingresos.
- **Mayor competitividad:** Mantenerse al día con la tecnología es esencial para mantener la competitividad en la industria del turismo. La implementación del sistema mejorará la imagen de JIWAKI y su capacidad para competir con otras agencias.
- **Análisis de datos mejorado:** La generación de informes precisos y la capacidad de tomar decisiones basadas en datos mejorarán la toma de decisiones estratégicas, lo que puede resultar en inversiones más efectivas y, en última instancia, en un mayor retorno de inversión.

En conjunto, la inversión en este sistema web de información está respaldada por la expectativa de que los ahorros de costos, el aumento de ingresos y la mejora en la eficiencia llevarán a un ROI positivo a lo largo del tiempo. Esto justifica económicamente la implementación del sistema como una inversión estratégica para el crecimiento y la rentabilidad a largo plazo de la agencia de turismo JIWAKI.

1-5.2 SOCIAL

En el marco de nuestra investigación, se concentra en la creación e implementación de un sistema web de información y una aplicación móvil destinados a guías turísticos y turistas dentro de la agencia de turismo JIWAKI. Más allá de los beneficios económicos que esta inversión puede generar, es esencial poner de relieve los aspectos sociales que desempeñan un rol fundamental en la justificación de este proyecto.

- **Enriquecimiento de la experiencia del turista:** Uno de los pilares fundamentales de esta iniciativa es enriquecer la experiencia de los turistas que visitan nuestra región. La aplicación móvil permitirá a los viajeros acceder de forma sencilla a información relevante acerca de destinos, actividades y servicios turísticos, lo que mejorará su estancia y posiblemente motive visitas repetidas.
- **Fomento al turismo:** La iniciativa tiene como propósito promover el turismo sostenible al proporcionar información sobre prácticas responsables y la conservación del entorno

natural y cultural. Esto contribuirá a la promoción de un turismo responsable y beneficioso tanto para la comunidad como para el entorno medioambiental.

- **Accesibilidad universal:** La inclusión de características de accesibilidad en la aplicación móvil garantizará que las personas puedan disfrutar plenamente de los servicios turísticos, promoviendo así la inclusión social y la equidad.
- **Fomento del desarrollo:** La capacitación de la comunidad local en la operación y el mantenimiento de la tecnología utilizada en este proyecto proporcionará una valiosa oportunidad para el desarrollo de habilidades y competencias en el campo de la tecnología, beneficiando a los habitantes locales.
- **Promoción de la identidad cultural local:** La incorporación de información sobre la cultura y la historia locales en la aplicación móvil contribuirá a preservar y promover la herencia cultural de la región, aspecto esencial para la identidad y el orgullo de la comunidad.

En resumen, la justificación social de este proyecto no se limita a consideraciones económicas, sino que resalta cómo el proyecto tendrá un impacto positivo y en la experiencia de los turistas. Esto refuerza la importancia del proyecto en términos de responsabilidad social y desarrollo comunitario.

1-5.3 TÉCNICA

La justificación técnica para la implementación del sistema web de información y la aplicación móvil en la agencia de turismo JIWAKI es un componente esencial en el desarrollo de este proyecto. Aquí te presento cómo podrías abordar la justificación técnica:

- **Selección de tecnología adecuada:** La elección de un sistema web y una aplicación móvil como solución tecnológica está respaldada por la versatilidad y la accesibilidad que estas plataformas ofrecen. Además, las tecnologías web y móviles se han convertido en estándares en la industria, lo que garantiza la compatibilidad y la capacidad de adaptación futura.
- **Optimización de la experiencia del usuario:** La implementación de una aplicación móvil permitirá a los usuarios acceder a información y servicios de manera rápida y conveniente

desde sus dispositivos móviles. Esto mejora significativamente la experiencia del usuario y la accesibilidad a los recursos turísticos.

- **Integración de funcionalidades:** La elección de tecnologías adecuadas permitirá la integración de funcionalidades clave, como la gestión de reservas, la consulta de itinerarios, la geolocalización y la personalización de la experiencia del usuario. Estas capacidades son esenciales para el éxito de la agencia de turismo.
- **Escalabilidad y mantenibilidad:** La elección de tecnologías escalables y de fácil mantenimiento garantizará que el sistema pueda crecer con las necesidades de la agencia y mantenerse actualizado a medida que evolucionen las tendencias tecnológicas.
- **Optimización de recursos y eficiencia operativa:** La automatización de procesos y la centralización de datos a través de un sistema web y una aplicación móvil permitirán una gestión más eficiente de recursos, lo que reducirá costos operativos y mejorará la eficiencia en la operación diaria.
- **Adaptación a futuras innovaciones:** La elección de tecnologías flexibles y adaptables permitirá a la agencia de turismo JIWAKI estar preparada para aprovechar futuras innovaciones tecnológicas y mantenerse competitiva en un entorno en constante cambio.

En resumen, la justificación técnica se basa en la selección de tecnologías adecuadas que permitirán la optimización de la operación de la agencia de turismo, mejorarán la experiencia del usuario y asegurarán la seguridad y la escalabilidad del sistema a largo plazo.

1-6 LÍMITES Y ALCANCES.

1-6.1 DELIMITACIONES:

- **Límites de cobertura geográfica:** El sistema de información se aplica específicamente a las operaciones y servicios de la agencia de turismo JIWAKY en una región geográfica determinada (La Paz - Bolivia). Fuera de esta región, el sistema no tiene influencia ni control.
- **Límites de funcionalidad:** El sistema está diseñado exclusivamente para la automatización y mejora de los procesos asociados con las reservas de viajes, Registro

y gestión de clientes, registro y administración de paquetes turísticos, seguimiento y planificación de viajes turísticos, gestión de personal en JIWAKY, apoyo al marketing, modernización tecnológica, registro de comprobantes de pago, generación de reportes e informes, y agilización en la transferencia de multimedia. su enfoque se limita específicamente a estas áreas operativas de la agencia, sin abarcar otras funciones.

- **Límites temporales:** El sistema se implementa con un enfoque a corto y mediano plazo para lograr mejoras en la eficiencia operativa. No abarca una estrategia a largo plazo para la agencia de turismo.
- **Límites de integración:** El sistema opera dentro de la infraestructura tecnológica existente de la agencia y puede estar limitado por la capacidad de integrarse con otros sistemas o bases de datos externos.
- **Límites de personal:** El sistema se utiliza para mejorar la eficiencia en los procesos operativos, pero no reemplaza completamente la necesidad de personal humano en funciones clave, como la toma de decisiones estratégicas.
- **Límites de acceso:** El acceso al sistema está restringido a empleados autorizados dentro de la agencia de turismo. El acceso externo o público está limitado a interfaces específicas diseñadas para clientes o partes interesadas.

1-6.2 ALCANCES:

- El sistema proporcionara información detallada sobre los destinos turísticos.
- El sistema ofrecerá una variedad de paquetes turísticos personalizados.
- El sistema proporcionara información a guías como a clientes.
- El sistema proporcionara información sobre las tarifas, viajes, promociones.

1-7 NOVEDADES TECNOLÓGICAS

La innovación tecnológica en el sistema de gestión de recursos para agencias de turismo, Guía en una aplicación móvil de viajes radica en la integración de diversas tecnologías y componentes

avanzados, dando lugar a una solución completa y altamente eficiente. Entre las tecnologías destacadas se encuentran:

- **Accesibilidad Universal:** La inclusión de características de accesibilidad en la aplicación móvil garantiza que personas con discapacidades o necesidades especiales puedan disfrutar plenamente de los servicios turísticos, lo que es un enfoque moderno y ético.
- **Tecnologías de Almacenamiento Local en la Aplicación Móvil:** El uso de SQLite y LocalStorage permite un almacenamiento eficiente de datos en dispositivos móviles, lo que mejora el rendimiento y la eficiencia de la aplicación.
- **Integración de Servicios de Mapas y Geolocalización:** La incorporación de Google Maps en la aplicación permite a los usuarios navegar y explorar destinos con facilidad, lo que mejora la experiencia de viaje.
- **Uso de Formatos de Datos Modernos (XML y JSON):** La transmisión de datos en formato JSON y XML permite una comunicación eficiente entre el cliente y el servidor, lo que mejora la velocidad y la precisión de la aplicación.
- **Interacción entre Cliente, Guía y Administrador:** El sistema facilita la conexión entre clientes, guías y administradores a través de la aplicación móvil, permitiéndoles compartir recursos multimedia entre sí de manera fluida y colaborativa.
- **Mejora de Funcionalidades Offline:** Las funcionalidades offline han sido potenciadas para brindar a los guías la capacidad de capturar fotografías incluso en situaciones sin conexión a internet. Cuando se recupera la conexión, la aplicación facilita la exportación de multimedia al sistema web, garantizando así una experiencia continua y sin interrupciones. Esta mejora asegura que, incluso en entornos con conectividad intermitente, los guías puedan documentar visualmente sus experiencias y compartir contenido de manera eficiente una vez restaurada la conexión.

CAPITULO II

2 - MARCO TEORICO

La agencia de turismo (JIWAKY) desempeñan un papel crucial en la industria turística al servir como intermediarias entre los proveedores de servicios (aerolíneas, hoteles, tours, etc.) y los viajeros. Ofrecen una variedad de servicios, como la gestión de reservas, itinerarios de viaje y asesoramiento a los clientes.

Este marco teórico proporciona un contexto sólido para investigar y desarrollar soluciones que aborden los desafíos en la operación de la agencia de turismo (JIWAKY) y promuevan la eficiencia operativa y la calidad del servicio.

2-1 SISTEMA DE INFORMACION

Un sistema de información colaborativo respalda las operaciones empresariales, destacando la tecnología, pero también incluyendo procedimientos manuales. Según Rodríguez Rodríguez y Daureo Campillo, aborda la recopilación, procesamiento, almacenamiento y distribución de información para un flujo eficaz en la organización.

2-1.1 ELEMENTOS DE UN SISTEMA DE INFORMACIÓN

Un sistema de información realiza tres actividades esenciales: entrada, procesamiento y salida. La entrada recopila datos, el procesamiento los transforma y la salida transmite la información procesada. Además, se requiere una función de retroalimentación, donde la información procesada se devuelve para evaluar y corregir la etapa de entrada según sea necesario (Laudon & Laudon, 2012).

- Entrada: Captura datos desde fuentes internas o externas, como transacciones, informes o formularios, marcando el inicio del flujo de información en el sistema.
- Proceso: Transforma, organiza y analiza los datos recopilados, realizando cálculos y operaciones para generar información valiosa y significativa para la organización.
- Salida: Presenta la información procesada de manera comprensible a través de informes, gráficos u otros formatos, facilitando la toma de decisiones y la ejecución de tareas. La salida comunica los resultados del procesamiento a usuarios finales.

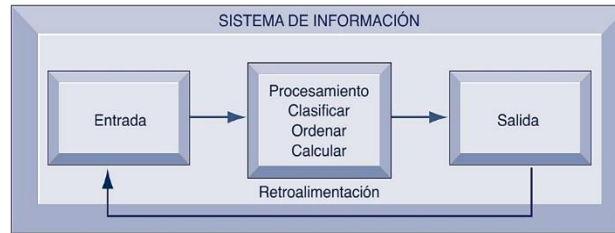


FIGURA 1. ACTIVIDADES DE UN SISTEMA DE INFORMACIÓN
FUENTE: (Laudon & Laudon, 2012)

2-1.2 TIPOS DE SISTEMAS DE INFORMACION

Los sistemas de información se adaptan a las necesidades específicas de las empresas, abordando diversas funciones. Desde los sistemas de procesamiento de transacciones para operaciones básicas hasta los sistemas de apoyo a la toma de decisiones en grupo y colaborativos para enfrentar decisiones complejas, estos sistemas cubren una amplia gama de niveles operativos, de gestión y estratégicos en una organización (Kendall & Kendall, 2005).

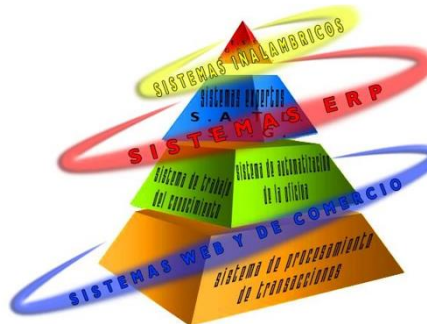


FIGURA 2. TIPOS DE SISTEMAS DE INFORMACION
FUENTE. (Kendall & Kendall, 2005)

2-1.2.1 SISTEMA DE PROCESAMIENTO DE TRANSACCIONES

Los sistemas de procesamiento de transacciones a nivel empresarial se refieren a aquellos sistemas diseñados para gestionar y registrar las actividades comerciales y financieras de manera rutinaria y cotidiana. Estas transacciones pueden abarcar una amplia gama de operaciones, como el procesamiento de nóminas, la facturación a clientes, el control de inventario y la gestión de depósitos bancarios. La naturaleza específica de estas transacciones puede variar significativamente en función del tipo de empresa y su sector de actividad.

2-1.2.2 SISTEMA DE APOYO A LA TOMA DE DECISIONES

Un Sistema de Apoyo a la Toma de Decisiones, en un sentido general, es un sistema informático diseñado para facilitar y mejorar el proceso de toma de decisiones. En un contexto más específico, se trata de un sistema de información interactivo, flexible y adaptable, especialmente creado para respaldar la resolución de problemas de gestión que no siguen una estructura predefinida, con el objetivo de mejorar la calidad de las decisiones tomadas. Estos sistemas se basan en el uso de datos, proporcionan una interfaz de usuario amigable y permiten a los usuarios tomar decisiones basadas en un análisis integral de la situación.

2-2 METODOLOGIA SCRUM

2-2.1 VISION GENERAL DE SCRUM

Scrum es un marco de trabajo ágil que aborda desafíos complejos y adaptables, permitiendo la entrega eficiente de productos de alto valor. Es ligero, fácil de entender para todos los involucrados, pero requiere tiempo y experiencia para dominarlo. Utilizado ampliamente desde la década de 1990, no es un proceso específico, sino un marco flexible para aplicar diversas técnicas y procesos en el desarrollo de productos.

“En un nivel más específico, según "SCRUM PRIMER" de Deemer, Benefield, Larman y Vodde (2012), Scrum se define como un enfoque en el cual equipos multifuncionales colaboran en la creación iterativa e incremental de productos o proyectos. El trabajo se estructura en ciclos llamados Sprints o iteraciones, con una duración típica de dos semanas. Cada Sprint se enfoca en un objetivo claro y tangible acordado por el equipo, y durante su transcurso no se pueden agregar elementos nuevos. Los cambios se adaptan en el siguiente Sprint.”

Al final de cada Sprint, el equipo realiza una revisión con las partes interesadas y demuestra lo que han logrado. El feedback recibido se incorpora en el siguiente Sprint. Scrum pone un fuerte énfasis en entregar un producto completamente funcional al final de cada iteración, lo que significa que cada incremento es una versión "terminada" del producto.

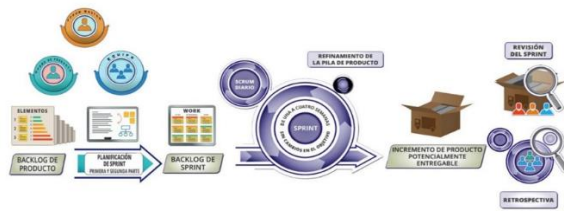


FIGURA 3. VISION GENERAL DE SCRUM
FUENTE. (Deemer, Benefield, Larman, & Vodde, 2012)

2-2.2 EL EQUIPO SCRUM

Un equipo Scrum, según Schwaber y Sutherland (2013), consta de tres roles principales: Dueño de Producto, Equipo de Desarrollo y Scrum Master. Estos equipos son autoorganizados y multifuncionales, lo que significa que tienen autonomía para tomar decisiones internas y poseen todas las habilidades necesarias para realizar sus tareas sin depender de fuentes externas. Esta estructura busca maximizar la flexibilidad, creatividad y productividad en el entorno de trabajo.

2-2.2.1 DUEÑO DE PRODUCTO (PRODUCT OWNER)

El propietario del producto (product owner) es quien toma las decisiones del cliente. Su responsabilidad es el valor del producto. Para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona. Si el cliente es una organización grande, o con varios departamentos, puede adoptar la forma de comunicación interna que consideren oportuna, pero en el equipo de desarrollo sólo se integra una persona en representación del cliente, y ésta debe tener el conocimiento suficiente del producto y las atribuciones, necesarias para tomar las decisiones que le corresponden. (Palacio, 2015)

2-2.2.2 MAESTRO SCRUM (SCRUM MASTER)

El Scrum Master desempeña un papel esencial en la implementación exitosa de Scrum, actuando como educador, mentor y líder del equipo. Su responsabilidad principal es guiar y apoyar al Equipo Scrum, eliminando obstáculos y creando un entorno propicio. Además, actúa como intermediario entre el equipo y las partes externas, facilitando la colaboración y optimizando la comunicación para maximizar el valor del equipo Scrum.

2-2.2.3 EL EQUIPO DE DESARROLLO (DEVELOPMENT TEAM)

El Equipo de Desarrollo en Scrum es un grupo autónomo y responsable de crear un Incremento de producto "Terminado" al final de cada Sprint. Con flexibilidad y autoridad para organizar su trabajo, este equipo colabora de manera sinérgica para lograr eficiencia y alta calidad en la entrega. Su autonomía les permite tomar decisiones independientes, promoviendo la eficacia en la ejecución de sus tareas.

2-2.3 PROCESOS DE SCRUM

El desarrollo se lleva a cabo de manera iterativa e incremental, organizado en ciclos definidos como "Sprints," que tienen una duración preestablecida, generalmente de 2 a 4 semanas. Al concluir cada Sprint, se obtiene una versión del software con nuevas funcionalidades listas para su uso. Durante cada nuevo Sprint, el equipo ajusta y mejora la funcionalidad existente, al tiempo que agrega nuevas características, priorizando aquellas que aportan un mayor valor de negocio.

El proceso de desarrollo se rige por los siguientes componentes clave:

- **Pila de Producto:** Lista de requisitos priorizados y descritos en lenguaje no técnico, ajustada regularmente.
- **Planificación del Sprint:** Reunión donde se seleccionan y organizan historias del Backlog para el Sprint.
- **Sprint:** Iteración fija donde el equipo convierte historias comprometidas en una versión funcional del software.
- **Pila del Sprint:** Lista de tareas necesarias para completar las historias del Sprint.
- **Reunión de Sprint Diario:** Encuentro breve para sincronizar actividades y discutir avances y obstáculos.
- **Demostración y Retrospectiva:** Reunión al final del Sprint para presentar historias completadas y reflexionar sobre mejoras.

2-3 LENGUAJE UNIFICADO DE MODELADO (UML)

El Lenguaje Unificado de Modelado (UML) es un lenguaje visual para especificar, visualizar y documentar sistemas de software. Propuesto por Booch, Rumbaugh y Jacobson, conceptualiza sistemas como colecciones de objetos colaborativos. Con cuatro objetivos clave, busca representar

sistemas completos, conectar conceptos modelados con componentes ejecutables, abordar la complejidad y ser utilizable por humanos y máquinas. La creación de un modelo conceptual implica bloques básicos, reglas de interacción y mecanismos comunes, proporcionando una base sólida para comunicar la estructura y el comportamiento de sistemas complejos.

Los autores tenían cuatro objetivos principales en mente al desarrollar el UML:

- Representar sistemas en su totalidad, superando la mera representación de un solo programa.
- Establecer una conexión explícita entre los conceptos modelados y los componentes ejecutables del sistema.
- Tener en cuenta la complejidad inherente a sistemas críticos y de gran escala.
- Crear un lenguaje de modelado que resulte utilizable tanto para seres humanos como para máquinas.

2-3.1 MODELOS

Un modelo es una representación, generalmente en un medio distinto al objeto o sistema que se desea representar. Esta representación busca capturar los aspectos esenciales y significativos del objeto o sistema desde un punto de vista específico, a menudo simplificando o omitiendo detalles menos relevantes.

- Modelo de Casos de Uso: Facilita la comunicación entre desarrolladores y clientes al definir actores, casos de uso y sus relaciones, estableciendo los requisitos funcionales del sistema.
- Modelo de Análisis: Organiza en paquetes el análisis más alto, abstrayendo subsistemas y capas de diseño para gestionar eficazmente un sistema complejo.
- Modelo de Diseño: Se centra en la realización física de los casos de uso, considerando requisitos funcionales y no funcionales, y sirve como base para la implementación.
- Modelo de Despliegue: Describe la distribución física del sistema entre nodos de cómputo, influyendo en el diseño y siendo crucial para la implementación.
- Modelo de Implementación: Detalla cómo los elementos del modelo de diseño se convierten en componentes de implementación, organizándolos según estructuras y modularización.

Modelo de Prueba: Enfocado en las pruebas de componentes ejecutables, asegura el correcto funcionamiento del sistema, abordando aspectos específicos que requieran pruebas particulares.

Cada uno de estos modelos cumple un papel crucial en el ciclo de desarrollo de software, abordando aspectos específicos y contribuyendo a la creación exitosa del sistema en su conjunto.

2-3.2 DIAGRAMAS

Según la Real Academia de la Lengua Española, un diagrama es un dibujo geométrico utilizado para ilustrar una proposición, resolver un problema o representar gráficamente la ley de variación de un fenómeno. A través de él, se presentan visualmente las relaciones entre las diversas partes de un conjunto o sistema.

2-3.2.1 DIAGRAMA DE CASOS DE USOS

Un diagrama de casos de uso es una representación gráfica que muestra un conjunto de casos de uso, actores y sus relaciones en un sistema. Estos diagramas capturan el comportamiento del sistema desde la perspectiva de los usuarios que interactúan con él. Esencialmente, un diagrama de casos de uso ilustra las interacciones entre el sistema, sus actores externos y los usuarios. Proporciona una visión gráfica de quiénes utilizan el sistema y cómo esperan interactuar con él.



FIGURA 4. COMPONENTES DEL DIAGRAMA DE CASOS DE USO
FUENTE. (Booch, Rambaugh, & Jacobson, 2004)

En detalle, los elementos de un diagrama de casos de uso incluyen:

- Casos de Uso: Representan requisitos funcionales y describen interacciones entre usuarios y el sistema, definiendo la interfaz externa y comportamiento funcional.
- Actores: Representan roles de usuarios, dispositivos u otros sistemas en las interacciones. Se visualizan como rectángulos con el estereotipo "actor" y se relacionan mediante asociación, generalización y uso.
- Relación de Dependencia: Indica que un cambio en un elemento puede afectar a otro, representado por una flecha discontinua orientada hacia el elemento dependiente.
- Relación de Generalización: Representa la relación entre un elemento general y un caso específico, indicando que los objetos hijos pueden usarse en cualquier lugar donde aparezca el elemento padre. Se representa con una flecha continua.
- Relación de Asociación: Denota la interacción entre un actor y un caso de uso, representada visualmente por una línea sólida. Se diferencian con flechas para indicar qué actor inicia el caso de uso.

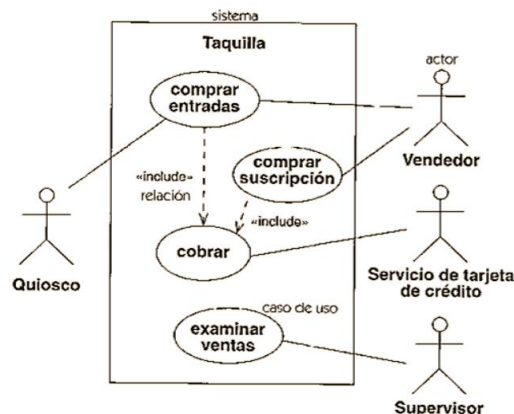


FIGURA 5. DIAGRAMA DE CASOS DE USO
FUENTE. (Booch, Rumbaugh, & Jacobson, 2004)

2-3.2.2 DIAGRAMAS DE CLASES

El diagrama de clases es una herramienta que se utiliza para representar la estructura estática de un sistema. Es importante destacar que se puede presentar a diferentes niveles de detalle, no siendo siempre necesario mostrarlo en su máxima complejidad. Este tipo de diagrama incluye la estructura de clases, que comprende sus atributos y métodos, y las relaciones que existen entre

ellas, con detalles como cardinalidad, roles, herencia y pertenencia. La representación de una clase puede variar dependiendo de la profundidad de información requerida.

La simbología en un diagrama de clases se detalla de la siguiente manera:

- Clases: Representan conjuntos de objetos con estructuras similares, incluyendo comportamiento y relaciones. Se visualizan como un rectángulo sólido con secciones para nombre, atributos y operaciones.
- Interfaces: Describen el comportamiento visible de una clase, componente o paquete. Se representan como un rectángulo etiquetado como "interface," dividiéndose en compartimentos con las operaciones soportadas.
- Relación de Dependencia: Se establece entre clases cuando un cambio en un elemento podría requerir cambios en otro elemento dependiente.
- Relación de Asociación: Representa relaciones estructurales entre clases y objetos, indicando una conexión sin especificar la duración de vida relevante.
- Relación de Generalización: Describe una relación jerárquica entre clases, donde una se considera superclase y la otra subclase. Puede haber varias subclases de una superclase.





| SÍMBOLO | SIGNIFICADO |
|--|-----------------------------|
| <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid black; margin-bottom: 2px;"><Nombre></div> <div style="border-bottom: 1px solid black; margin-bottom: 2px;"><Atributo></div> <div><Operaciones></div> </div> | Representa una Clase |
|  | Herencia |
|  | Agregación |
|  | Asociación |
|  | Dependencia o instanciación |

FIGURA 6. SIMBOLOGÍA DEL DIAGRAMA DE CLASES
FUENTE. Elaboración Propia

2-3.2.3 DIAGRAMAS DE SECUENCIA

Según Booch, Rumbaugh y Jacobson (2004), un diagrama de secuencia representa una interacción de manera gráfica y bidimensional. La dimensión vertical corresponde al eje del tiempo, donde el tiempo transcurre a través de la página. La dimensión horizontal muestra los roles que representan objetos individuales en la colaboración. Cada uno de estos roles se representa mediante una

columna vertical que contiene un símbolo de inicio y una línea vertical, a menudo llamada "lifeline."

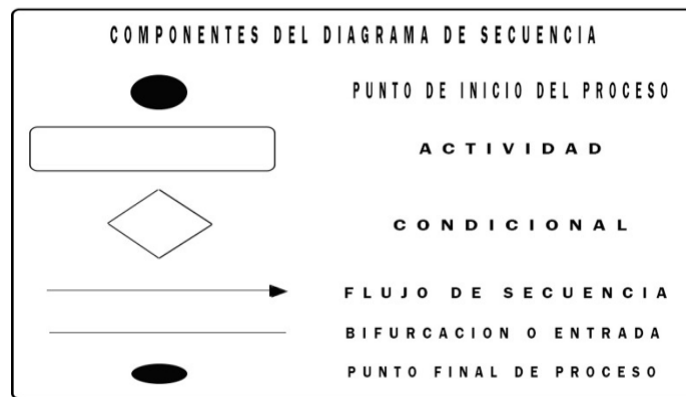


FIGURA 7. COMPONENTES DEL DIAGRAMA DE SECUENCIA
FUENTE. ELABORACIÓN PROPIA

A continuación, se detallan los componentes clave de un diagrama de secuencia:

- **Participantes:** Representan los roles u objetos que interactúan en la secuencia. Cada participante se muestra como una línea vertical que refleja su actividad en el tiempo.
- **Mensajes:** Son las interacciones o comunicaciones entre los participantes. Estos mensajes se muestran mediante flechas que indican la dirección de la comunicación y su contenido.
- **Activaciones:** Indican el período durante el cual un participante está activo o involucrado en la interacción. Se representan mediante cajas verticales adyacentes a la línea del participante.
- **Fragmentos de Interacción:** Estos fragmentos ayudan a mostrar la lógica condicional o bucles en la secuencia de interacción. Pueden incluir condiciones como "if" o "else."
- **Notas y Comentarios:** Se utilizan para proporcionar información adicional o aclaraciones en el diagrama.

2-3.2.4 DIAGRAMA DE ESTADOS

Los diagramas de estados se emplean para crear una representación visual de las máquinas de estados finitos. Una máquina de estados finitos es un modelo de comportamiento que detalla las diferentes secuencias de estados que un objeto puede experimentar a lo largo de su vida en

respuesta a eventos específicos, así como las acciones o respuestas asociadas a dichos eventos. Este enfoque permite una comprensión visual y estructurada de cómo un objeto o sistema se comporta en diferentes situaciones, lo que es fundamental para el diseño y la descripción de sistemas que implican transiciones de estado y eventos.

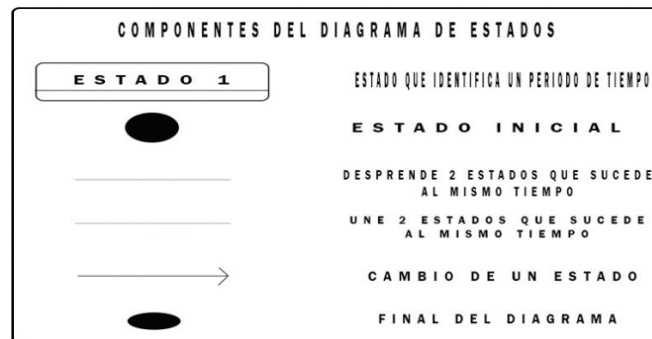


FIGURA 8. COMPONENTES DEL DIAGRAMA DE ESTADOS
FUENTE. ELABORACION PROPIA

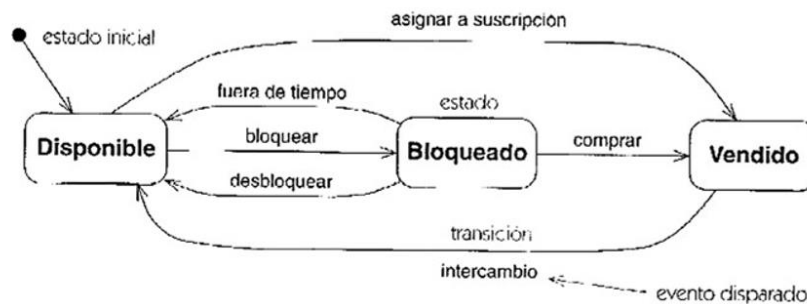


FIGURA 9. DIAGRAMA DE ESTADOS
FUENTE. (Booch, Rumbaugh, & Jacobson, 2004)

2-3.2.5 DIAGRAMA DE ACTIVIDADES

De acuerdo con la definición de Fowler (1999), un diagrama de actividades es una herramienta que se utiliza para representar un procedimiento lógico, un proceso de negocio o un flujo de trabajo. La interpretación de estos diagramas varía según su contexto en el proyecto. En las fases conceptuales, las actividades pueden considerarse como tareas que deben ser realizadas por una computadora o un ser humano.

2-4 TECNICA DE MODELADO DE OBJETOS (OMT)

OMT (Object Modeling Technique) fue desarrollada por James Rumbaugh y Michael Blaha en 1991, mientras James lideraba un equipo de investigación en los laboratorios General Electric (Medina, 2005). La esencia del análisis y diseño orientado a objetos radica en la identificación y organización de conceptos dentro del dominio de la aplicación, independientemente de su posterior implementación en un lenguaje de programación orientado a objetos (Rumbaugh & Blaha, 1991).

Para construir sistemas complejos, los desarrolladores deben abstraer distintas perspectivas del sistema, crear modelos utilizando notaciones precisas, asegurarse de que estos modelos cumplan con los requisitos del sistema y, de manera gradual, agregar detalles para transformar los modelos en una implementación concreta (Rumbaugh & Blaha, 1991).

OMT divide el ciclo de vida del software en cuatro fases consecutivas:

- **Análisis de Objetos:** En esta etapa, se enfoca en comprender y modelar el problema dentro del dominio de la aplicación.
- **Diseño del Sistema:** Aquí se determina la arquitectura del sistema en términos de subsistemas.
- **Diseño de Objetos:** Esta fase refina y optimiza el análisis de objetos para prepararlo para la implementación.
- **Implementación:** En esta etapa, se procede a la codificación y pruebas de lo diseñado previamente.

OMT, como técnica, proporciona un enfoque estructurado y sistemático para el análisis y diseño de sistemas basados en objetos, lo que contribuye a la construcción eficiente de sistemas de software robustos y de alta calidad.

2-4.1 ANALISIS DE OBJETOS

El proceso de desarrollo de software sigue un enfoque estructurado que comienza con la descripción del problema y la obtención de requisitos claros. Se procede a la creación de modelos, incluyendo el Modelo de Objetos para la estructura estática, el Modelo Dinámico que aborda el comportamiento temporal, y el Modelo Funcional que describe las funciones del sistema. Cada modelo se verifica exhaustivamente, iterando para lograr refinamientos. Este enfoque garantiza

una cobertura completa y metódica de todos los aspectos clave del sistema, contribuyendo a un desarrollo de software sólido.

2-4.1.1 MODELO DE OBJETOS

Los pasos para construir el modelo de objetos son los siguientes:

- **Identificación de Objetos y/o Clases:** En esta etapa se identifican los objetos y clases que formarán parte del sistema, considerando sus roles y responsabilidades.
- **Creación de un Diccionario de Datos:** Se elabora un diccionario que contiene información detallada sobre los objetos, clases y conceptos del sistema, incluyendo sus atributos y relaciones.
- **Identificación de Atributos y Conexiones para la Creación del Diagrama de Clases u Objetos:** Aquí se definen los atributos de los objetos y se establecen las conexiones entre ellos para representar las relaciones entre clases. Con esta información, se puede crear el diagrama de clases u objetos que muestra la estructura estática del sistema.
- **Realización de Iteraciones para el Refinamiento del Modelo:** Es importante realizar iteraciones en todo el proceso para garantizar que el modelo sea preciso y completo. Esto implica revisar y ajustar el modelo según sea necesario.

2-4.1.2 MODELO DINAMICO

Los pasos para construir el modelo dinámico son los siguientes:

- **Preparación de Escenarios de Secuencias Típicas de Iteración:** En esta etapa inicial, se preparan escenarios que representan las secuencias típicas de interacción entre los objetos dentro del sistema. Estos escenarios ayudan a comprender cómo los objetos interactúan en situaciones específicas.
- **Identificación de Sucesos que Actúan entre Objetos:** Se identifican los eventos o sucesos que ocurren entre los objetos en cada escenario. Estos sucesos representan las interacciones y acciones que los objetos realizan en respuesta a eventos.
- **Construcción de un Diagrama de Estado para Cada Objeto:** Para cada objeto que tiene un comportamiento dinámico, se crea un diagrama de estado que representa sus posibles estados y las transiciones entre ellos. Estos diagramas de estado muestran cómo un objeto responde a eventos y cambia de estado en función de su comportamiento.

- Construcción del Diagrama Global de Flujo de Sucesos (Diagrama de Secuencia): Finalmente, se construye un diagrama de secuencia que representa la secuencia global de sucesos en el sistema. Este diagrama muestra cómo los objetos interactúan a lo largo del tiempo y cómo se comunican entre sí en función de los sucesos identificados en los escenarios.

En conjunto, estos pasos permiten modelar de manera efectiva el comportamiento dinámico del sistema, lo que incluye cómo los objetos interactúan, responden a eventos y evolucionan a lo largo del tiempo. La construcción de diagramas de estado y diagramas de secuencia es fundamental para representar estos aspectos del sistema de manera clara y comprensible.

2-4.1.3 MODELO FUNCIONAL

- Identificación de Entradas y Salidas: Clarificación de datos que ingresan y salen del sistema para comprender la interacción con el entorno.
- Construcción de Diagramas de Flujo de Datos: Creación de representaciones gráficas que muestran las dependencias funcionales y transformaciones de datos dentro del sistema.
- Descripción de Funciones: Detalle de operaciones y procesos, incluyendo cálculos y transformaciones, en respuesta a los datos de entrada.
- Identificación de Restricciones: Documentación de limitaciones como tiempo, recursos o regulaciones que afectan las funciones del sistema.

Estos pasos son esenciales para comprender y representar eficazmente cómo el sistema realiza sus funciones y procesa los datos, ofreciendo una visión clara de su lógica interna y logro de objetivos.

2-5 ARQUITECTURA DEL SOFTWARE

La Arquitectura del Software, como lo define Clements en su trabajo en 1996, es un concepto fundamental en el diseño y desarrollo de sistemas informáticos. Se puede desglosar de la siguiente manera:

- Representación de Alto Nivel: Ofrece una visión simplificada de la arquitectura del software, enfocándose en componentes clave y relaciones para lograr los objetivos del sistema.
- Componentes Fundamentales: Identifica los elementos esenciales del sistema, como módulos de software o componentes de hardware, actuando como bloques de construcción.
- Comportamiento desde la Perspectiva del Sistema: Se centra en cómo interactúan y se comportan en conjunto los componentes para alcanzar los objetivos del sistema.
- Interacciones y Coordinación entre Componentes: Describe cómo los componentes se comunican y cooperan, definiendo interfaces, protocolos y flujos de datos para una colaboración efectiva.
- Cumplir la Misión del Sistema: La arquitectura del software se crea con el propósito de asegurar que el sistema cumpla su función o misión, ya sea procesamiento de datos, gestión de bases de datos u otros objetivos específicos.

2-5.1 PATRON MODELO VISTA CONTROLADOR

Patrón de arquitectura que separa datos y lógica de la interfaz de usuario. Se divide en Modelo (gestión de datos), Vista (representación de la interfaz) y Controlador (gestión de eventos). Proporciona una organización estructurada y modular para facilitar mantenimiento y escalabilidad.

2-5.1.1 EL MODELO

Cumple diversas responsabilidades, como acceder a la capa de almacenamiento de datos, definir reglas de negocio y notificar a las vistas sobre cambios en los datos. Estas funciones permiten independencia del sistema de almacenamiento y facilitan la actualización de datos por agentes externos.

2-5.1.2 LA VISTA

Responsables de recibir datos del controlador o modelo y presentarlos al usuario. Mantienen un registro del controlador asociado y ofrecen el servicio de "Actualización" para refrescar la vista en casos de cambios en datos para modelos pasivos. Se ejemplifica su uso en la navegación web.

2-5.1.3 CONTROLADOR

Responsable de recibir eventos de entrada del usuario, como clics de ratón o cambios en campos de texto. Contiene reglas de gestión de eventos, determinando acciones a ejecutar en respuesta a eventos específicos, como actualizar vistas o realizar solicitudes al modelo.

2-5.2 ARQUITECTURA DE 3 CAPAS

El Patrón de arquitectura por capas es una técnica ampliamente utilizada por arquitectos de software para estructurar sistemas de software complejos. Este enfoque conceptualiza un sistema como una serie de capas o niveles, similar a las capas de un pastel, donde cada capa se basa en la que está debajo de ella.

Los beneficios de trabajar con un enfoque por capas son:

- Permite comprender y trabajar en cada capa de manera aislada, sin necesidad de considerar las otras capas en ese momento.
- Las capas se pueden sustituir con implementaciones alternativas que ofrecen los mismos servicios básicos, lo que facilita la flexibilidad y adaptabilidad del sistema.
- Minimiza las dependencias entre las capas, lo que hace que el sistema sea más modular y fácil de mantener.
- Promueve la estandarización de servicios dentro de cada capa, lo que mejora la coherencia y la consistencia en todo el sistema.
- Una vez que se ha construido una capa, puede ser utilizada por múltiples servicios de nivel superior, lo que fomenta la reutilización de componentes.

2-5.2.1 CAPA DE PRESENTACIÓN

La interfaz de usuario se refiere a la forma en que los usuarios interactúan con el software. Puede ser tan básica como un menú de línea de comandos o tan sofisticada como una aplicación basada en formularios. Su función principal es mostrar información al usuario, interpretar los comandos que este introduce y llevar a cabo algunas validaciones sencillas de los datos ingresados.

2-5.2.2 CAPA DE REGLAS DE NEGOCIO (EMPRESARIAL)

También conocida como Lógica de Dominio, esta capa contiene la funcionalidad que implementa la aplicación. Engloba cálculos basados en la información proporcionada por el usuario y los datos almacenados, así como las validaciones necesarias. Controla la ejecución de la capa de acceso a datos y servicios externos. La lógica de la capa de negocios puede diseñarse para su uso directo por parte de los componentes de presentación o encapsularse como un servicio que se llama a través de una interfaz de servicios, coordinando así la comunicación con los clientes del servicio o invocando cualquier flujo o componente de negocio.

2-5.2.3 CAPA DE DATOS

Esta capa contiene la lógica de comunicación con otros sistemas encargados de llevar a cabo tareas dentro de la aplicación. Estos sistemas pueden ser monitores transaccionales, otras aplicaciones, sistemas de mensajería, entre otros. En el contexto de aplicaciones empresariales, esta capa suele estar representada por una base de datos, que se encarga del almacenamiento persistente de la información. Su función principal es abstraer por completo a las capas superiores, como la capa de negocio, del dialecto utilizado para comunicarse con los repositorios de datos, como PL/SQL, Transact-SQL, u otros.

2-6 SEGURIDAD

La Norma ISO 27001 es un estándar internacional de seguridad de la información que proporciona un marco para establecer, implementar, mantener y mejorar un sistema de gestión de seguridad de la información (SGSI). Algunos conceptos clave de la norma ISO 27001 que podrían estar aplicando incluyen:

- Política de Seguridad de la Información: Establecer una política que defina los objetivos de seguridad de la información y el compromiso de la alta dirección con la seguridad.
- Análisis de Riesgos: Identificar y evaluar los riesgos de seguridad de la información para determinar las medidas de seguridad adecuadas.
- Tratamiento de Riesgos: Implementar controles y medidas para mitigar los riesgos identificados.
- Documentación: Crear y mantener documentos que describan el SGSI, incluyendo la política, los procedimientos y los registros.

- Ciclo de Mejora Continua: Seguir un enfoque de mejora continua, incluyendo la revisión periódica de la eficacia de los controles y la política de seguridad de la información.
- Auditoría Interna: Realizar auditorías internas regulares para garantizar el cumplimiento de los controles y procedimientos de seguridad.
- Certificación ISO 27001: Buscar la certificación ISO 27001, que demuestra el compromiso de la organización con la seguridad de la información.
- Gestión de Incidentes de Seguridad: Implementar un proceso para la detección, gestión y respuesta a incidentes de seguridad de la información.

2-6.1 PROPIEDADES DE UN SISTEMA DE INFORMACION SEGURO

2-6.1.1 INTEGRIDAD

Este principio de seguridad informática garantiza la autenticidad y precisión de la información en cualquier momento que se solicite, asegurando que los datos no han sido alterados ni destruidos de manera no autorizada. Para prevenir y detectar posibles fallos en la integridad de los datos, el sistema debe estar equipado con mecanismos que permitan identificar y abordar los errores que puedan surgir.

2-6.1.2 CONFIDENCIALIDAD

La OCDE (Organización para la Cooperación y el Desarrollo Económico) define la confidencialidad, en sus Directrices para la Seguridad de los Sistemas de Información, como el hecho de que los datos o información estén disponibles únicamente para las personas, entidades o mecanismos autorizados, en los momentos autorizados y de una manera autorizada. Para garantizar la confidencialidad y prevenir posibles fallos, es fundamental diseñar un control de acceso al sistema que determine quién puede acceder, a qué parte del sistema, en qué momento y para qué tipo de operaciones.

2-6.1.3 DISPONIBILIDAD

La disponibilidad de la información garantiza que esta esté accesible para los usuarios autorizados cuando la necesiten, lo que implica que se pueda acceder al sistema de información en un periodo de tiempo considerado aceptable. La disponibilidad está estrechamente relacionada con la fiabilidad técnica de los componentes del sistema de información. Para asegurar la disponibilidad

de la información, se deben implementar medidas de protección de los datos, así como establecer sistemas de copias de seguridad y mecanismos para la restauración de datos que puedan haber sido dañados o destruidos, ya sea de manera accidental o intencionada.

2-6.2 MÉTODOS

2-6.2.1 CORTAFUEGOS (FIREWALL)

Un cortafuegos o firewall es la parte de un sistema informático que se utiliza para impedir un acceso a través de la red no autorizado, pero permitiendo sin problemas todos los accesos autorizados. Los cortafuegos pueden ser software o hardware. (García-Cervigón Hurtado & Alegre Ramos, 2011)

2-6.2.2 AUTENTICACIÓN Y AUTORIZACIÓN

La autenticación y la autorización están estrechamente relacionadas debido a que la autenticación es el establecimiento de confirmación para dar paso a la autorización de acceso a las zonas restringidas, realizar acciones permitidas con su correspondiente nivel de privilegio, controlar el acceso a recursos protegidos, para prevenir ataques de escalada de privilegios.

2-6.2.3 CIFRADO DE DATOS

El cifrado de datos es uno de los métodos de protección más fiables en seguridad informática. Este proceso implica la transformación de los datos de tal manera que resulten incomprensibles para cualquier persona no autorizada a acceder a ellos. El cifrado puede llevarse a cabo mediante componentes lógicos o físicos, y es esencial para garantizar la confidencialidad de la información. Sin embargo, es importante tener en cuenta que el cifrado de datos puede ser intensivo en recursos computacionales, lo que significa que puede requerir un considerable poder de procesamiento y tiempo, dependiendo de la complejidad del cifrado empleado.

2-6.2.4 SESIÓN

Una sesión en una aplicación web implica interacciones entre un cliente y un servidor. En el contexto RESTful, las sesiones se gestionan sin estado mediante tokens de acceso, como JWT. Estos tokens, emitidos tras la autenticación, transportan información de estado en cada solicitud,

eliminando la necesidad de mantener una sesión continua en el servidor y permitiendo la escalabilidad de la aplicación.

2-7 MODELO DE ESTIMACION DE COSTOS COCOMO

El Modelo de Estimación de Costos COCOMO (Constructive Cost Model) es un método ampliamente utilizado para estimar los costos, el tiempo y los recursos necesarios en el desarrollo de proyectos de software. Este modelo fue desarrollado por Barry W. Boehm en la década de 1980 y ha evolucionado en varias versiones a lo largo del tiempo. COCOMO se basa en una serie de fórmulas y parámetros que ayudan a los equipos de desarrollo de software a calcular los recursos requeridos y las estimaciones de costos con base en las características y el alcance del proyecto.

El modelo COCOMO se compone de tres niveles o modos de estimación, cada uno adaptado a diferentes niveles de detalle en la planificación del proyecto:

- **COCOMO Básico:** Este nivel de estimación se utiliza en las primeras etapas del proyecto, cuando se tienen pocos detalles específicos sobre el sistema. Se utiliza para realizar estimaciones de alto nivel y se basa principalmente en el tamaño estimado del software (en líneas de código o puntos de función).
- **COCOMO Intermedio:** En este nivel, se consideran factores adicionales, como la complejidad del proyecto, la experiencia del equipo y otras características específicas del proyecto. Estas estimaciones son más detalladas que las del COCOMO Básico y se utilizan cuando se cuenta con más información sobre el proyecto.
- **COCOMO Detallado:** Este es el nivel más detallado de estimación y se utiliza en etapas avanzadas del proyecto, cuando se cuenta con información detallada sobre los requerimientos, la arquitectura y otros aspectos del sistema. Permite realizar estimaciones muy precisas de costos y recursos.

2-8 METRICAS DE CALIDAD DEL SOFTWARE

La búsqueda de software de alta calidad es esencial en la ingeniería de software, donde la calidad se define por la capacidad de cumplir de manera confiable y eficaz con los requisitos del usuario. Este objetivo se vuelve crucial con la creciente integración de aplicaciones informáticas en nuestras vidas. La calidad del software abarca aspectos como funcionalidad, confiabilidad, rendimiento, seguridad, usabilidad y mantenibilidad, adaptándose a las necesidades específicas de

cada proyecto y cliente. En este contexto, el Modelo de Estimación de Costos COCOMO emerge como una herramienta valiosa para gerentes y equipos de desarrollo, proporcionando una metodología sólida para la estimación precisa de recursos y costos a lo largo del ciclo de vida del desarrollo de software.

2-8.1 EL ESTANDAR ISO/IEC 25010

La ISO (International Organization for Standardization) y la IEC (International Electrotechnical Commission) constituyen un sistema especializado de estandarización a nivel mundial. En el ámbito de las tecnologías de la información, ISO e IEC han colaborado en la creación de un comité técnico conjunto conocido como ISO/IEC JTC, cuya responsabilidad principal es desarrollar estándares internacionales (Pérez Medina & Sánchez, 2012).

El modelo de calidad del producto definido por la ISO/IEC 25010 está compuesto por las ocho características de calidad que se presentan en la siguiente figura:

Este modelo se ha convertido en un marco fundamental para evaluar y garantizar la calidad de productos y sistemas de software, brindando una base sólida para la estandarización y la mejora continua en el desarrollo de software y tecnologías relacionadas.



2-8.1.1 ADECUACIÓN FUNCIONAL

La característica de "Funcionalidad" representa la capacidad del producto de software para ofrecer funciones que satisfacen las necesidades explícitas e implícitas, cuando el producto se utiliza en las condiciones especificadas. Esta característica se desglosa en las siguientes subcaracterísticas:

- **Complejidad Funcional:** Esta subcaracterística evalúa en qué medida el conjunto de funcionalidades del producto abarca todas las tareas y objetivos especificados por el usuario.
- **Corrección Funcional:** La corrección funcional se refiere a la capacidad del producto o sistema para proporcionar resultados correctos con el nivel de precisión requerido.
- **Pertinencia Funcional:** Evalúa la capacidad del producto de software para ofrecer un conjunto adecuado de funciones que se ajusten a las tareas y objetivos específicos del usuario.

2-8.1.2 EFICIENCIA DE DESEMPEÑO

La característica de "Eficiencia del Desempeño" se refiere al rendimiento del software en relación a la cantidad de recursos utilizados en condiciones específicas. Esta característica se divide en las siguientes subcaracterísticas:

- **Comportamiento Temporal:** Evalúa los tiempos de respuesta y procesamiento, así como las tasas de transferencia de datos de un sistema cuando realiza sus funciones en condiciones predefinidas en comparación con un conjunto de pruebas de referencia (benchmark).
- **Utilización de Recursos:** Mide las cantidades y tipos de recursos que el software utiliza al llevar a cabo sus funciones en condiciones determinadas.
- **Capacidad:** Se refiere al grado en que los límites máximos de los parámetros de un producto o sistema de software cumplen con los requisitos especificados.

2-8.1.3 COMPATIBILIDAD

La característica de "Compatibilidad" se refiere a la capacidad de dos o más sistemas o componentes para intercambiar información y realizar sus funciones requeridas cuando comparten el mismo entorno hardware o software. Esta característica se divide en las siguientes subcaracterísticas:

- **Coexistencia:** Evalúa la capacidad del producto para operar en conjunto con otro software independiente en un entorno compartido, compartiendo recursos comunes sin afectar negativamente su funcionamiento.

- Interoperabilidad: Mide la capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada de manera efectiva.

2-8.1.4 USABILIDAD

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Capacidad para reconocer su adecuación: capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Capacidad de aprendizaje: capacidad del producto que permite al usuario aprender su aplicación.
- Capacidad para ser usado: capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario: capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario: capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- Accesibilidad: capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

2-8.1.5 FIABILIDAD

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Madurez: capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- Disponibilidad: capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- Tolerancia a fallos: capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

- Capacidad de recuperación: capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo. (iso 25000, 2015)

2-8.1.6 SEGURIDAD

La característica de "Seguridad" se refiere a la capacidad del software para proteger la información y los datos de manera que personas o sistemas no autorizados no puedan acceder a ellos o modificarlos. Esta característica se desglosa en las siguientes subcaracterísticas:

- Confidencialidad: Capacidad del sistema para proteger los datos e información contra el acceso no autorizado, ya sea de manera accidental o intencionada.
- Integridad: Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizadas en los datos o programas de computadora.
- No repudio: Capacidad del sistema para demostrar las acciones o eventos que han tenido lugar, de modo que dichas acciones o eventos no puedan ser negadas posteriormente.
- Responsabilidad: Capacidad del sistema para rastrear de manera inequívoca las acciones de una entidad.
- Autenticidad: Capacidad del sistema para demostrar la identidad de un sujeto o un recurso.

2-8.1.7 MANTENIBILIDAD

La característica de "Mantenibilidad" se refiere a la capacidad del producto software para ser modificado de manera efectiva y eficiente, ya sea para abordar necesidades evolutivas, correcciones de errores o mejoras. Esta característica se desglosa en las siguientes subcaracterísticas:

- Modularidad: Capacidad de un sistema o programa de computadora, compuesto por componentes discretos, que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad: Capacidad de un activo que permite que sea utilizado en más de un sistema de software o en la construcción de otros activos.
- Analizabilidad: Facilidad con la que se puede evaluar el impacto de un cambio específico en el software en el resto del sistema, diagnosticar deficiencias o causas de fallos en el software, o identificar las partes que requieren modificaciones.

- Capacidad para ser modificado: Capacidad del producto que permite que sea modificado de manera efectiva y eficiente sin introducir defectos o degradar el rendimiento.
- Capacidad para ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y llevar a cabo pruebas para determinar si se cumplen dichos criterios.

2-8.1.8 PORTABILIDAD

La característica de "Portabilidad" se refiere a la capacidad del producto o componente de software para ser transferido de manera efectiva y eficiente entre diferentes entornos de hardware, software, operacionales o de uso. Esta característica se desglosa en las siguientes subcaracterísticas:

- Adaptabilidad: Capacidad del producto que le permite ser adaptado de manera efectiva y eficiente a diferentes entornos específicos de hardware, software, operaciones o uso.
- Capacidad para ser instalado: Facilidad con la que el producto se puede instalar y/o desinstalar con éxito en un entorno determinado.
- Capacidad para ser reemplazado: Capacidad del producto para ser utilizado en lugar de otro producto de software específico con el mismo propósito y en el mismo entorno.

2-9 HERRAMIENTAS DE DESARROLLO WEB

2-9.1 SISTEMA GESTOR DE BASE DE DATOS MYSQL

Sistema de gestión de bases de datos (DBMS) líder y ampliamente utilizado con arquitectura cliente-servidor. Compatible con SQL, ofrece transacciones, escalabilidad, soporte multiplataforma y diversos motores de almacenamiento. Utilizado en aplicaciones web, CMS, sistemas empresariales y más, se integra con varios lenguajes y herramientas de desarrollo. Dispone de ediciones como la gratuita Community Edition y la Enterprise para entornos empresariales, respaldadas por una activa comunidad y soporte en línea.

2-9.1.1 FUNCIONAMIENTO

Una base de datos MySQL opera mediante un servidor central que gestiona solicitudes de clientes. Los clientes, que pueden ser aplicaciones web o de escritorio, se conectan al servidor a través de conexiones de red. MySQL admite varios motores de almacenamiento, destacando InnoDB por su soporte de transacciones. Los clientes envían comandos SQL al servidor, que crea un plan de

ejecución para acceder y manipular datos. La gestión de conexiones, seguridad, transacciones y administración de la base de datos son esenciales. La optimización y el mantenimiento aseguran un rendimiento eficiente.

2-9.1.2 BASE DE DATOS RELACIONAL

Una base de datos relacional, desde la perspectiva del usuario, puede describirse como un conjunto de tablas interconectadas que permiten el almacenamiento de información con el propósito de acceder y utilizar estos datos en el futuro. Estas bases de datos se fundamentan en el modelo de datos relacional para organizar y gestionar las tablas, el cual, a su vez, se apoya en conceptos derivados de la teoría de conjuntos para establecer relaciones entre los datos. Este enfoque estructurado y basado en tablas proporciona un medio efectivo para almacenar y recuperar información de manera lógica y coherente, siguiendo las formas normales de base de datos, como la Primera Forma Normal (1NF), la Segunda Forma Normal (2NF) y la Tercera Forma Normal (3NF). Estas formas normales son esenciales para garantizar la integridad y eficiencia de los datos, evitando la redundancia y asegurando una gestión óptima de la información.

2-9.2 LENGUAJE DE PROGRAMACION PHP ("PREPROCESADOR DE HIPERTEXTO")

Lenguaje de programación de código abierto, interpretado en tiempo real, integrado directamente en HTML y ejecutado en el servidor. A diferencia de lenguajes de script como JavaScript, PHP es de servidor, lo que oculta su código fuente en el navegador del cliente. Con una sintaxis similar a C, se utiliza en desarrollo web, scripts de línea de comandos y aplicaciones de interfaz gráfica, destacando por su versatilidad y amplio uso.

2-9.3 MARCO DE DESARROLLO YIIFRAMEWORK II

Framework PHP de código abierto, Yii2, destaca por su enfoque en seguridad, eficiencia y facilidad de uso. Con un patrón MVC, herramientas integradas de seguridad, rendimiento optimizado, soporte AJAX, generación automática de código y una comunidad activa, Yii2 es una opción robusta para el desarrollo eficiente de aplicaciones web de alta calidad.

2-9.4 CARACTERISTICAS GENERALES DE ANGULAR

Framework versátil para desarrollo web y móvil con arquitectura basada en componentes, TypeScript, enrutamiento SPA, inyección de dependencias, directivas personalizadas, manejo de formularios, comunicación con el servidor, pruebas unitarias y E2E, optimización de rendimiento y amplio soporte de la comunidad. Angular ofrece una estructura sólida, herramientas integradas y flexibilidad para construir aplicaciones web, móviles y de escritorio.

2-9.5 CARACTERISTICAS GENERALES DE IONIC

Marco de desarrollo para aplicaciones móviles híbridas, utiliza tecnologías web estándar (HTML, CSS, JavaScript) y se integra con Angular. Características destacadas: desarrollo multiplataforma, interfaz atractiva, acceso a capacidades nativas, desarrollo rápido con CLI, soporte PWA, comunidad activa, ecosistema de plugins, herramientas en tiempo real, personalización, facilidad de prueba y rendimiento optimizado. Ionic simplifica la creación de aplicaciones móviles con una apariencia moderna y funcionalidad nativa.

2-9.6 CARACTERISTICAS GENERALES DE CAPACITOR

Facilita la creación de aplicaciones móviles nativas con tecnologías web estándar (HTML, CSS, JavaScript). Características clave incluyen multiplataforma, acceso a API nativas, integración de plugins, personalización visual, soporte PWA, desarrollo en tiempo real, CLI, implementación sencilla, comunidad activa y código abierto. Capacitor permite a los desarrolladores web aplicar sus conocimientos para crear aplicaciones móviles, siendo una herramienta versátil y gratuita.

2-9.7 HTML (LENGUAJE DE MARCADO DE HIPERTEXTO)

Lenguaje estándar para crear páginas y aplicaciones web. Utiliza etiquetas angulares para estructurar contenido. Compuesto por elementos como encabezados, párrafos, imágenes y formularios. Permite enlaces, multimedia, formularios interactivos y metadatos. Recomendado junto con CSS para control avanzado de presentación. Compatible con la mayoría de navegadores. Evolucionado a HTML 5, es la base de la programación web junto con CSS y JavaScript.

2-9.8 CSS (STYLE)

CSS (Cascading Style Sheets - Hojas de Estilo en Cascada) es un lenguaje de diseño ampliamente utilizado en desarrollo web para controlar la apariencia y presentación visual de páginas y

aplicaciones web. Su concepto central es separar la estructura (HTML) del contenido del diseño y estilo, lo que ofrece flexibilidad, consistencia y mantenibilidad en el diseño web.

2-9.8 JAVASCRIPT:

JavaScript es un lenguaje de programación esencial en el desarrollo web, destinado a añadir interactividad y dinamismo a las páginas web. Se trata de un lenguaje de secuencias de comandos interpretado por el navegador del usuario, lo que significa que se ejecuta en su dispositivo.

2-9.8.1 JQUERY:

jQuery es una biblioteca de JavaScript de código abierto que simplifica la manipulación del DOM y la interacción con JavaScript. Proporciona una serie de funciones y métodos predefinidos para realizar tareas comunes de manera más sencilla y eficiente.

2-9.8.2 AJAX (ASYNCHRONOUS JAVASCRIPT AND XML):

Técnica de desarrollo web con JavaScript que realiza solicitudes asincrónicas al servidor sin recargar la página. A pesar de su nombre, puede manejar diferentes formatos como JSON. Características clave: solicitudes asincrónicas, carga selectiva de partes de una página, mejora la interfaz de usuario, comunica con el servidor sin recargar la página, amplia compatibilidad con navegadores. Es esencial en aplicaciones web modernas para mejorar la experiencia del usuario.

2-9.8.3 JQUERY UI

Biblioteca de interfaz de usuario (UI) de código abierto basada en jQuery, simplifica la creación de interfaces interactivas en aplicaciones web. Ofrece widgets personalizables, interacción con el usuario, temas, efectos visuales y compatibilidad con navegadores. Mejora la usabilidad y experiencia del usuario, siendo valiosa en el desarrollo web.

2-9.9 BOOTSTRAP

Marco de diseño frontend de código abierto, mantenido por la comunidad. Facilita la creación de sitios web responsivos con sistema de rejilla, componentes CSS y JavaScript. Destaca por su fácil uso y personalización a través de temas, sin necesidad de escribir CSS o JavaScript personalizado. Solo se incorporan los archivos en el código HTML.

CAPITULO III

3.1 TIPO DE INVESTIGACIÓN

El presente proyecto se fundamenta en una metodología de investigación descriptiva. A través de esta metodología, nuestro objetivo es identificar y detallar minuciosamente las características más relevantes que se observen en la recopilación de datos. Esta recopilación se llevará a cabo mediante entrevistas, encuestas y observaciones. El enfoque descriptivo nos permitirá comprender en profundidad la dinámica y los procesos que subyacen en el funcionamiento de las agencias de turismo, identificando de manera precisa los aspectos clave que requerirán nuestra atención para la implementación exitosa del sistema de información.

3.2 DISEÑO DE LA INVESTIGACIÓN

En el marco de nuestro proyecto para la agencia de turismo "JIWAKI", hemos decidido adoptar una metodología de investigación mixta que combina enfoques cuantitativos y cualitativos. Este enfoque mixto es la elección idónea debido a la complejidad y la diversidad de aspectos que caracterizan el funcionamiento de nuestra agencia.

La parte cuantitativa de nuestra investigación se basará en la recopilación de datos numéricos a través de encuestas, lo que nos permitirá obtener información cuantitativa precisa sobre varios aspectos de nuestras operaciones, como la satisfacción de nuestros clientes y la eficacia de nuestros procedimientos.

La componente cualitativa se centrará en el análisis subjetivo y se llevará a cabo a través de entrevistas en profundidad con expertos que tienen un profundo conocimiento en el ámbito de las agencias de turismo. Estas entrevistas nos brindarán perspectivas cualitativas valiosas que complementarán y enriquecerán nuestra comprensión de los desafíos y oportunidades que enfrentamos en "JIWAKI".

La combinación de estos métodos cuantitativos y cualitativos nos permitirá obtener una visión integral y equilibrada de nuestra realidad operativa en "JIWAKI". Esto, a su vez, facilitará la toma de decisiones informadas y la implementación efectiva de soluciones en nuestro proyecto.

3.3 SELECCIÓN DE LA POBLACIÓN

3.3.1 UNIVERSO

El universo de estudio en nuestro proyecto se refiere a la población o conjunto de unidades que son objeto de investigación y que podrían ser observadas individualmente en el estudio. Basándonos en la formulación del problema, hemos identificado a los principales involucrados en nuestro proyecto, que son los miembros del equipo de administración de la agencia de turismo "JWAKI". La principal fuente de información para nuestro estudio proviene del área de Administración.

El universo considerado para este proyecto de grado incluye al personal de administración, que abarca al gerente (dueño), dos subadministradores para los turnos de la mañana y la tarde, tres guías turísticos y dos conductores de vehículos. Además, hemos tenido en cuenta a los clientes registrados en nuestra base de datos. Es importante destacar que el tamaño del universo de clientes es variable y puede considerarse como indeterminado debido al flujo constante de clientes que experimenta la agencia de turismo " JWAKI ". Por lo tanto, se trata de un conjunto en constante cambio y crecimiento.

3.3.2 MUESTRA

En el marco de nuestro proyecto en la agencia de turismo "JIWAKY", el muestreo se refiere al proceso de seleccionar una muestra representativa del universo de estudio. Dado que el universo de estudio incluye al personal de administración, los guías turísticos, los conductores de vehículos y los clientes registrados, es fundamental seleccionar una muestra que refleje con precisión la diversidad y las características de estos grupos.

Para llevar a cabo el muestreo, utilizaremos un enfoque estratificado. Esto implica dividir el universo en grupos o estratos, como el personal de administración, los guías turísticos, los conductores y los clientes. Luego, seleccionaremos una muestra de cada estrato de manera que cada grupo esté representado en proporción a su tamaño en el universo. Esto garantizará que obtengamos una muestra equitativa y representativa.

En el caso de los clientes registrados, dado que el tamaño de este grupo es variable y puede ser considerado como indeterminado debido al flujo constante de clientes, utilizaremos un enfoque de muestreo por conveniencia. Seleccionaremos clientes aleatoriamente en momentos específicos para capturar una variedad de opiniones y experiencias.

El proceso de muestreo es esencial para garantizar que los resultados de nuestro estudio sean aplicables y representativos de la población en "JIWAKY". Una vez que se haya recopilado la muestra, procederemos con la recopilación de datos a través de encuestas, entrevistas y observaciones, lo que nos permitirá obtener información valiosa para el proyecto.

3.4 INSTRUMENTOS DE RECOLECCIÓN DE DATOS

En el proceso de investigación para nuestro proyecto en la agencia de turismo "JIWAKY", utilizaremos una variedad de instrumentos de recolección de datos para obtener información relevante y valiosa sobre el funcionamiento y los desafíos que enfrentamos. Los instrumentos de recolección de datos desempeñan un papel crucial en la obtención de información precisa y enriquecedora para nuestro estudio.

3.4.1 ENTREVISTA

La entrevista se presenta como un instrumento fundamental en nuestro proceso de investigación. Nos permitirá mantener un diálogo interactivo con los miembros del equipo de administración, guías turísticos, conductores de vehículos y expertos en el campo del turismo. A través de entrevistas en profundidad, podremos explorar las perspectivas subjetivas de los involucrados, comprender sus experiencias, identificar desafíos específicos y recopilar información cualitativa valiosa que complementará nuestros datos cuantitativos.

3.4.2 OBSERVACIÓN

La observación directa también desempeñará un papel importante en nuestro proyecto. Al observar los procedimientos y procesos en "JIWAKY", podremos documentar de manera objetiva cómo se llevan a cabo las operaciones diarias. Esto nos permitirá identificar posibles áreas de mejora, comprender la dinámica interna y garantizar que nuestra investigación esté respaldada por evidencia visual.

3.4.3 ENCUESTA

La encuesta es otro instrumento crucial que emplearemos para recopilar datos cuantitativos. Se dirigirá tanto al personal de administración como a los clientes registrados. A través de encuestas estructuradas, obtendremos información numérica y objetiva sobre la satisfacción de los clientes, la eficacia de los procesos y la percepción del personal. Las encuestas nos proporcionarán datos que respaldarán nuestra toma de decisiones basada en evidencia.

Estos instrumentos de recolección de datos se utilizarán en conjunto para obtener una visión completa y equilibrada de la realidad operativa de "JIWAKY". A través de entrevistas, observación y encuestas, buscaremos comprender a fondo los desafíos y oportunidades que enfrentamos y garantizar que nuestro proyecto se base en datos confiables y representativos.

CAPITULO VI

4.1 RESULTADOS DEL RELEVAMIENTO DE INFORMACIÓN:

El aporte del presente proyecto, es el de automatizar los procesos manuales de las compras y ventas para ayudar al personal a tener un mejor control de las mismas, aumentando la productividad de la empresa. El sistema coadyuva la buena toma de decisiones y a la planificación de gestiones posteriores en base a los clientes, elaborando reportes e historiales confiables y precisos hacia el personal que es parte de la empresa.

4.1.1 ENTREVISTA

Al preguntar sobre los principales desafíos que enfrenta actualmente el personal en el proceso manual, se identificaron áreas clave que podrían beneficiarse de la automatización:

- **Automatizar Reservas:** El personal expresó desafíos en la gestión de reservas manuales, señalando la necesidad de una herramienta que simplifique y optimice este proceso, mejorando la eficiencia y reduciendo posibles errores.
- **Automatizar Reportes:** La generación manual de informes fue identificada como una tarea laboriosa y propensa a errores. La automatización de la creación de informes se considera esencial para proporcionar datos precisos de manera más eficiente.
- **Automatizar Procesos:** Se resaltó la necesidad de automatizar procesos operativos clave más allá de las transacciones financieras, buscando mejorar la eficiencia general y reducir la carga de trabajo manual.

4.1.2 OBSERVACION

Durante la observación, surgió la imperiosa necesidad de implementar todo lo descrito anteriormente que aborde todos los aspectos cruciales para optimizar el funcionamiento eficiente y facilitar la mejora continua de la agencia de turismo. Asimismo, se destacó la necesidad de abordar específicamente ciertos aspectos. Entre los aspectos clave identificados se encuentran:

- **Apoyo al Marketing:** Se identificó la necesidad de integrar funciones que respalden las estrategias de marketing, como la creación de publicaciones, promociones y campañas, para mejorar la adquisición y retención de clientes.

- **Automatización de Procesos para la Satisfacción del Cliente:** La observación resaltó la importancia de automatizar procesos no solo para la eficiencia interna, sino también para mejorar la experiencia del cliente. La automatización se percibe como un medio para proporcionar un servicio más rápido, preciso y personalizado.
- **Interacción con Clientes:** Se notó la necesidad de una herramienta que facilite la interacción con los clientes, ya sea a través de notificaciones automáticas, seguimiento de preferencias o proporcionando información relevante de manera proactiva.
- **Feedback del Cliente:** La observación reveló la importancia de recopilar y analizar de manera sistemática el feedback del cliente, subrayando la necesidad de integrar funciones que faciliten la retroalimentación y mejoren la calidad del servicio.

En resumen, la observación subraya la importancia de una solución integral que no solo automatice transacciones financieras, sino que también aborde aspectos críticos para el éxito del negocio, como el apoyo a el marketing y la mejora continua de la satisfacción del cliente a través de la automatización de procesos.

4.1.3 ENCUESTA

En base a las respuestas recopiladas, el resultado para la pregunta "**¿Cómo percibe la eficiencia actual en la gestión de reservas y transacciones financieras en la agencia de turismo JIWAKY?**" es el siguiente:

- a) Ineficiente: 80%
- b) Neutral: 10%
- c) Eficiente: 5%
- d) Muy ineficiente: 3%
- e) Muy eficiente: 2%

En base a las respuestas recopiladas para la pregunta "**¿Cómo percibe la importancia de la automatización en la mejora continua de la experiencia turística ofrecida por JIWAKY?**", los resultados son los siguientes:

- a) Muy importante: 75%

- b) Importante: 15%
- c) Neutral: 5%
- d) Poco importante: 3%
- e) No importante en absoluto: 2%

En base a las respuestas recopiladas para la pregunta "**¿Cómo evalúa la necesidad de integrar funciones de apoyo al marketing en el sistema de la agencia para fortalecer la adquisición y retención de clientes?**". En base a las respuestas recopiladas:

- a) Muy necesaria: 70%
- b) Necesaria: 20%
- c) Neutral: 5%
- d) Poco necesaria: 3%
- e) No necesaria en absoluto: 2%

En base a las respuestas recopiladas para la pregunta "**¿Cuál cree que sería el impacto más positivo de la automatización en la satisfacción del cliente?**". En base a las respuestas recopiladas:

- a) Mayor rapidez en los servicios: 65%
- b) Reducción de errores en reservas y transacciones: 20%
- c) Mayor personalización en la atención: 8%
- d) Facilitación de la interacción con el cliente: 5%
- e) Otra (por favor, especificar): 2%

En base a las respuestas recopiladas para la pregunta "**¿Cómo percibe la importancia de la automatización en la mejora continua de la experiencia turística ofrecida por JIWAKY?**". En base a las respuestas recopiladas:

- a) Muy importante: 75%

- b) Importante: 15%
- c) Neutral: 5%
- d) Poco importante: 3%
- e) No importante en absoluto: 2%

4.2 IDENTIFICACIÓN DE LA METODOLOGÍA DE LA PROPUESTA

La identificación de la metodología para la propuesta de solución, que implica la implementación de un sistema de información para abordar los problemas descritos anteriormente en la agencia de turismo JIWAKY, podría seguir el siguiente enfoque:

Metodología Propuesta: Desarrollo Iterativo e Incremental (DII)

Análisis y Requerimientos:

- Realizar un análisis exhaustivo de los problemas identificados.
- Entrevistas con el personal de la agencia para comprender sus necesidades.
- Identificar requisitos específicos para cada módulo del sistema.

Diseño:

- Desarrollar prototipos y maquetas para obtener retroalimentación temprana.
- Diseñar la arquitectura del sistema, considerando la integración de módulos.
- Especificar interfaces de usuario intuitivas y amigables.

Desarrollo Iterativo:

- Implementar el sistema en iteraciones funcionales.
- Priorizar módulos críticos para abordar problemas más urgentes.
- Realizar pruebas continuas para asegurar la calidad del código.

Evaluación Continua:

- Obtener retroalimentación constante del personal y los usuarios.
- Realizar pruebas de usabilidad para asegurar la facilidad de uso.
- Ajustar y mejorar continuamente basándose en los comentarios recibidos.

Implementación Gradual:

- Implementar el sistema en fases, comenzando con módulos críticos.
- Proporcionar capacitación al personal para la transición suave.
- Garantizar una gestión del cambio efectiva.
- Monitoreo y Mejora Continua:

4.3 DESARROLLO DE LA PROPUESTA

Implementación de un Sistema de Información Integral para JIWAKY

Relevamiento de Requerimientos:

1. Se llevaron a cabo entrevistas con el personal de JIWAKY para identificar desafíos y necesidades específicas.
2. Se observaron los procesos manuales actuales, destacando áreas de ineficiencia y posibles mejoras.
3. Se diseñó una encuesta para recopilar percepciones y opiniones adicionales sobre la necesidad de automatización.

Metodología Propuesta: Desarrollo Iterativo e Incremental (DII):**Análisis y Requerimientos:**

1. Identificación detallada de los problemas, desde la gestión de clientes hasta la entrega de multimedia.
2. Definición de requisitos específicos para cada módulo del sistema.

Diseño:

1. Desarrollo de prototipos interactivos para obtener retroalimentación temprana.

2. Diseño de una arquitectura modular que aborda problemas críticos identificados.

Desarrollo Iterativo:

1. Implementación de módulos clave, priorizando aquellos que abordan desafíos urgentes.
2. Pruebas continuas para garantizar la calidad y funcionalidad del código.

Evaluación Continua:

1. Sesiones de retroalimentación con el personal y usuarios para ajustar el diseño y la funcionalidad.
2. Pruebas de usabilidad para garantizar la facilidad de uso y la aceptación del sistema.

Implementación Gradual:

1. Implementación por fases, comenzando con módulos críticos como el registro y gestión de clientes.
2. Entrenamiento del personal para asegurar una transición suave de los procesos manuales al sistema.

Monitoreo y Mejora Continua:

1. Establecimiento de un sistema de monitoreo para evaluar el rendimiento en tiempo real.
2. Identificación y abordaje de áreas de mejora mediante retroalimentación de usuarios.

Documentación Exhaustiva:

1. Creación de manuales de usuario detallados y materiales de capacitación.
2. Documentación técnica que abarca desde la arquitectura hasta los procedimientos de mantenimiento.

Garantía de Calidad:

1. Implementación de pruebas exhaustivas de seguridad y rendimiento.
2. Auditorías regulares de código para garantizar la robustez del sistema.

Resultado Esperado:

La implementación de esta propuesta conducirá a un sistema integral que aborda los desafíos identificados, mejorando la eficiencia operativa, la toma de decisiones y la satisfacción del cliente en la agencia de turismo JIWAKY. La metodología DII permite una adaptación ágil a los cambios y una mejora continua para mantenerse alineado con las necesidades cambiantes del negocio y del mercado.

4.4 ESTIMACIÓN DE COSTOS DE DESARROLLO (ESTUDIO DE COSTOS, ANÁLISIS VAN Y TIR)

Estimación de Costos de Desarrollo para el Sistema de Administración y Aplicación Móvil de JIWAKY

Desarrollo del Sistema de Administración:

- Desarrolladores Front-end y Back-end: \$2,500
- Diseñadores de UX/UI: \$1,000
- Especialistas en Bases de Datos: \$1,200
- Gerente de Proyecto: \$1,500
- Licencias de software: \$300
- Plataformas de desarrollo: \$400
- Cursos de formación: \$700
- Servidores y alojamiento en la nube: \$1,200
- **Costo Total: \$8,800**

Desarrollo de la Aplicación Móvil:

- Desarrolladores de aplicaciones móviles: \$2,500
- Diseñadores de interfaz móvil: \$1,000
- Especialistas en integración de servicios móviles: \$1,200
- Licencias de desarrollo para plataformas móviles: \$500

- Herramientas de diseño para aplicaciones móviles: \$100
- **Costo Total: \$5,300**

Costo Total del Proyecto: \$14,100

2. Análisis de Valor Actual Neto (VAN):

Inversiones Iniciales:

- Desarrollo del sistema de administración.
- Desarrollo de la aplicación móvil.
- Infraestructura y herramientas tecnológicas.

Flujo de Efectivo Neto:

- Ahorro en costos operativos debido a la eficiencia mejorada.
- Incremento de ingresos por la mejora en la adquisición y retención de clientes.

Tasa de Descuento:

- Aplicar una tasa de descuento conservadora para evaluar los flujos de efectivo futuros.

3. Tasa Interna de Retorno (TIR):

Rentabilidad del Proyecto:

- Evaluar la tasa de rendimiento del proyecto en comparación con la tasa de descuento.

4. Evaluación Económica:

Beneficios Tangibles:

- Reducción de costos operativos.
- Incremento de ingresos por mejoras en el servicio al cliente.

Beneficios Intangibles:

- Mejora en la imagen y competitividad de la agencia.
- Mayor satisfacción del cliente.

5. Consideraciones Adicionales:

Costos de Mantenimiento:

- Presupuestar para actualizaciones, correcciones y soporte continuo.

Contingencias:

- Reservar un porcentaje para posibles imprevistos o cambios en los requisitos.

Conclusión:

La estimación de costos debe abordar tanto los aspectos tangibles como los intangibles del proyecto. Un análisis financiero exhaustivo, incluyendo el VAN y la TIR, proporcionará una visión clara de la viabilidad económica del proyecto, considerando tanto la inversión inicial como los beneficios a largo plazo.

4.5 CÁLCULO DEL VAN Y TIR

4.5.1 VALOR ACTUAL NETO (VAN)

Si los beneficios netos son constantes y asumimos que cada año generan \$4,000, entonces el cálculo del Valor Actual Neto (VAN) con una tasa de descuento del 10% para un período de dos años sería el siguiente:

$$VAN = \frac{\text{Flujo de efectivo neto anual}}{(1 + \text{tasa de descuento})^{\text{número de años}}}$$

$$VAN = \frac{\$4,000}{(1 + 0.10)^2}$$

$$VAN \approx \frac{\$4,000}{1.21}$$

$$VAN \approx \$3,305.79$$

4.5.2 TASA INTERNA DE RETORNO (TIR)

Dado que estamos usando un flujo de efectivo constante de \$6,000 anuales.

Vamos a probar con un flujo de efectivo neto anual de \$8,000. La ecuación sería entonces:

$$(1 + TIR)^2 = \frac{\$8,000}{\$10,000}$$

Resolviendo esta ecuación para TIR , obtenemos:

$$(1 + TIR)^2 = 0.8$$

$$1 + TIR = \sqrt{0.8}$$

$$TIR = \sqrt{0.8} - 1$$

$$TIR \approx 0.3410$$

La viabilidad financiera del proyecto queda respaldada por el resultado positivo obtenido en la Tasa Interna de Retorno (TIR). El valor positivo de la TIR, aproximadamente del 34.10%, indica que el proyecto tiene la capacidad de generar rendimientos financieros superiores a la tasa de descuento utilizada. Esta conclusión sugiere que la inversión propuesta es económicamente atractiva y puede generar beneficios significativos durante el periodo de evaluación de dos años.

4.6 CONTROL DE CALIDAD (APLICACIÓN DE NORMAS DE CALIDAD)

El desarrollo del sistema de gestión para la agencia de turismo JIWAKY se ha llevado a cabo bajo estrictos estándares de calidad, incluyendo la verificación de las normas ISO pertinentes. A continuación, se detallan los resultados de la verificación de estándares ISO en las áreas clave:

ISO 9001 - Gestión de Calidad:

Se implementaron procesos y procedimientos de gestión de calidad para garantizar que el desarrollo del sistema cumpla con los requisitos y expectativas del cliente. La documentación de los procesos y la atención a la mejora continua son aspectos fundamentales de esta norma, y se han seguido rigurosamente.

ISO 27001 - Seguridad de la Información:

Se establecieron políticas y medidas de seguridad de la información para proteger los datos confidenciales. La gestión de riesgos, el control de accesos y la continuidad del negocio se abordaron según los requisitos de la norma.

ISO 25010 - Calidad del Producto Software:

La calidad del producto software se evaluó en términos de funcionalidad, fiabilidad, usabilidad, eficiencia y mantenibilidad. Las pruebas realizadas abarcaron todas estas dimensiones para garantizar un producto final de alta calidad.

4.7 PRUEBAS DE SOFTWARE

4.7.1 PRUEBAS DE CAJA NEGRA

Este informe detalla las pruebas de caja negra realizadas tanto en el sistema de administración como en la aplicación móvil desarrollados para la agencia de turismo JIWAKY. Estas pruebas se llevaron a cabo con el objetivo de garantizar la calidad, usabilidad y confiabilidad de ambas soluciones en el contexto del proyecto académico.

Objetivo de las Pruebas:

El enfoque principal de las pruebas fue evaluar la funcionalidad de ambas soluciones desde la perspectiva del usuario final, asegurando que cumplan con los requisitos del proyecto y brinden una experiencia satisfactoria.

Procedimientos de Prueba:

Se diseñaron casos de prueba específicos para abordar cada funcionalidad clave tanto del sistema de administración como de la aplicación móvil. Estos casos se basaron en los requisitos del proyecto, las especificaciones de ambas soluciones y las mejores prácticas de desarrollo. Durante

las pruebas, se simularon diversas interacciones de usuario para evaluar la usabilidad y el rendimiento.

Resultados de las Pruebas:

Los resultados de las pruebas de caja negra para ambas soluciones indican un rendimiento positivo y funcionalidades sólidas:

Sistema de Administración:

Se verificó la eficiencia en la automatización de procesos manuales, incluyendo compras, ventas, toma de decisiones y generación de reportes.

Aplicación Móvil:

La aplicación móvil se evaluó positivamente en aspectos como accesibilidad universal, tecnologías de almacenamiento local, integración de servicios de mapas, uso de formatos de datos modernos y la interacción entre cliente, guía y administrador.

Conclusiones:

Las pruebas de caja negra confirman la calidad y eficacia tanto del sistema de administración como de la aplicación móvil desarrollados en el marco del proyecto de grado. No se identificaron problemas significativos durante las pruebas, validando el trabajo realizado.

Recomendaciones:

Se sugiere realizar pruebas de usuario final para evaluar la experiencia completa del usuario y abordar cualquier retroalimentación específica.

4.7.2 PRUEBAS DE CAJA BLANCA

Este informe detalla las pruebas de caja blanca realizadas en el sistema de administración y la aplicación móvil desarrollados para la agencia de turismo JIWAKY. Estas pruebas se llevaron a cabo con el objetivo de evaluar la estructura interna del código y garantizar la calidad del software desde una perspectiva técnica.

Objetivo de las Pruebas:

El objetivo principal de las pruebas de caja blanca fue evaluar la estructura del código, identificar posibles vulnerabilidades y asegurar que el software cumple con los estándares de programación y las mejores prácticas.

Procedimientos de Prueba:

Se examinó el código fuente de ambas soluciones para evaluar la calidad del diseño, la modularidad y la eficiencia. Además, se realizaron pruebas de cobertura de código para garantizar que todas las partes esenciales del software se hayan ejecutado durante las pruebas.

Resultados de las Pruebas:

Los resultados de las pruebas de caja blanca indican una sólida estructura interna en ambas soluciones:

Sistema de Administración:

Se verificó la coherencia en la implementación de la automatización de procesos y la correcta gestión de datos.

Aplicación Móvil:

La estructura del código de la aplicación móvil se evaluó positivamente en términos de eficiencia y adherencia a las mejores prácticas de desarrollo.

Conclusiones:

Las pruebas de caja blanca confirman la solidez técnica del sistema de administración y la aplicación móvil desarrollados en el marco del proyecto de grado. No se identificaron vulnerabilidades críticas ni problemas estructurales significativos durante las pruebas.

Recomendaciones:

Se sugiere realizar auditorías de seguridad periódicas para mantener la integridad del software y abordar cualquier posible vulnerabilidad en el futuro.

4.8 SEGURIDAD

Informe de Seguridad – Aplicación panel de administración para Agencia de Turismo JIWAKY

Este informe destaca los aspectos de seguridad implementados en el sistema de administración y la aplicación móvil desarrollados para la agencia de turismo JIWAKY utilizando Yii Framework 2. Se abordan las medidas de seguridad clave implementadas para garantizar la integridad, confidencialidad y disponibilidad de los datos.

Marco de Desarrollo:

Yii Framework 2, conocido por su enfoque robusto en seguridad, ha sido utilizado para el desarrollo de este sistema. Su arquitectura segura y las funciones integradas para prevenir vulnerabilidades comunes contribuyen a fortalecer la seguridad del software.

Principales Puntos de Seguridad Implementados:

Autenticación y Autorización:

Se ha implementado un sistema de autenticación sólido para garantizar que solo usuarios autorizados tengan acceso al sistema.

La autorización se gestiona de manera granular, asignando roles específicos a los usuarios y restringiendo el acceso a funciones sensibles.

Protección contra Inyecciones SQL: Se han aplicado medidas para prevenir inyecciones SQL, incluyendo el uso de consultas preparadas y el uso seguro de las funciones de acceso a la base de datos.

Validación de Datos de Entrada: Se realiza una validación exhaustiva de los datos de entrada para prevenir ataques de inyección de código y garantizar que los datos ingresados cumplan con los formatos esperados.

Protección contra Cross-Site Scripting (XSS): Se han implementado técnicas para prevenir ataques XSS, como la sanitización de la entrada del usuario y el uso de funciones de escapado de salida.

Seguridad en Sesiones: Las sesiones de usuario se gestionan de manera segura, con la implementación de tokens anti-CSRF y medidas para evitar la fijación de sesiones.

Auditorías de Seguridad: Se incorporan auditorías de seguridad regulares para identificar posibles vulnerabilidades y abordarlas de manera proactiva.

Conclusiones: El sistema desarrollado con Yii Framework 2 demuestra un enfoque sólido en la seguridad, abordando múltiples frentes para proteger la aplicación contra amenazas comunes en entornos web.

Informe de Seguridad - Aplicación Móvil Desarrollada con Ionic para Agencia de Turismo JIWAKY

Este informe detalla las medidas de seguridad implementadas en la aplicación móvil desarrollada con Ionic para la agencia de turismo JIWAKY. Se destacan las prácticas de seguridad utilizadas para salvaguardar la integridad, confidencialidad y disponibilidad de la información.

Marco de Desarrollo:

La aplicación móvil se ha construido utilizando Ionic, un marco de desarrollo de aplicaciones móviles híbridas que aprovecha tecnologías web estándar. Aunque Ionic se basa en tecnologías web, se han aplicado medidas específicas para garantizar la seguridad de la aplicación.

Principales Puntos de Seguridad Implementados:

Comunicación Segura:

La comunicación entre la aplicación móvil y los servidores se realiza a través de protocolos seguros, como HTTPS, para garantizar la confidencialidad de los datos durante la transmisión.

Control de Acceso y Sesiones:

La aplicación implementa un sistema de control de acceso sólido, asegurando que solo usuarios autorizados tengan acceso a funciones específicas.

La gestión de sesiones se realiza de forma segura, con la implementación de tokens de sesión y medidas para prevenir ataques de fijación de sesiones.

Prevención de Ataques de Script entre Sitios (XSS):

Se aplican prácticas seguras de desarrollo para prevenir ataques XSS, incluyendo la validación y escapado adecuados de los datos de entrada y salida.

Almacenamiento Seguro:

Los datos almacenados localmente en el dispositivo se cifran para garantizar su seguridad en caso de acceso no autorizado al dispositivo.

Actualizaciones Seguras:

Se implementa un mecanismo seguro para la actualización de la aplicación, asegurando que solo versiones auténticas y seguras sean instaladas en los dispositivos de los usuarios.

Conclusiones:

La aplicación móvil desarrollada con Ionic muestra un enfoque integral en la seguridad, incorporando prácticas recomendadas para proteger la aplicación y los datos del usuario.

CAPITULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Identificación de Necesidades:

El análisis detallado reveló la urgencia de automatizar procesos en la agencia JIWAKY, desde la gestión de reservas hasta el apoyo al marketing, para superar desafíos y mejorar la eficiencia operativa.

Impacto Económico y Social:

La implementación del sistema busca no solo la eficiencia operativa, sino también beneficios económicos como ahorro de costos y aumento de ingresos. Además, destaca el impacto social, enriqueciendo la experiencia turística y fomentando el turismo sostenible.

Innovaciones Tecnológicas:

El proyecto incorpora innovaciones como accesibilidad universal, integración de servicios de mapas y mejoras en funcionalidades offline, adoptando un enfoque moderno y eficiente.

5.2 RECOMENDACIONES

Seguimiento Continuo:

Se recomienda un seguimiento constante de las tendencias tecnológicas y las necesidades del mercado para mantener el sistema actualizado y relevante.

Capacitación del Personal:

Es esencial proporcionar capacitación continua para asegurar que el personal adopte eficazmente el nuevo sistema y maximice sus beneficios.

Mantenimiento y Actualizaciones:

Un plan de mantenimiento regular y actualizaciones periódicas garantizará la seguridad, eficiencia y compatibilidad del sistema a lo largo del tiempo.

Monitoreo de Seguridad:

La seguridad debe ser una prioridad constante. Implementar un sistema de monitoreo para detectar y abordar posibles vulnerabilidades.

Evaluación de Impacto Social:

Realizar evaluaciones periódicas del impacto social del sistema, asegurando que las iniciativas de responsabilidad social estén alineadas con los objetivos del proyecto.

CAPITULO IV

BIBLIOGRAFÍA

- Sommerville, I. (2018). "Ingeniería de Software". Pearson Educación.
- ISO/IEC 9126. (2001). "Software Engineering - Product Quality". International Organization for Standardization.
- IEEE Std 829-2008. (2008). "IEEE Standard for Software and System Test Documentation". Institute of Electrical and Electronics Engineers.
- Pressman, R. S. (2014). "Ingeniería del Software: Un enfoque práctico" (7a ed.). McGraw-Hill Education.
- Boehm, B. W. (1981). "Software Engineering Economics". Prentice-Hall.
- Cockburn, A. (2001). "Agile Software Development." Addison-Wesley.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). "Design Patterns: Elements of Reusable Object-Oriented Software". Addison-Wesley.
- McConnell, S. (2004). "Code Complete: A Practical Handbook of Software Construction" (2nd ed.). Microsoft Press.
- Martin, R. C. (2003). "Agile Software Development: Principles, Patterns, and Practices". Pearson Education.
- Ambler, S. W. (2002). "Introduction to UML 2 Class Diagrams." Agile Modeling.
- Fowler, M. (2004). "UML Distilled: A Brief Guide to the Standard Object Modeling Language" (3rd ed.). Addison-Wesley.
- Gamma, E., Beck, K., Contribuyentes al Grupo de Patrones de Diseño. (1995). "Design Patterns: Abstraction and Reuse of Object-Oriented Design". In Proceedings of the 18th International Conference on Software Engineering (ICSE '96).

- W3Schools. <https://www.w3schools.com/>
Recurso educativo en línea para aprender tecnologías web como HTML, CSS, JavaScript, y más.
- Yiiframework. <https://www.yiiframework.com/>
Sitio oficial del framework Yii, que proporciona documentación detallada y recursos para desarrolladores.
- Ionic Framework. <https://ionicframework.com/>

Página principal del framework Ionic, con documentación, guías y recursos para el desarrollo de aplicaciones móviles.

- ISO - International Organization for Standardization. <https://www.iso.org/>
Sitio oficial de ISO, donde puedes encontrar información sobre estándares internacionales, incluidos los relacionados con la ingeniería de software.
- IEEE Xplore Digital Library. <https://ieeexplore.ieee.org/>
Biblioteca digital del IEEE con artículos, conferencias y estándares en ingeniería eléctrica, electrónica y disciplinas relacionadas.
- Agile Alliance. <https://www.agilealliance.org/>
Recursos y principios ágiles proporcionados por la Agile Alliance.
- Martin Fowler's Website. <https://martinfowler.com/>
Blog y artículos de Martin Fowler, una figura influyente en el mundo del desarrollo de software.
- GitHub. <https://github.com/>
Plataforma de desarrollo colaborativo que alberga muchos proyectos de código abierto y recursos útiles.
- Stack Overflow. <https://stackoverflow.com/>
Comunidad de programadores donde puedes encontrar preguntas y respuestas sobre una amplia variedad de temas relacionados con la programación.

- Medium. <https://medium.com/>
Plataforma de blogs donde los profesionales comparten artículos sobre diversos temas, incluyendo desarrollo de software y gestión de proyectos.
- TechCrunch. <https://techcrunch.com/>
Fuente de noticias tecnológicas que puede contener análisis y casos de estudio relevantes para proyectos de desarrollo.
- DZone. <https://dzone.com/>
Comunidad en línea que comparte noticias, tutoriales y artículos sobre desarrollo de software y tecnología.
- Towards Data Science. <https://towardsdatascience.com/>

Plataforma en Medium con enfoque en ciencia de datos y tecnologías emergentes.

- Hacker Noon. <https://hackernoon.com/>

Plataforma de publicación para programadores y profesionales de tecnología, donde se comparten experiencias y análisis.

- SitePoint. <https://www.sitepoint.com/>

Recursos educativos y artículos sobre desarrollo web, diseño y negocios en línea.

- A List Apart. <https://alistapart.com/>

Artículos sobre diseño web, desarrollo y contenido digital.

- Smashing Magazine. <https://www.smashingmagazine.com/>

Recursos para diseñadores y desarrolladores web, con artículos sobre tendencias y mejores prácticas.

- Tourism Review. <https://www.tourism-review.com/>

Portal de noticias y análisis en la industria del turismo, que podría ofrecer perspectivas sobre las tendencias y desafíos en el sector.

- Forrester. <https://go.forrester.com/>

Sitio de investigación de mercado que podría contener informes relevantes sobre tecnologías y estrategias en la industria.

YouTube:

Canal de Academind: <https://www.youtube.com/c/Academind>

The Net Ninja: <https://www.youtube.com/c/TheNetNinja>

Traversy Media: <https://www.youtube.com/user/TechGuyWeb>

Udemy:

<https://www.udemy.com/>

Plataforma con una amplia variedad de cursos pagados y gratuitos sobre desarrollo web, programación, diseño y gestión de proyectos.

Coursera:

<https://www.coursera.org/>

Ofrece cursos de universidades y organizaciones en línea sobre diversos temas, incluyendo tecnología y desarrollo.

LinkedIn Learning:

<https://www.linkedin.com/learning/>

Plataforma de aprendizaje con cursos sobre desarrollo web, diseño, gestión de proyectos y más.

Codecademy:

<https://www.codecademy.com/>

Enfocado en la enseñanza interactiva de programación y desarrollo web.

freeCodeCamp:

<https://www.freecodecamp.org/>

Comunidad de aprendizaje en línea con tutoriales y proyectos prácticos.

The Net Ninja (YouTube):

<https://www.youtube.com/c/TheNetNinja>

Ofrece tutoriales sobre desarrollo web y tecnologías relacionadas.

Academind (YouTube):

<https://www.youtube.com/c/Academind>

Contenido educativo sobre desarrollo web y tecnologías modernas.