



UNIVERSIDAD ANDINA DEL CUSCO

FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



TESIS:

“SOFTWARE PARA LA GESTION DE PAQUETES TURISTICOS DENOMINADO:
TURISMO SMART VERSION 2.5 IMPLANTADO EN EL OPERADOR TURISTICO
PERU JAMUY CUSCO-PERU.”

**TESIS PARA OPTAR EL TITULO PROFESIONAL
DE INGENIERO DE SISTEMAS**

PRESENTADO POR:

Br. Brain Mijail Garrido Cusipaucar

ASESOR:

Ing. Adriel Ramírez Vargas

CUSCO – PERÚ

2022



Dedicatoria

- A mi familia quienes por ellos soy lo que soy, pero sobre todo a mis padres por su amor, confianza, apoyo, comprensión y consejo además por ser mi ejemplo y mi motivación principal de mi constante superación.
- A mis compañeros y docentes porque sin duda fueron parte crucial para la culminación de este trabajo de investigación que será un aporte indispensable al turismo y a la empresa donde trabajo.

Bach. Brain Mijail Garrido Cusipaucar



Agradecimientos

A nuestra Universidad Andina del Cusco por haberme dado la oportunidad de brindarme las competencias y conocimientos en el mundo laboral y tenerla como casa de estudios y alma mater.

A nuestra querida escuela profesional de INGENIERÍA DE SISTEMAS que a lo largo de más de 29 años de actividad académica sigue formando profesionales.

A nuestros docentes por su abnegada dedicación y por transmitirnos sus conocimientos en el transcurso de nuestra vida universitaria con el único fin de hacernos útiles a la sociedad y el desarrollo del país.



Resumen

Actualmente los sistemas información son parte indispensable en el funcionamiento de las empresas y brindan apoyo neurálgico en el ámbito de sistematización de procesos. Dichos sistemas pueden ser utilizados a través de distintos dispositivos electrónicos mediante el internet de una manera muy dinámica y constante, la implantación de un Sistema de Información Web (SIW) que permitió gestionar los procesos informáticos del operador turístico “PERU JAMUY”.

Este trabajo de investigación de tipo aplicada se enfocó en el desarrollo e implementación de un Sistema de Información Web (SIW) en el ámbito turístico dentro del operador turístico “PERU JAMUY” ubicado en la ciudad del Cusco, teniendo como alcances facticos el desarrollo de un Sistema de Información Web (SIW) con una estructura definida y funcional la cual se logró con la conjunción de un equipo de desarrollo, software de desarrollo, y la articulación del hardware más el almacenamiento de la información en un servidor remoto con un dominio privado al alcance del usuario final que es la misma empresa.

El objetivo principal de este sistema información web es gestionar la articulación de los procesos de venta de paquetes turísticos que promocionan el operador turístico “PERU JAMUY”. La implementación de la investigación opto por el desarrollo de una metodología ágil.

Toda la gestión de los procesos informáticos son articulados por el Sistema de Información Web (SIW) el cual produce información valiosa para el operador turístico “PERU JAMUY”, esta información o resultado de procesos son almacenados en una base de datos ubicada en un servidor remoto con dominio privado en línea.

PALABRAS CLAVE: Sistema de Información Web (SIW), Turismo, Aplicación Web, Perú, Paquetes Turísticos, Pruebas, Gestionar, Bases de datos, PERU JAMUY.



Abstract

Currently, information systems are an indispensable part in the operation of companies and provide neuralgic support in the field of process systematization. Said systems can be used through different electronic devices through the Internet in a very dynamic and constant way. The implementation of a Web Information System allowed managing the IT processes in the tour operator "PERU JAMUY".

This applied research work focused on the development and implementation of a computer system in the tourism field within the tour operator "PERU JAMUY" located in the city of Cusco, having as factual scope the development of an information system web, with a defined and functional structure which was achieved with the conjunction of a development team, development software, and the articulation of hardware plus the storage of information on a remote server with a private domain available to the end user that it is the same company.

The main objective of this web information system is to manage the articulation of the tourist package sales processes promoted by the tour operator "PERU JAMUY". This implementation was guided under the development guidelines of an agile methodology.

All the management of the computerized processes are articulated by the web information system which produces valuable information for the tour operator "PERU JAMUY". This information or process results are stored in a database located on a remote server with domain private online.

KEY WORDS: Web information system, Tourism, Web Application, Perú, Tour Packages, Testing, Managing, Databases, PERU JAMUY.



Tabla de general

Introducción	1
Capítulo I: Problema de investigación	3
1.1. Ámbito de influencia	3
1.1.1. Ámbito de influencia teórica	3
1.1.2. Área de dominio	3
1.1.3. Línea de investigación.....	4
1.2. Planteamiento del problema	4
1.2.1. Descripción de la situación actual del lugar de intervención	4
1.2.2. Descripción del problema.....	6
1.2.3. Formulación del problema	7
1.2.4. Objetivos	8
1.2.5. Justificación.....	8
1.2.6. Alcance y limitaciones	9
Capitulo II: Marco teórico de la tesis	11
2.1. Antecedentes del desarrollo, implementación o transferencia de tecnología.....	11
2.1.1. Antecedentes nacionales	11
2.1.2. Antecedentes internacionales	14
2.2. Bases teóricas	16
2.2.1. Sistema de información	16
2.2.2. Aplicación web.....	18
2.2.3. Firma digital	19
2.2.4. Protocolo SET	20



2.2.5.	Internet	21
2.2.6.	Intranet	22
2.2.7.	Extranet – Servicios en la nube	23
2.2.8.	Navegar por la web	24
2.2.9.	Navegador web.....	24
2.2.10.	Agencia de viajes.....	25
2.2.11.	Visitante.....	28
2.2.12.	Programa turístico.....	29
2.2.13.	Cadena de secuencia de servicios	31
2.2.14.	Comercio electrónico.....	32
2.2.15.	Metodología de desarrollo	37
2.2.16.	Lenguaje de programación	46
2.2.17.	Tecnología	49
2.2.18.	Microsoft Sql Server.....	50
2.2.19.	Servidor	51
2.2.20.	Arquitectura modelo MVC	52
2.2.21.	Dominio	55
2.2.22.	Herramientas complementarias	56
Capitulo III: Desarrollo, implementación o transferencia tecnológica		58
3.1.	Desarrollo e implantación del Sistema de Información Web (SIW) “Turismo Smart Versión 2.5”	58
3.2.	Cronograma y recursos	59
3.2.1.	Administración del proyecto – Diagrama de Gantt.....	59
3.3.	Estudio de factibilidad	60



3.3.1.	Factibilidad técnica	60
3.3.2.	Factibilidad económica	61
3.3.3.	Factibilidad operativa.....	63
3.4.	Normas y roles del Sistema de Información Web (SIW)	63
3.4.1.	Normas internas.....	63
3.4.2.	Definición de roles del proceso de implementación del Sistema de Información Web (SIW)	64
3.5.	Fase análisis de requerimientos del Sistema de Información Web (SIW)	65
3.5.1.	Funciones de las áreas	66
3.6.	Metodología de desarrollo – SCRUM.....	68
3.6.1.	Intervención metodológica.....	68
3.7.	Fase de análisis de requerimiento	69
3.7.1.	Requerimientos del aplicativo web	69
3.7.2.	Product backlog.....	70
3.8.	Release backlog (lista de historia de usuario por orden de importancia)	80
3.9.	Definición de los sprint.....	82
3.10.	Planificación de los sprint.....	86
3.11.	Sprint BurnDown chart	89
3.12.	Desarrollo del sistema.....	91
	Semana 01, 02 a la fecha 13/06/2019.....	93
	Semana 03, 04 a la fecha 27/06/2019.....	99
	Semana 05 a la fecha 13/07/2019.....	104
	Semana 06, 07 a la fecha 26/07/2019.....	108
	Semana 08, 09 a la fecha 24/08/2019.....	113



Semana 10,11 a la fecha 20/09/2019.....	119
Semana 12, 13 a la fecha 04/10/2019.....	124
Semana 14,15 a la fecha 25/10/2019.....	129
Semana 16 a la fecha 27/10/2019.....	134
3.13. Arquitectura e implementación.....	138
3.13.1. Capa Presentación.....	138
3.13.2. Capa Proceso (Controlador)	139
3.13.3. Capa Modelo.....	139
Capitulo IV: Resultados	140
4.1. Comprobación de la prospectiva	140
4.2. Cumplimiento de los objetivos	140
4.3. Contribuciones (impacto)	141
Glosario	143
Conclusiones	148
Recomendaciones.....	149
Referencias	150
Anexos.....	157



Índice de tablas

Tabla 1: Planteamiento del problema.....	6
Tabla 2: Primer antecedente nacional	11
Tabla 3: Segundo antecedente nacional	12
Tabla 4: Primer antecedente internacional	14
Tabla 5: Segundo antecedente internacional	15
Tabla 6: Recurso humano para la implantación	62
Tabla 7: Recursos tecnológicos para la implantación	62
Tabla 8: Recurso humano para el desarrollo	63
Tabla 9: Recurso humano para el desarrollo	64
Tabla 10: Recurso humano actual del negocio.....	66
Tabla 11: Recurso humano para el desarrollo	69
Tabla 12: Historia de usuario HUU01.....	71
Tabla 13: Historia de usuario HUU02.....	71
Tabla 14: Historia de usuario HUU03.....	72
Tabla 15: Historia de usuario HUU04.....	72
Tabla 16: Historia de usuario HUU05.....	73
Tabla 17: Historia de usuario HUU06.....	73
Tabla 18: Historia de usuario HUU07.....	75
Tabla 19: Historia de usuario HUU08.....	75
Tabla 20: Historia de usuario HUU09.....	76
Tabla 21: Historia de usuario HU10	76
Tabla 22: Historia de usuario HU11	77
Tabla 23: Historia de usuario HU12	78



Tabla 24: Historia de usuario HU13	78
Tabla 25: Historia de usuario HU14	79
Tabla 26: Historia de usuario HU15	79
Tabla 27: Product backlog.....	80
Tabla 28: Release backlog.....	80
Tabla 29: Definición de los sprint.....	82
Tabla 30: Estimación del tiempo para el sprint 01	83
Tabla 31: Estimación del tiempo para el sprint 02.....	84
Tabla 32: Estimación del tiempo para el sprint 03.....	85
Tabla 33: Estimación del tiempo para el sprint 04.....	85
Tabla 34: Planificación del sprint 01.....	86
Tabla 35: Planificación del sprint 02.....	87
Tabla 36: Planificación del sprint 03.....	88
Tabla 37: Planificación del sprint 04.....	88
Tabla 38: Taskboard inicial de desarrollo	89
Tabla 39: Taskboard semana 01, 02.....	91
Tabla 40: Taskboard semana 01, 02.....	95
Tabla 41: Tabla funcional semana 01, 02	97
Tabla 42: Taskboard semana 03, 04.....	100
Tabla 43: Tabla funcional semana 03, 04	103
Tabla 44: Taskboard semana 05.....	105
Tabla 45: Tabla funcional semana 05	107
Tabla 46: Taskboard semana 06, 07.....	109
Tabla 47: Tabla funcional semana 06,07	111



Tabla 48: Taskboard semana 08, 09	114
Tabla 49: Tabla funcional semana 08, 09	117
Tabla 50: Taskboard semana 10, 11	120
Tabla 51: Taskboard semana 12, 13	125
Tabla 52: Tabla funcional semana 12, 13	127
Tabla 53: Taskboard semana 14, 15	130
Tabla 54: Tabla funcional semana 14, 15	132
Tabla 55: Taskboard semana 14, 15	135
Tabla 56: Tabla funcional semana 16	137



Índice de figuras

Figura 1: Situación actual del negocio	5
Figura 2: Implementación del Sistema de Información Web (SIW).....	10
Figura 3: Funciones de un sistema de información.....	17
Figura 4: Arquitectura de aplicación web centrada en el usuario por capas.....	18
Figura 5: Mecánica de una firma digital y de la comprobación de la firma.....	19
Figura 6: Funcionamiento de Protocolo SET.....	20
Figura 7: Visitante y su Clasificación	29
Figura 8: El Programa Turístico.....	29
Figura 9: Tipos de Comercio Electrónico	35
Figura 10: Marco de trabajo SCRUM.....	37
Figura 11: Logo de PHP.....	46
Figura 12: Logo Laravel.....	47
Figura 13: Tecnologías agrupadas bajo el concepto de AJAX	49
Figura 14: Componentes de SQL SERVER.....	50
Figura 15: Modelo vista controlador + 3 capas.....	52
Figura 16: Flujo de proceso MVC.....	55
Figura 17: Arquitectura de Desarrollo del Software	56
Figura 18: Cronograma de Actividades	59
Figura 19: Gráfico de Trabajo (BurnDown Chart). Semana N° 0	91
Figura 20: Gráfico de Trabajo (BurnDown Chart). Semana N° 01, 02	93
Figura 21: Diseño de Interfaz del Back-end e integración de plantilla.....	94
Figura 22: Integración de plantilla framework Page.....	94
Figura 23: Gráfico de Trabajo (BurnDown Chart). Semana N° 01, 02	97



Figura 24: Creación de la Base de datos (Proceso de accesos)	99
Figura 25: Creación de la Base de datos (Proceso de cotización).....	99
Figura 26: Creación de la Base de datos (Proceso de asignación y pago)	100
Figura 27:Gráfico de Trabajo (BurnDown Chart). Semana N° 03, 04	102
Figura 28: Acceso al sistema (Login)	104
Figura 29: Credenciales de Acceso al sistema (Login)	104
Figura 30: Gráfico de Trabajo (BurnDown Chart). Semana N° 05	107
Figura 31:Administración de operadores (clientes directos).....	108
Figura 32: Gráfico de Trabajo (BurnDown Chart). Semana N° 06, 07	111
Figura 33: Administración de usuarios	113
Figura 34: Administración de Roles.....	113
Figura 35: Administración de Paquetes turísticos.....	114
Figura 36: Gráfico de Trabajo (BurnDown Chart). Semana N° 08,09	116
Figura 37: Administración de Personal Staff	119
Figura 38: Cotizar servicios	119
Figura 39: Gráfico de Trabajo (BurnDown Chart). Semana N° 10, 11	122
Figura 40: Tabla funcional semana 10, 11	122
Figura 41: Reportes y control.....	124
Figura 42: Front-end.....	124
Figura 43: Gráfico de Trabajo (BurnDown Chart). Semana N° 12,13	127
Figura 44: Integración de la plantilla	129
Figura 45: Categorización de cuentas actualizar BD	129
Figura 46: Gráfico de Trabajo (BurnDown Chart). Semana N° 14,15	132
Figura 47: Ingreso al registro del cliente.....	134



Figura 48: Registrar al cliente	134
Figura 49: Gráfico de Trabajo (BurnDown Chart). Semana N° 16	137



Introducción

En la actualidad existen múltiples herramientas online que buscan apoyar el trabajo en el sector de los operadores turísticos de la ciudad del Cusco, sin embargo, estas herramientas muchas veces no cubren los requerimientos específicos del modelo de negocio.

Para iniciar la gestación de la idea sobre la cual se planteó y desarrolló el software como una herramienta de solución informática nace de los aspectos básicos de los procesos administrativos y logísticos del operador turístico “PERU JAMUY”, así como comprender las herramientas actuales que se usan. Posteriormente, una vez terminado el desarrollo de la investigación y recopilación de información, se definieron una serie de actividades en las que se contemplaban el diseño, desarrollo e implementación de la solución que resolvió el problema identificado, en la que se usó el conocimiento adquirido a lo largo de ciclos estudiados en la escuela profesional de Ingeniería de Sistemas de la UAC, tales como la captura de requerimientos, el diseño de un software y el desarrollo de pruebas sobre la solución implementada. Finalmente, para completar el desarrollo de la aplicación se integró el conocimiento investigado sobre el mundo del operador turístico en la región y desarrollo de contenidos con el desarrollo de software en un solo modelo funcional con el cual se realizó las pruebas sobre el usuario final.

Dicho sistema de información surge de las necesidades que el operador turístico “PERU JAMUY”, debido a diversos problemas que enfrenta especialmente en las áreas:

- Ventas; en cuanto al inicio, proceso y finalización de una cotización de servicios.
- Operaciones; en cuanto a la programación y compra de boletos, alojamiento, vuelos, trenes y otros, como también en la asignación de proveedores a cada servicio confirmado.
- Administración; en cuanto al manejo de caja y a los pagos de proveedores.

El sistema que se ha implantado contribuye a mejorar el registro de cada actividad que involucra una cotización generada, mejorando los tiempos y procesos en cada actividad realizada; dando soporte de una base de datos.



La información necesaria para la implantación del Sistema de Información Web (SIW), ha sido recopilada a través de documentos, archivos y entrevistas facilitados por las áreas intervenidas de la operadora turística “PERU JAMUY”.

El trabajo de investigación aplicada ha sido organizado a través de capítulos de acuerdo a la resolución de la FIA Res-100-2018-CFIA-UAC-esquema-tesis-FIA, logrando concretar detalladamente los detalles solicitado en cada capítulo, además de los anexos requeridos en dicha resolución.

En el Capítulo I, se describe los principales problemas que se observaron en la empresa OPERADOR TURISTICO PERU JAMUY, fueron relacionados a la articulación de procesos funcionales organizativos e informáticos visualizados en la organización.

En el Capítulo II, se recopiló la información que sirvió de marco teórico y las bases teóricas de la tesis donde se citó y referenció a diferentes autores que contribuyeron a formar la base científica para la creación de esta investigación aplicada.

En el Capítulo III, se logró el desarrollo del Sistema de Información Web (SIW), que da solución a los requerimientos establecidos en la empresa OPERADOR TURISTICO PERU JAMUY, concretando la transferencia tecnológica.

En el Capítulo IV, se realizó el proceso de retroalimentación o verificación de resultados, tendiendo como meta alcanzada la implantación del Sistema de Información Web (SIW), propuesta en el objetivo de la tesis de investigación aplicada.



Capítulo I: Problema de investigación

1.1. **Ámbito de influencia**

El ámbito de influencia de la investigación está enmarcado dentro del área de tecnologías de información y comunicación y la línea de investigación se definió como tecnologías de información.

1.1.1. **Ámbito de influencia teórica**

La presente investigación de desarrollo aplicativo surgió por la necesidad de desarrollar e implementar un Sistema de Información Web (SIW) el cual administre los procesos informáticos del OPERADOR TURISTICO PERU JAMUY en la provincia del Cusco, Departamento del Cusco, y los servidores privados donde se encuentran en el dominio y el nombre hosting, donde reside el sistema informático al cual se le denominó Turismo Smart Versión 2.5. Las áreas de dominio de que se consideró en la tesis pertenecen a las tecnologías de información y de comunicación especificadas por la Escuela profesional de Ingeniería de Sistemas.

1.1.2. **Área de dominio**

En el área de dominio de las tecnológicas de la información y comunicación se realizó un proceso de identificación y selección para el desarrollo e implementación del Sistema de Información Web (SIW) Turismo Smart Versión 2.5. Es así que se determinó el uso de lenguajes de programación, y Entornos de desarrollo (IDE) tales como:

- Herramientas de diseño BPMN, con el modelo MVC.
- Lenguaje de programación para el desarrollo de software (PHP)
- Herramientas de gestión de base de datos Microsoft SQL Server Management Studio.
- Implementación de hardware (Sensores, dispositivos electrónicos)
- Framework (Laravel)



- Hosting y dominios, o en su defecto localhost de ubicación virtual del software.

Para lo cual se consultó diferentes Notas bibliográficas y artículos científicos que nos muestren la conceptualización y uso de las herramientas antes mencionadas. Esta investigación de los conceptos teóricos y la documentación de los sistemas y sus teorías, permitieron sustentar el ámbito de influencia teórico de nuestro proyecto de desarrollo.

Es por ello que el área de dominio está considerada dentro de las tecnologías de la información.

Además, la configuración de los protocolos de comunicación de Cloud store que guarda el repositorio de datos.

1.1.3. Línea de investigación

En el diseño, desarrollo e implementación del software o Sistema de Información Web (SIW) Turismo Smart Versión 2.5 se utilizaron herramientas de diseño, de programación, sistemas gestores de datos. Los cuales han contribuido a la creación y formación de un sistema información Web, el cual está enmarcado en la línea de investigación se definió como tecnologías de información.

1.2. Planteamiento del problema

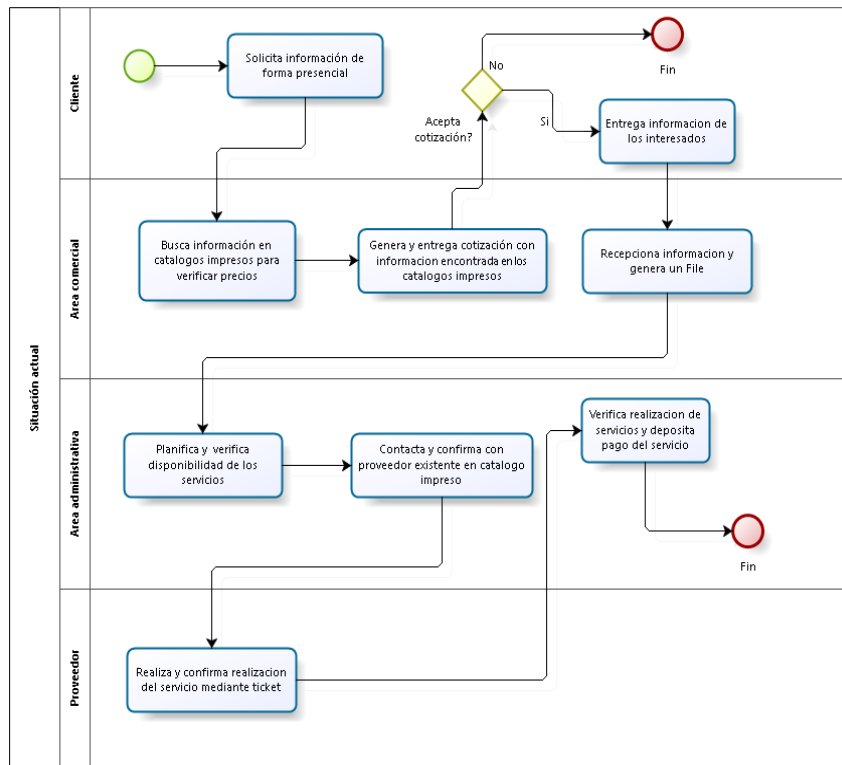
Los principales problemas que se observaron en la empresa operador turístico PERU JAMUY, fueron relacionados a la articulación de procesos funcionales organizativos e informáticos.

1.2.1. Descripción de la situación actual del lugar de intervención

La MYPE PERU JAMUY, es una empresa cusqueña, la cual está dedicada al rubro turístico. Actualmente, el operador turístico cuenta con su oficina principal en la Av. El Sol 930, en la cual cuenta con tres computadoras que gestionan el funcionamiento administrativo y operaciones turísticas, las cuales son utilizadas por el personal de la empresa. Todas estas máquinas de escritorio cuentan con un internet fluido y estable.

La generación de paquetes turísticos. Está en función del proceso que comienza y termina al generar una cotización por medio de un sistema manual establecido dentro del operador turístico PERU JAMUY todo es realizado por el personal de la oficina.

Figura 1: Situación actual del negocio



Fuente: Creación propia

1.2.1.1. Organización

Las diferentes áreas del operador turístico PERU JAMUY no tienen definido sus procesos administrativos y organizacionales. Estas opiniones a usualmente entran en conflicto en cuanto a la forma en la que debe distribuir los recursos.

1.2.1.2. Administración

El trabajo de la gerencia es dar sentido a las distintas situaciones a las que se enfrentan las organizaciones, tomar decisiones y formular planes para solucionar los problemas organizacionales. En este caso la gerencia toma la decisión de adoptar un sistema de información, el cual ayudara a la organización a definir sus procesos administrativos y ofrecer nuevos productos y servicios.



1.2.2. Descripción del problema

Tabla 1: Planteamiento del problema

Problemas	Causas	Consecuencias
Falta de datos de las personales, empresas o clientes actualizados y digitalizados que visitan a operador turístico PERU JAMUY.	No existía un proceso de recaudación información de las entidades que interactúan con el operador turístico PERÚ JAMUY.	Información no digitalizada y organizada de las entidades que interactúan con el operador turístico PERÚ JAMUY.
Desorganización de los procesos administrativos e inadecuada utilización de recursos de la empresa.	Falta de definición de procesos administrativos organizacionales.	Procesos de gestión interna de la empresa turística no limitados
Generación de paquetes turísticos no organizados ni programáticos de forma manual no digitalizada, los cuales no satisfacían la necesidad de los clientes.	No existía un mecanismo o herramienta que permitiera generar paquetes turísticos óptimos para el cliente de la empresa.	Venta de paquetes turísticos de forma no organizada y no programada, produciendo malestar e incomodidad en los clientes.

Fuente: Descripción concisa del problema que se va a mejorar, Autor: Creación propia

Debido a los problemas planteados en el punto anterior, la solución que se propuso fue el desarrollar un Sistema de Información Web (SIW) que permita fortalecer la funcionalidad organizativa para la generación de paquetes turísticos en el operador turístico PERU JAMUY.

El Sistema de Información Web (SIW) ha sido desarrollado basándose en los procesos internos de la empresa y los modelos de sistemas informáticos vigentes en el mercado, generando un Sistema de Información Web (SIW) a la medida de la empresa: La



obtención de datos de los clientes y los servicios brindados por la empresa son muy importantes. Ambos procesos son fundamentales en la definición del perfil del producto o paquete turístico que solicitan a la empresa, ya que si no se cuenta con su información no se pueden desarrollar los procesos que continúan, gestión de la información, entre otros, los cuales se desarrollarán en futuros proyectos.

El desarrollo del proyecto se gestionó con la metodología de desarrollo ágil SCRUM, se aplicó las buenas prácticas que esta metodología plantea y se utilizó los artefactos necesarios para realizar una correcta gestión.

Para la implementación del Sistema de Información Web (SIW) del trabajo de investigación aplicada se utilizó la herramienta de desarrollo Laravel 5.8 (PHP) y el motor de base de datos SQL Server Express. Todo esto en el sistema operativo Windows Server 2018. Para un correcto funcionamiento, se recomienda un procesador Intel Core Duo o más y memoria RAM de 3GB o más, contar con una infraestructura de Internet constante. Además de las consideraciones ya mencionadas en el punto de Área de dominio.

- Falta de datos de las personales, empresas o clientes actualizados y digitalizados que visitan al operador turístico PERU JAMUY.
- Desorganización de los procesos administrativos e inadecuada utilización de recursos de la empresa.
- Generación de paquetes turísticos no organizados ni programáticos de forma manual no digitalizada, los cuales no satisfacían las necesidades de los clientes.

1.2.3. Formulación del problema

¿Cómo gestionar un paquete turístico con Sistema de Información Web (SIW), de acuerdo a la información de los clientes, utilizando los procesos funcionales y los servicios internos que ofrece el operador turístico PERU JAMUY?



1.2.4. Objetivos

General

Desarrollar un Sistema de Información Web (SIW) que permita gestionar paquetes turísticos, integrando los procesos funcionales del operador turístico “PERU JAMUY”

Específicos

1. Analizar los procesos funcionales que permitan gestionar paquetes turísticos.
2. Determinar los requerimientos para la implementación del Sistema de Información Web (SIW)
3. Diseñar la estructura funcional del Sistema de Información Web (SIW) (Product Backlog)
4. Documentar e implementar el Sistema de Información Web (SIW)
5. Testear los resultados producidos por el Sistema de Información Web (SIW)
6. Desplegar la solución en las instancias definidas por la organización para la empresa y el cliente.

1.2.5. Justificación

Actualmente las empresas dedicadas al rubro del turismo brindan servicios y productos turísticos a diversos clientes en el ámbito nacional e internacional, y necesitan procesos de administración ágiles y organizados apoyados en la TIC para mejorar la realización de sus prestaciones y satisfagan las necesidades del turista.

Al igual que otras empresas en el rubro de turismo, el operador turístico PERU JAMUY tubo la necesidad de mejorar la administración de los procesos que realizan en sus servicios turísticos y actividades, ya que en el año 2018 se controlaban de forma manual; con frecuencia se generan inconvenientes al momento de solicitar información referente a los programas de servicios, es por lo que se propuso el “Desarrollo del Sistema de Información Web (SIW) denominado Turismo Smart Versión 2.5, permite la administración de los servicios y actividades, y venta de servicios turísticos”, que se acomoda a los requerimientos señalados por los usuarios que utilizaran el sistema;



proporcionando así una herramienta tecnológica capaz de agilizar los procesos ayudando a reducir los tiempos de respuestas, toma de decisiones y en la maximización de los recursos disponibles materiales como humano.

Analizar los procesos funcionales de la empresa permitió diseñar, desarrollar e implementar una infraestructura tecnológica, la cual se sostiene en la metodológica de desarrollo ágil SCRUM, que permitió gestionar la evolución e implementación del Sistema de Información Web (SIW) denominado Turismo Smart Versión 2.5.

1.2.6. Alcance y limitaciones

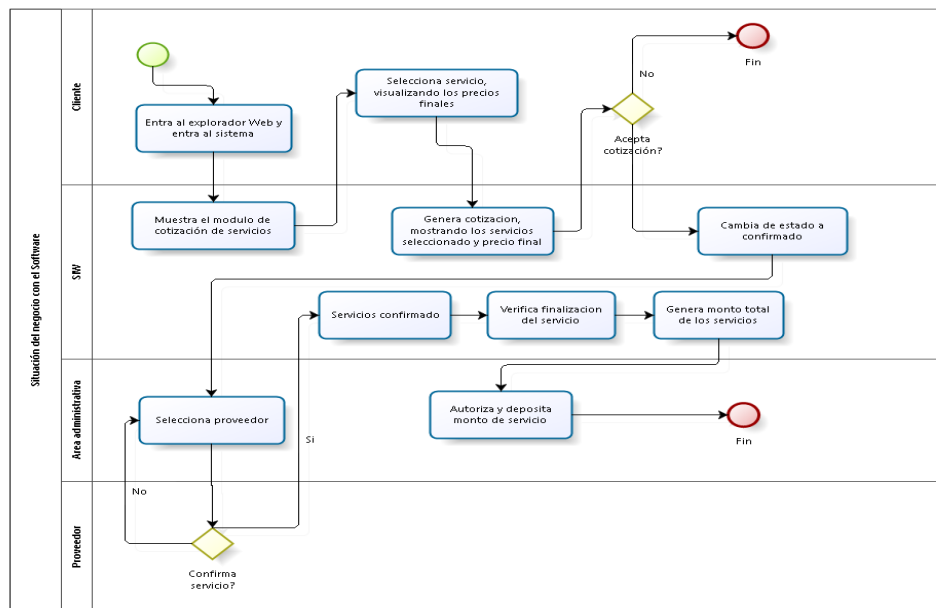
Alcances

Actualmente las empresas con fines de lucro que se dedican a brindar servicios turísticos en la región del Cusco están en constante crecimiento y se ven en la necesidad de evolucionar en el tipo administración para la información que requieren llevar a cabo en cada una de sus actividades o procesos dentro de su modelo de negocio, para lo cual adquieren tecnología que les permita el control y manejo de la información.

Al igual que otras empresas turísticas de la región del Cusco el operador turístico “PERU JAMUY tubo la necesidad de mejorar la administración de los procesos dentro de su modelo de negocio dentro de los servicios turísticos que presta y las actividades de control que realizaba de forma manual, con frecuencia se generan inconvenientes al momento de solicitar información referente a los programas de servicios, es por lo que se propone el “Sistema de Información Web (SIW) denominado Turismo Smart Versión 2.5”, será utilizado por todo el personal para la administración de los servicios y actividades, que se adecue a los requerimientos señalados por los usuarios que utilizaran el sistema; proporcionando así una herramienta tecnológica capaz de agilizar los procesos ayudando a reducir los tiempos de respuestas, toma de decisiones y en la maximización de los recursos disponibles materiales como humano.



Figura 2: Implementación del Sistema de Información Web (SIW)



Fuente: Creación propia

Limitaciones

En el desarrollo del trabajo de investigación aplicada se encontraron las siguientes limitaciones:

- Limitación de recursos tecnológicos de la empresa para el desarrollo del Sistema de Información Web (SIW), en cuando a los equipos de hardware y el software utilizado, tales como lenguajes de programación y gestores de bases de datos, lo que limitó alcanzar el cronograma establecido en el proceso de diseño, desarrollo e implantación del Sistema de Información Web (SIW) denominado Turismo Smart Versión 2.5
- Adjuntar archivos e información referente al proceso interno de manejo de información del operador turístico “PERU JAMUY”
- Realización de pruebas del prototipo del software denominado Turismo Smart Versión 2.5 en tiempo real en las oficinas del operador turístico “PERU JAMUY”.
- El brote mundial de COVID-19 ha llevado al mundo a su paralización, y el turismo ha sido el más afectado de todos los grandes sectores económicos del



país. Lo que implicó el retraso en el despliegue del Sistema de Información Web (SIW) creado.

Capítulo II: Marco teórico de la tesis

2.1. Antecedentes del desarrollo, implementación o transferencia de tecnología

2.1.1. Antecedentes nacionales

Sistema de información para la gestión de reserva de paquetes turísticos en la agencia de viaje rap travel

Tabla 2: Primer antecedente nacional

Título	Sistema de información para la gestión de reserva de paquetesturísticos en la agencia de viaje rap travel
Autor	Bach. Estefany Huaman Quispe
Año	2017
Lugar	Cusco – Perú
Universidad	Universidad Andina del Cusco
Descripción	<p>Para desarrollar apropiadamente el sistema de información se utilizó la metodología del proceso de desarrollo unificado del software, debido a que esta metodología contempla desde la identificación de los problemas, análisis, diseño del proyecto para ser implementado, las pruebas respectivastomando en consideración los requerimientos de la agencia de viaje, asimismo se utilizó las herramientas necesarias como las entrevistas y encuestas a los trabajadores.</p> <p>Se identificó los problemas en el proceso de la gestión de reservas de los paquetes turísticos que lo realizaban manualmente además</p>



que no contaban con un control adecuado para el seguimiento del pago a sus proveedores, por ello se desarrolló el sistema de información, el cual permita a los trabajadores desarrollar sus actividades de manera óptima. (Quispe, 2017)

Aporte Personal La implementación del sistema de información de gestión de reservas permite visualizar el estado en que se encuentran las cotizaciones, asimismo muestra el estado del file y las reservas que se realizan por cada file confirmado, además de contar con los servicios y proveedores de cada una de estos servicios.

Nota 2: Descripción del primer antecedente nacional Autor: Creación propia

Tecnologías de información aplicables al sector turismo en el Perú

Tabla 3: Segundo antecedente nacional

Título	Tecnologías de información aplicables al sector turismo en de Perú
Autor	Javier Ubaldo Gonzales Estrada Luz Maribel Julca Rodríguez
Año	2005
Lugar	Lima – Perú
Universidad	Universidad nacional mayor de San Marcos
Descripción	Proveedores, agentes turísticos y consumidores han sido impactados por las nuevas tecnologías. Sistemas de información y herramientas electrónicas que permiten minimizar las barreras de tiempo y distancia, facilitan el acceso a Notas de información y potencian el desarrollo de nuevos productos y servicios, además, inciden favorablemente en la eficiencia operativa de las organizaciones, la calidad de los servicios prestados, el



acercamiento con los clientes y la coordinación de actividades entre los diversos actores (públicos y privados) que interviene en el turismo.

En el transcurso de los últimos años, los avances en las tecnologías de la información y las comunicaciones, en especial de Internet, permiten tener acceso en línea, las 24 horas del día, los 365 días del año, a: información actualizada de todos los países del mundo, servicios meteorológicos, herramientas para calcular el cambio de divisas, horarios internacionales, facilidades de búsqueda, conversión de medidas, y en general, a toda la información referente al lugar de destino, con objeto de que el trayecto resulte familiar y seguro (Gonzales Estrada, 2005)

Aporte Personal La industria turística mundial ha encontrado en las tecnologías de Información y comunicación (TIC) un componente fundamental para su desarrollo y expansión. Las empresas turísticas en el Perú no son ajenas a ello, y ya se pueden ver casos concretos en que las iniciativas tecnológicas se han convertido en un valor agregado para estas empresas. Con el presente trabajo de investigación y desarrollo se espera conocer la real dimensión en cuanto al uso de las TIC por parte de las empresas turísticas peruanas, y de ese modo identificar sus necesidades concretas para, finalmente, poder plantear una estrategia de negocio, materializada en una solución web, cuya aplicación brinde una ventaja competitiva a las empresas del sector y que a través de su difusión muchas más empresas puedan gozar de los beneficios que actualmente otorgan las TIC.

Nota 3: Descripción del segundo antecedente nacional Autor: Creación propia



2.1.2. Antecedentes internacionales

Sistema de información vía web para la promoción y administración de servicios caso: EMISTUR

Tabla 4: Primer antecedente internacional

Título	Sistema de información vía web para la promoción y administración de servicios caso: EMISTUR
Autor	Ingrid Nathaly Quisbert Saavedra
Año	2009
Lugar	La Paz – Bolivia
Universidad	Universidad mayor de San Andrés
Descripción	Estamos viviendo en una sociedad de información y economía globalizada, las organizaciones el cual dependen cada vez más de la administración y la distribución de la información a través de redes de comunicación. Las organizaciones buscan a través de nuevas estrategias y herramientas que les permiten el logro y cumplimiento de sus objetivos y la adquisición de ventajas competitivas. Eso con lleva el uso de Nuevas Tecnologías de Información, por medio de las cuales las organizaciones consiguen grandes beneficios, como el mejoramiento de sus operaciones, una mejor entrega de servicios, optimización de recursos logrando aumentar su eficiencia. (Saavedra, 2009)
Aporte Personal	Con el presente trabajo se pretende contribuir con el desarrollo de la empresa, mejorando la asistencia que presta la empresa en cuanto a la emisión de información tanto turística como de los servicios que ofrece, de igual manera mejorar el manejo de los datos que son de total importancia para la institución.

Nota 4: Descripción del primer antecedente internacional Autor: Creación propia



Elaboración y creación de una página web para fomentar el turismo en el sector del tingo y dar a conocer su cultura, paisajes naturales, eventos culturales, platos típicos y tradicionales del sector ubicado al sureste del distrito metropolitano de quito en la parroquia de Alangasí, así logrando el desarrollo turístico del tingo.

Tabla 5: Segundo antecedente internacional

Título	Elaboración y creación de una página web para fomentar el turismo en el sector del tingo y dar a conocer su cultura, paisajes naturales, eventos culturales, platos típicos y tradicionales del sector ubicado al sureste del distrito metropolitano de quito en la parroquia de Alangasí, así logrando el desarrollo turístico del tingo.
Autor	Edwin Jonathan Cuvi Pagalo
Año	2014
Lugar	Quito – Ecuador
Instituto	Instituto tecnológico Cordillera
Descripción	<p>El presente trabajo de investigación aplicada tiene como finalidad principal la creación de un sitio web interactivo para fomentar el turismo en el sector del Tingo ubicado al suroeste del DMQ en la parroquia de Alangasí. Se realizó un sitio web interactivo el cual ayudará al sector darse a conocer como destino turístico en donde se podrá encontrar información del sector sus servicios, mediante galerías interactivas del sector.</p> <p>En este sitio web podrán encontrar todo lo referente al sector festividades, platos típicos para de esta forma que el sector sea un lugar y un destino turístico muy reconocido en el Distrito Metropolitano de Quito y de esta manera el, de lograr una</p>



permanente mejoras en el servicio para poder brindar un servicio de calidad a la altura de grandes lugares turísticos para que el turista se sienta a gusto de visitar el sector y lo recomiendo como un destino turístico a los demás turistas, y como meta final logra una completa satisfacción en todos los turistas que visiten el sector.

Aporte Personal Con el presente trabajo se pretende contribuir con el desarrollo de la empresa, mejorando la asistencia que presta la empresa en cuanto a la emisión de información tanto turística como de los servicios que ofrece, de igual manera mejorar el manejo de los datos que son de total importancia para la institución.

Nota 5: Descripción del segundo antecedente internacional Autor: Creación propia

2.2. Bases teóricas

2.2.1. Sistema de información

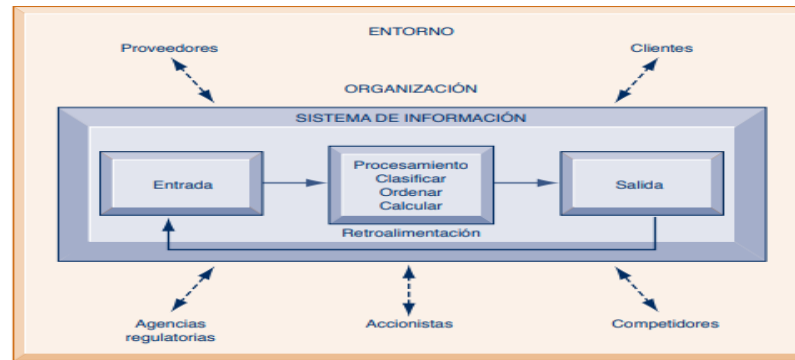
Definición de sistema

Un sistema real, en cambio es una entidad material formada por componentes organizados que interactúan de forma en que las propiedades del conjunto no pueden deducirse por completo de las propiedades de las partes (denominadas propiedades emergentes) (Fernández Alarcón, 2006).

Los sistemas reales comprenden intercambios de energía, información o materia con su entorno. Las células y la biosfera son ejemplos de sistemas naturales. Existen tres tipos de sistemas reales: abiertos (recibe flujos de su ambiente, adaptando su comportamiento de acuerdo a esto), cerrados (sólo intercambia 13 energía con su entorno) y aislados (no realiza ningún tipo de intercambio con su entorno) (Gómez Vieites, 2010).



Figura 3: Funciones de un sistema de información



Nota 6: Tres actividades básicas (entrada, procesamiento y salida) producen la información que necesitan las empresas. Nota: (Laudon, 2018)

Conjunto de componentes interrelacionados que obtienen, procesan, almacenan y distribuyen información para apoyar la toma de decisiones, la coordinación y el control en una organización (Laudon, 2018).

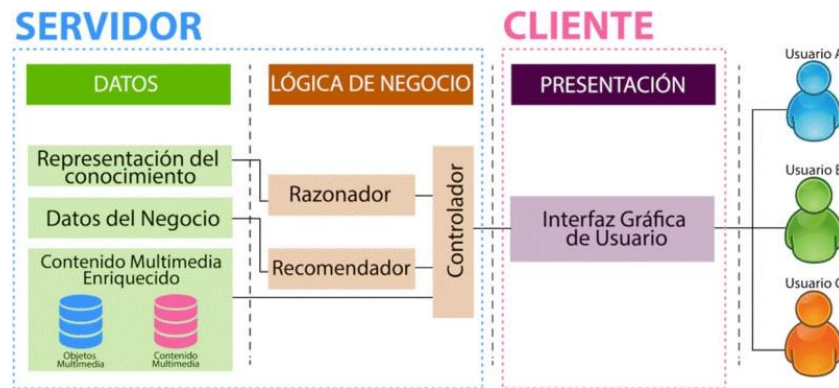
Puede ser cualquier combinación organizada de personas, hardware, software, redes de comunicación y recursos de información que almacene, recupere, transforme y disemine información en una organización. Las personas han confiado en los sistemas de información para comunicarse entre sí mediante una variedad de dispositivos físicos (hardware), instrucciones y procedimientos de procesamiento de información (software), canales de comunicación (redes) y datos almacenados (recursos de información) desde los albores de la civilización (Marakas, 2019).

Un sistema de información se puede definir técnicamente como un conjunto de componentes relacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control en una organización (Heredero, 2018).

2.2.2. Aplicación web

Definición de aplicación web

Figura 4: Arquitectura de aplicación web centrada en el usuario por capas.



Nota 7: (Laudon, 2018; Hernández, Mancilla, Narvaez, & Iregui, 2013)

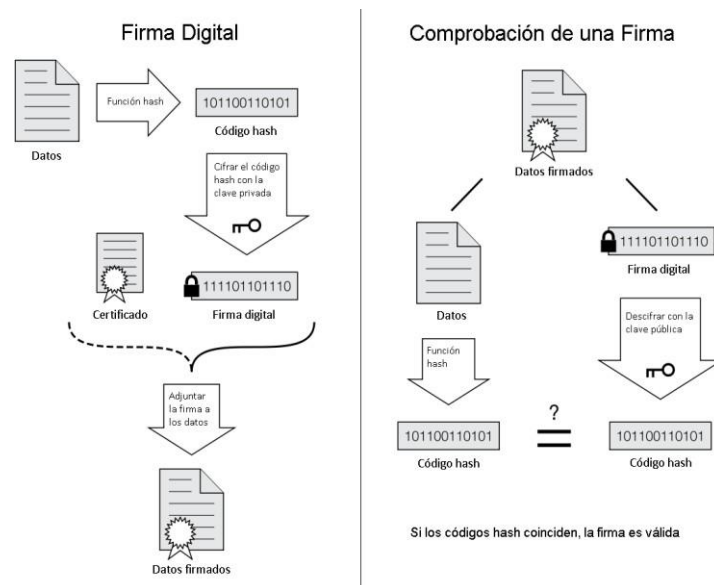
De acuerdo a (Mora, 2020), en las aplicaciones web suelen distinguirse tres niveles (como en las arquitecturas cliente/servidor de tres niveles):

- El nivel superior que interacciona con el usuario (el cliente web, normalmente un navegador).
- El nivel inferior que proporciona los datos (la base de datos)
- El nivel intermedio que procesa los datos (el servidor web).

En este capítulo se describen el cliente y el servidor web y se comentan los entornos web en los que se ejecutan las aplicaciones web: Internet, intranet y extranet. Además, se comentan las principales ventajas que poseen las aplicaciones web. También se describen las arquitecturas típicas de las aplicaciones web. Por último, se presenta una metodología de desarrollo de sitios web.

2.2.3. Firma digital

Figura 5: Mecánica de una firma digital y de la comprobación de la firma.



Nota 8: (Acdx, 2019)

De acuerdo a (Figuerola, Zazo Rodríguez, & Alonso Berrocal), a mediados de los años 70 unos investigadores diseñaron una serie de técnicas destinadas a resolver los problemas de la confidencialidad y autenticidad de los documentos electrónicos. Tales técnicas se conocen en la actualidad de forma genérica como Criptografía de Clave Pública; el modelo más difundido es conocido como RSA, por las siglas de sus diseñadores: Rivest, Shamir y Adelman. Básicamente, consiste en lo siguiente: Mediante un sencillo programa de ordenador se generan parejas de números o claves. Estos números son por lo general muy grandes (cada uno ocupa una larga lista de dígitos) y están matemáticamente relacionados con su pareja. Uno de ellos se conoce como Clave Privada y el otro como Clave Pública. La generación de ambas claves es siempre pareja, y están, como se ha dicho, matemáticamente relacionada; de tal forma que, si dos claves públicas son distintas, también lo será sus correspondientes claves privadas, y viceversa. Cuando alguien desea utilizar o aplicar firma digital, debe estar provisto de su pareja de claves, la pública y la privada. La clave privada debe guardarse en secreto, mientras que la pública, como su propio nombre indica, se difunde libremente, utilizando incluso servicios o directorios diseñados expresamente para ello.



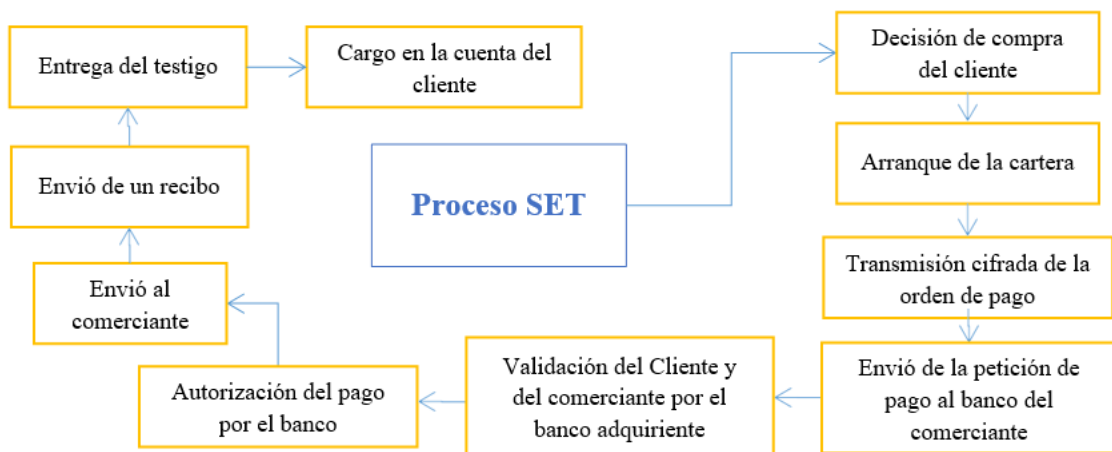
2.2.4. Protocolo SET

De acuerdo a (Jiménez González & Vega Castro, 2021), Secure Electronic Transaction SET, es un protocolo creado con el objetivo de brindar seguridad en el momento de realizar una transacción en internet, este protocolo fue creado única y exclusivamente para realizar transacciones electrónicas seguras ofreciendo servicios como:

- Autenticación
- Confidencialidad
- Integridad
- Intimidad
- Verificación inmediata
- No repudio

Las entidades que participan en SET son 3 básicamente, el banco o pasarela de pago, el comprador dueño de la tarjeta y el comerciante, y el proceso consta de 10 pasos como a continuación se explica en la gráfica:

Figura 6: Funcionamiento de Protocolo SET



Nota 9: (Jiménez González & Vega Castro, 2021)

El protocolo específico por SET detalla el formato de los mensajes, las operaciones criptográficas y las codificaciones que deben ser usadas.



2.2.5. Internet

De acuerdo a (Mogetta, Audisio, & Casco, 2018), el internet es la red de redes, es decir, la red de computadoras más extendida del planeta, que conecta y comunica a millones de personas en todo el mundo. Es el más extenso conjunto de redes de computadoras y equipos físicamente unidos mediante cables que conectan puntos de todo el mundo. Estos cables se presentan en muchas formas: desde cables de red local (varias máquinas conectadas en una oficina o campus) a cables telefónicos convencionales, digitales y de fibra óptica que forman las "carreteras" principales. Esta gigantesca Red se difumina en ocasiones porque los datos pueden transmitirse vía satélite, o a través de servicios como la telefonía celular.

Los clientes son las computadoras que utilizan el común de las personas para conectarse y consultar los datos, generalmente utilizando un módem y una línea telefónica.

Los servidores son las computadoras que guardan la información (host) y están conectadas las 24 horas atendiendo peticiones de datos, el correo electrónico, la transferencia de archivos o cualquier otro servicio brindado a esa red.

La WWW (World Wide Web) o simplemente la web es un conjunto de servidores distribuidos en todo el mundo que proveen información organizada en sitios, con cierta cantidad de páginas electrónicas relacionadas. La Web está basada principalmente en documentos escritos en HTML (HyperText Markup Language), un lenguaje de marcado que sirve principalmente para crear hipertexto en Internet. El hipertexto permite que cuando se hace clic en un hipervínculo o link (enlace) éste pueda abrir otra página del mismo sitio o dirigirse a otro sitio, incluso de distinto país. Además, el lenguaje HTML es válido para adecuar el aspecto visual de un documento e incluir objetos multimedia en el texto (imágenes, esquemas de diálogo, sonido, video, etc.). El HTTP (HyperText Transfer Protocol) es el Protocolo de Transferencia de Hipertexto o mecanismo de intercambio de información, base funcional de la WWW.

Web 2.0

De acuerdo a (Mogetta, Audisio, & Casco, 2018), el término está asociado a aplicaciones web que facilitan el compartir información, la interoperabilidad, el diseño



centrado en el usuario y la colaboración en la WWW. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos. Ejemplos de la Web 2.0: servicios de redes sociales (Facebook), de alojamiento de videos (YouTube), las wikis (Wikipedia), blogs, etc.

Web 3.0

De acuerdo a (Mogetta, Audisio, & Casco, 2018), describe la evolución del uso y la interacción de las personas en Internet tendientes a la transformación de la red en una base de datos, crear contenidos accesibles por múltiples aplicaciones non browser (sin conexión), inteligencia artificial, la Web 3D y la web semántica (agregado de metadatos que describen el contenido, el significado y la relación de los datos, para que sea posible evaluarlas automáticamente por máquinas).

2.2.6. Intranet

De acuerdo a (Mogetta, Audisio, & Casco, 2018) es un esquema de red donde se utiliza, sobre una plataforma de red organizacional o inter organizacional existente, con el medio de transporte entre sus unidades, el protocolo de internet, el TCP/IP, aquellas tecnologías avanzadas de publicación electrónica WEB(http), correo electrónico(email), archivos y recursos compartidos(file server, printer server, backup server, ftp server), acceso remoto (vpn, ssh), etc. brindando así todo el potencial del esquema cliente servidor en un ámbito acotado, seguro y racionalizado de acuerdo a los objetivos de eficacia y eficiencia estimados en dicha organización.

Seguridad: define el ámbito y normativa para asegurarlo de virus, violación de acceso, fuga de información, etc.

Integridad: protege los cambios o eliminación de datos no queridos.

Autenticación: control de ingreso a la red de acuerdo a los estándares de verificación de identidades.

Auditoría: control de la actividad en tiempo real y posterior, con posibilidad de retroalimentación del sistema (autoaprendizaje).



Disponibilidad y oportunidad: asegura la presencia de los servicios y datos requeridos en tiempo y forma predefinidos.

Las Intranets son posibles gracias a la concepción de la informática distribuida, que surgió en el mundo de la empresa en la década de los ochenta. Uno de los avances en este campo, la llegada de lo que se denominan Redes de Área Extensa, o WAN, ha hecho de las Intranets un fenómeno posible en la realidad actual. Cuando hablamos de informática distribuida nos referimos a las redes de ordenadores (las más conocidas son posiblemente las redes LAN), que han sustituido en parte a las tradicionales mainframes corporativas. A la hora de construir cualquier red de ordenadores, y, en especial, una Intranet, necesitaremos unos elementos básicos: En primer lugar, es necesario disponer de un hardware o soporte físico adecuado para la Intranet, que incluye:

PCs o Estaciones de Trabajo que actúan como Servidores de Web, bajo el modelo Cliente-Servidor.

2.2.7. Extranet – Servicios en la nube

De acuerdo a (Mogetta, Audisio, & Casco, 2018) es una especie de extensión de la Intranet a través de Internet. A ella normalmente pueden tener acceso ciertos usuarios en el – caso de una empresa: clientes, proveedores y trabajadores – para acceder a datos y aplicaciones de la organización.

Esto es diferente del sitio Web de la organización que es accesible para todos.

Al área de Extranet de una organización se accede por Internet, pero no es un lugar público, y sólo pueden acceder los usuarios autorizados por la propia organización mediante un proceso de autenticación en donde se debe ingresar un identificador de usuario y una clave. En cierta medida, la mayoría de las interfaces gráficas de usuario (GUI) se asemejan a un tablero de instrumentos. Sin embargo, algunos desarrolladores de productos emplean conscientemente esta metáfora para que el usuario reconozca al instante la similitud.



2.2.8. Navegar por la web

De acuerdo a (Mogetta, Audisio, & Casco, 2018), la tecnología Web –sobre la que se basa Internet, Intranet y Extranet– permite localizar, acceder, almacenar, administrar y transmitir información. El mecanismo principal para realizar esto consiste en la exploración de documentos, esta exploración se denomina navegación y los documentos que son los contenedores de la información se denominan páginas.

El concepto de navegar en la Web (Net surfing) se debe a la facilidad con la que es posible desplazarse a través de las páginas.

Para navegar por Internet el explorador que se utilice debe tener disponible una conexión activa, para que de esta manera se pueda acceder a los servicios de búsqueda de información y demás servicios disponibles en la Web; sin embargo, es posible utilizar las mismas técnicas de navegación en los sitios que conforman una Intranet ya que, como se dijo anteriormente, una Intranet es una especie de Web privada para compartir recursos dentro de una misma organización y no necesita una conexión activa a Internet para funcionar.

2.2.9. Navegador web

De acuerdo a (Mogetta, Audisio, & Casco, 2018) el navegador, explorador o browser son los términos con los que se denomina a los programas que permiten transitar por la Web. En el idioma inglés la palabra browser hace referencia a ojear un documento, acción bastante parecida a la que ocurre cuando un usuario navega por la red.

Este tipo de software es una aplicación diseñada para visualizar todo tipo de información y navegar por los sitios Web; cuenta con funcionalidades multimedia que permiten, además de visualizar información, reproducir videos, animaciones y sonidos.

Los navegadores proporcionan una interfaz gráfica interactiva que permite buscar, localizar, ver y administrar la información en la Web.

Estos programas de navegación permiten que, cualquiera sea la computadora que se utilice, la información de la Web esté siempre accesible ya que se encargan de codificar y decodificar el formato HTML adaptándolo a la computadora cliente.



Como ejemplo de navegadores podemos citar a Internet Mozilla Firefox, Opera, Google Chrome, Safari, entre otros.

2.2.10. Agencia de viajes

Turismo

De acuerdo a (López J. L., 2017) según el Profesor, José Ignacio de Arrillaga, experto en turismo, define el turismo “como todo desplazamiento voluntario y temporal determinado por causas ajenas al lucro, el conjunto de bienes, servicios y organización que cada nación determinan y hacen posibles esos desplazamientos, y las relaciones y hechos que entre estos y los viajeros tienen lugar”.

Según los profesores suizos Kurt Krapf y Walter Hunziker, de la Universidad de Berna, definen el turismo como “el conjunto de relaciones y fenómenos producidos por el desplazamiento y permanencia de personas fuera de su lugar de domicilio, en tanto que dichos desplazamientos y permanencias no estén motivados por una actividad lucrativa”.

Definición de agencia de viajes

De acuerdo a (Quispe, 2017), empresa que concibe, crea, planea, organiza y ejecuta programas de servicio de viajes para el organizador o el cliente en los que incluye normalmente alojamiento, alimentación, transporte de aproximación y local, así como excursiones en sitio y a los alrededores para el grupo de participantes en el evento a petición del cliente; bien sea en forma directa o como intermediarios entre los usuarios y los prestadores de servicios turísticos tanto nacionales como internacionales.

Tipos de agencias

- Agencia de Viajes y Turismo Mayorista: Aquella que proyecta, elabora y organiza todo tipo de servicios turísticos y viajes para ser ofrecidos a otras Agencias de Viajes y Turismo, no pudiendo ofrecer ni vender sus productos directamente al turista.
- Operador de Turismo: Aquel que proyecta, elabora, diseña organiza y opera sus productos y servicios dentro del territorio nacional, para ser ofrecidos y vendidos a



través de las Agencias de Viajes y Turismo Mayorista y Agencias de viajes y turismo minorista, pudiendo también ofrecerlos y venderlos directamente al turista.

Clasificación

Son Actividades de las Agencias de Viajes y turismo, de acuerdo a su clasificación, las siguientes:

- Agencias de viajes y turismo minoristas
 - Promoción del turismo en los ámbitos nacional y/o internacional.
 - Intermediación de programas organizados por las Agencias de Viajes y Turismo Mayoristas u Operadores de Turismo.
 - Intermediación de servicios de líneas aéreas.
 - Intermediación del servicio de alquiler de vehículos con y sin conductor.
 - Fletamento de aviones, barcos, autobuses, trenes y otros medios de transporte.
 - Organización y promoción de congresos, convenciones y otros eventos similares.
 - Contratación de Guías de Turismo.
 - Intermediación en la reserva y venta de boletos y pasajes en cualquier medio de transporte.
 - Intermediación en la reserva y contratación de hospedaje.
 - Intermediación de otros prestadores de servicios turísticos.
 - Asesoría y comercialización de programas y demás servicios turísticos, a nivel nacional e internacional.
 - Recepción y traslado de clientes.
 - Brindar orientación, información y asesoría al cliente.
 - Venta de pólizas de seguros de viaje.



- Alquiler y/o venta de útiles y equipos para la práctica de diversas modalidades de turismo, así como venta de souvenirs o artesanías.
- Agencia de Viajes y Turismo Mayorista
 - Promoción del turismo en los ámbitos nacional y/o internacional.
 - Representación de empresas intermediarias u Agencias de Viajes y Turismo no domiciliadas en el país.
 - Representación de líneas aéreas.
 - Representación de otros prestadores de servicios turísticos.
 - Alquiler de vehículos con y sin conductor.
 - Fletamento de aviones, barcos, autobuses, trenes y otros medios de transporte.
 - Contratación de Guías de Turismo.
 - Proyección, elaboración, organización y comercialización de programas y demás servicios turísticos, a nivel nacional e internacional.
- Agencia de viajes y Turismo Operador de Turismo
 - Promoción del turismo en los ámbitos nacional y/o internacional.
 - Proyección, elaboración, producción, organización, operación, asesoría y comercialización de programas y demás servicios turísticos, a nivel nacional e internacional.
 - Representación de empresas intermediarias u Agencias de Viajes y Turismo no domiciliadas en el país.
 - Representación de líneas aéreas.
 - Intermediación en la reserva y venta de boletos y pasajes en cualquier medio de transporte.
 - Intermediación en la reserva y contratación de hospedaje.



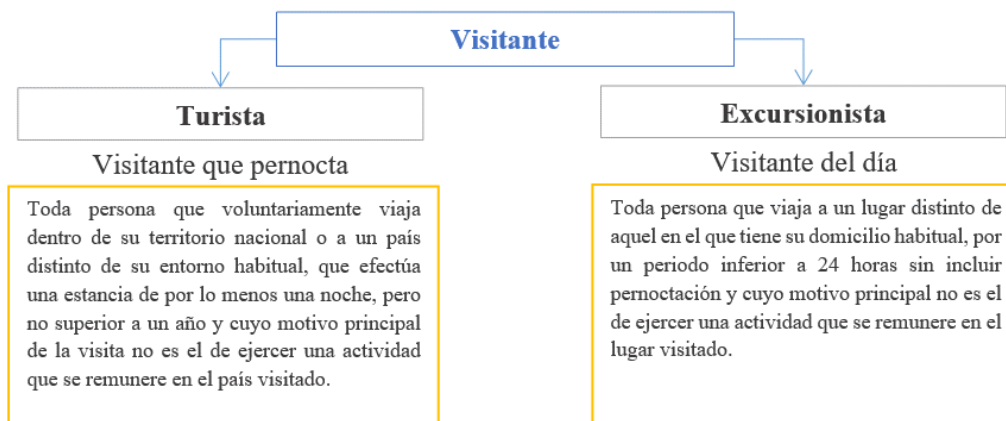
- Representación de otros prestadores de servicios turísticos.
- Alquiler de vehículos con y sin conductor.
- Fletamento de aviones, barcos, autobuses, trenes y otros medios de transporte.
- Organización y promoción de congresos, convenciones y otros eventos similares.
- Contratación de Guías de Turismo.
- Recepción y traslado de clientes.
- Brindar orientación, información y asesoría al cliente.
- Alquiler de útiles y/o venta de equipos para la práctica de diversas modalidades de turismo, así como venta de souvenirs o artesanías.

De acuerdo a (Mincetur, 2016), las actividades señaladas, serán prestadas por las Agencias de Viajes y Turismo, de acuerdo a su clasificación, en el territorio nacional, sin perjuicio de la contratación directa de tales servicios, por parte de los turistas, con las empresas de transporte, establecimientos de hospedaje y otros prestadores de servicios turísticos.

2.2.11. Visitante

De acuerdo a (Quispe, 2017), es toda persona que se desplaza a un lugar distinto al de su entorno habitual y cuya finalidad principal de viaje no es la de ejercer una actividad remunerativa en el lugar visitado. Los visitantes son parte de la población afectada por un problema que un proyecto de inversión pública, en el sector turismo, intenta solucionar. Se clasifican en dos grupos:

Figura 7: Visitante y su Clasificación

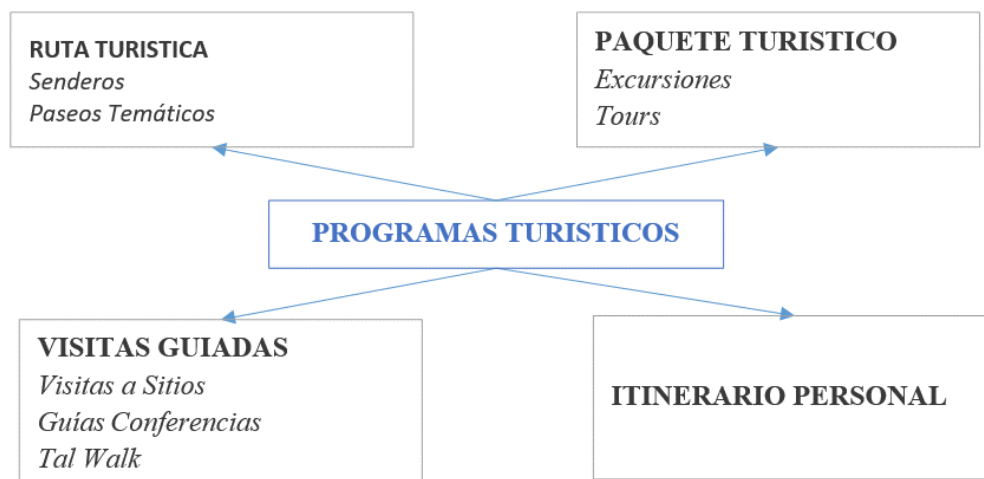


Nota 10: (Mincetur, 2016)

2.2.12. Programa turístico

Todo programa turístico se compone de un circuito o recorrido por un determinado sitio o destino y un itinerario donde se condensa las prestaciones incluidas. La expresión de ese programa se manifiesta de formas diversas, según su función y características, tal como se ilustra en la siguiente imagen.

Figura 8: El Programa Turístico



Nota 11: (Chan, 2018)

Paquete turístico

De acuerdo a (Chan, 2018), tiene por función la comercialización de las prestaciones turísticas de un destino y entre sus principales características se destacan la ganancia en



términos de tiempo que percibe el turista y el posibilitar la visita masiva a un destino sin alterar su capacidad de carga diaria.

Se le considera un conjunto de servicios debido a que el momento de su operación cada programa involucra servicios tales como:

- Alojamiento
- Transporte
- Alimentos, Bebidas y Otros
- Guiado
- Zonas Arqueológicas

Estos servicios pueden pertenecer a la misma organización o no, los cuales se distribuyen de acuerdo a las necesidades de la gente.

Según se trate de un programa de estructura simple o complejo, los servicios serán prestados en forma simultánea o cronológicamente.

Visita guiada

De acuerdo a (Chan, 2018), Tiene por función encaminar al visitante y brindarle información previamente seleccionada sobre un determinado sitio turístico. Entre sus principales características se encuentran: la interacción entre el guía y el visitante y la facilitación del encuentro entre el visitante y el medio.

Itinerario

De acuerdo a (Chan, 2018), Itinerario es el recorrido, el detalle de actividades y prestaciones incluidas en un programa. El recorrido define las rutas o caminos a tomar, las ciudades de visita y pernocte, los medios de transporte seleccionados para trasladarse de una ciudad a otra, los kilómetros y tiempos que se emplean en dicho traslado. Este servicio tiene una peculiaridad, para efectuarse, ósea para que el cliente lo consuma realmente. Este deberá trasladarse a un espacio físico definido.



2.2.13. Cadena de secuencia de servicios

Preventa

De acuerdo a (Quispe, 2017), antes de comenzar con el proceso de ventas, éste debe ser planificado en todos sus aspectos y en forma participativa entre los directivos y colaboradores. Nada debe resultar espontáneo pues los clientes finales (turista o visitante) y corporativos (agentes de viajes) son cada vez más exigentes.

Venta

De acuerdo a (Quispe, 2017), es importante distinguir que cuando vendemos, debemos diferenciar entre clientes finales (que compran vía Internet o directamente en las oficinas) y clientes corporativos (agencias de viajes).

Reserva

De acuerdo a (Quispe, 2017), planificar con tiempo las reservas, debido a que, al igual que las ventas, éstas se realizan con mucha anticipación. Una vez recibida la información de clientes el personal de reservas debe estar en la capacidad de distinguir entre clientes finales o clientes corporativos para procesar la información.

Operación y uso del servicio

De acuerdo a (Quispe, 2017), planificar las operaciones pudiendo ser semanales, quincenales o por meses, esto le ayudará a preparar el material de logística y otros con anticipación (Preparar el servicio o previsión del servicio).

Despedida del cliente

De acuerdo a (Quispe, 2017), El cierre del servicio puede concretarse en la medida de sus posibilidades adicionando en los servicios un valor agregado, como un traslado de cortesía, una cena show, un almuerzo, un tour por la ciudad u otro detalle adicional que lo diferencie de sus competidores.

Manejo documentario y reporte

De acuerdo a (Quispe, 2017), durante la cadena del servicio, se generan una serie de documentos y reportes administrativos, señalaremos aquellos de más importancia



(aunque cada empresa puede particularizar sus documentos) y que puedan ayudarle en el manejo de la empresa.

2.2.14. Comercio electrónico

De acuerdo a (Panorama del Comercio Electrónico, 2019), el Comercio Electrónico es, en estos momentos, un concepto que está revolucionando la percepción de los escenarios en los que se desarrollan las iniciativas empresariales y los mercados financieros.

Los elementos que hacen posible el Comercio Electrónico están ligados a las Tecnologías de Información y Comunicación, conformando una nueva infraestructura para los procesos de negocio. Permite el desarrollo de actividades empresariales en un entorno cada vez más globalizado, y con relaciones cada vez más virtuales, entre los distintos agentes que conforman la cadena de valor.

Para dar una definición de lo que es el comercio electrónico, se debe tener en cuenta que éste puede entenderse de distintas formas, según la perspectiva que se adopte. Por lo tanto, dado su empleo en los ámbitos técnicos y de organización empresarial, se puede abordar desde tres perspectivas, principalmente: desde la de las comunicaciones, la de los procesos de negocio y la temporal.

- Desde el punto de vista de las Comunicaciones, el comercio electrónico es aquel que distribuye información, productos, servicios o transacciones financieras, a través de Redes de Telecomunicación Multimedia, conformando estructuras empresariales de carácter virtual.
- Teniendo en cuenta los Procesos de Negocio, el comercio de red es el que utiliza las tecnologías que facilitan el soporte y la automatización de los flujos de trabajo y procedimientos de negocio de la empresa, con lo que ésta consigue eficiencias en los costes, servicios con una mejor calidad y ciclos de producción más cortos.
- Por último, desde una perspectiva Temporal, el comercio electrónico es el instrumento que permite establecer nuevos canales para el intercambio de productos, servicios e información en tiempo real.



Este tipo de comercio permite tanto disponer como transmitir información que se procesa, almacena, trata y vende de diferentes formas, haciendo que se genere valor y, por ende, creando una nueva forma de hacer negocio.

En definitiva y para acoplar los diferentes conceptos de comercio electrónico en una sola definición, se concluye que el comercio electrónico es aquel que consiste en el desarrollo de una actividad comercial, con multiplicidad de operaciones, que se puede realizar por vía telemática (electrónica) y basada en la cesión de productos, prestación de servicios e intercambio de datos (información), pudiendo realizarlos en tiempo real. (Anteportamlatinam Valero, 2014)

Aspectos claves

De acuerdo a (Panorama del Comercio Electrónico, 2019), el comercio electrónico ofrece una serie de ventajas respecto al comercio tradicional, están basadas en una serie de circunstancias de la sociedad actual.

- La globalización: La red permite la apertura de negocios en todo el mundo durante las 24 horas del día, por lo que aumentamos las posibilidades de éxito de nuestro negocio. Una capacidad de respuesta competitiva en tiempo real. Además, los costes de esta presencia global son reducidos.
- La Disminución de las cadenas de distribución: Hace que se produzca un acercamiento entre proveedor-cliente, teniendo dos efectos directos uno en el caso de la eliminación de intermediarios que podrían aumentar el valor final del producto de compra y, por otro lado, se establece una estrategia de fidelización con el cliente. Igualmente, los plazos y tiempos de respuesta se acortarán proporcionalmente, incluso pueden ser inmediatos sobre todo en oferta electrónica, como por ejemplo en programas informáticos, revistas y libros digitales, etc.
- Cliente anónimo versus cliente conocido: Una persona que entra a una tienda a comprar no deja “rastros” de su identidad. A lo sumo, paga con tarjeta de crédito donde sale su nombre o conoce bien al dueño y sabe cómo se llama. Pero no es “parte del negocio” conocer los datos personales del comprador como nombre,



dirección, teléfono, número de tarjeta de crédito, nacionalidad y otro. Esto, sin embargo, sí forma parte integral del modelo de negocios del comercio electrónico, ya que es la única forma de identificar a quién se le vende y a quién se le debe hacer la entrega. La gran ventaja que existe de este modelo de negocios es la capacidad para determinar las principales características del consumidor: gustos, zona geográfica donde reside, capacidad de compra, entre algunos de los aspectos a considerar. A pesar de ser más “impersonal”, es un sistema más interactivo y muchos lo califican de óptimo en este sentido.

- **Conocimiento del comportamiento del cliente:** En el e-commerce la empresa puede observar la conducta del consumidor sin hacerle sentir que esto es invasivo para él. Existen herramientas de medición on-line como Vividence y Accrue, que llevan un registro de muchos comportamientos: sitios Web visitados, duración de las visitas en un sitio, páginas visitadas en un sitio, contenido de listas de deseos y carritos de compras, adquisiciones, comportamiento en compras repetidas, número de visitantes que completan el proceso de compra y otras mediciones. Todo lo anterior no es posible (o no resulta costoso) en las compras tradicionales.
- **Una interfaz con el cliente basado en la tecnología:** En los negocios tradicionales los clientes realizan transacciones diariamente en persona o por teléfono con los empleados o gerentes de ventas. La interfaz basada en la tecnología permite que el cliente tenga una interacción a través de una pantalla. Esta interacción sucede con los cajeros automáticos, en los monitores de los PC/ordenadores, PDA entre otros dispositivos electrónicos.
- **Variedad Limitada versus Variedad Ilimitada:** En una librería caben muchos libros, pero no millones como es el caso de cualquier tienda electrónica. Además, en una tienda virtual se puede entregar herramientas de búsqueda que permitan rápidamente encontrar los productos que se desea por distintos criterios de búsqueda. Esa facilidad para hallar lo que se busca, sea lo que sea, es uno de los grandes atractivos del comercio electrónico.



- Asistencia en la venta: En una tienda tradicional un vendedor no puede atender a varios clientes a la vez, y si lo realiza estamos seguros que la atención no será de la mejor forma. De igual modo, si la variedad es muy grande es difícil que una persona pueda ser experta en todos y cada uno de los productos que se venden en una tienda tradicional.
- Utilización de espacio: Los términos del negocio se han invertido: un comercio tradicional usa el espacio físico para vender y por ello requiere de mucha superficie para llegar a más gente con mayor variedad de productos y de la tecnología para ahorrar costes, mediante un buen sistema de control de inventarios y facturación, por ejemplo. Una tienda virtual, en cambio, usa la tecnología para vender vía Internet y el espacio físico para ahorrar costes mediante un sistema logístico optimizado.

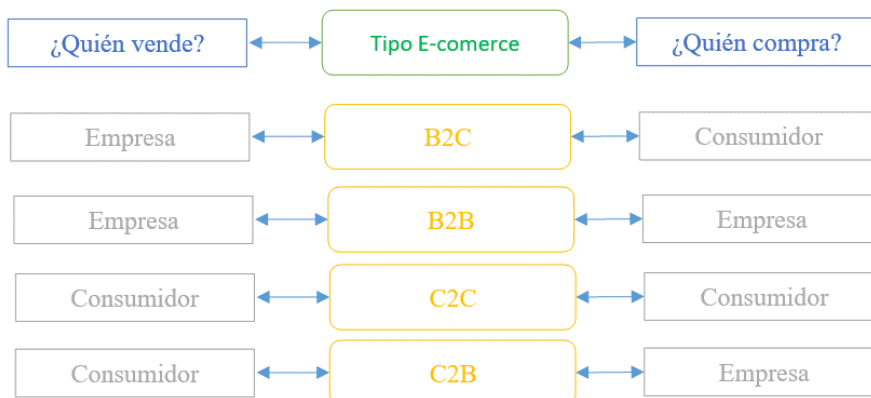
Tipo de comercio electrónico

Para poder establecer una clasificación de los distintos tipos de comercio electrónico que podemos existen, se va a tener en cuenta los siguientes factores:

- Quién interviene
- De qué manera intervienen
- Qué tipo de intercambio se realiza

A continuación, se explican los distintos tipos por medio de un gráfico:

Figura 9: Tipos de Comercio Electrónico



Nota 12: (Anteportamlatinam Valero, 2014)



Business To Consumer (B2C)

De acuerdo a (Anteportamlatinam Valero, 2014), la modalidad de comercio electrónico entre empresas y consumidores o B2C es la aplicación al mercado virtual de las estrategias habituales de venta del mercado tradicional. No obstante, este nuevo canal de venta presenta diferencias muy importantes.

Se conoce como Business to Consumer a la realización por medios electrónicos de todas o parte de las actividades que convencionalmente conducen al intercambio de productos entre una empresa y sus clientes. En otras palabras, es aquel tipo de comercio electrónico realizado entre empresas que ofrecen ciertos productos y los consumidores del mismo. Incluye desde la simple promoción por parte del vendedor de sus productos a través de la página web, hasta el cierre de la transacción económica originada por el intercambio, el pago del producto contra una tarjeta de crédito o débito e incluso, el control de las operaciones de distribución física o la distribución misma si se tratara de un producto que se pueda enviar total o parcialmente a través de Internet.

Business To Business (B2b)

De acuerdo a (Anteportamlatinam Valero, 2014), se puede considerar que el comercio electrónico entre empresas ha sido el origen de los negocios on-line. En el mundo físico se relaciona este tipo de transacción con el canal mayorista.

El “Business to Business” es el tipo de comercio electrónico en el que una empresa intercambia bienes o servicios con otra empresa. Es decir, una empresa actúa de proveedor de la otra, que es el cliente de la primera. La transacción comercial habitual es el productor del bien o servicio, que lo vende a un mayorista quien a su vez lo distribuirá al consumidor final. Se refiere a la cadena de suministro y a la relación comercial entre empresas mediante medios electrónicos.

Customer To Customer (C2c)

Se entiende por “Customer to Customer” a la relación de intercambio directa entre un individuo (o varios) y otro individuo (o varios) a través de Internet. Es el menos significativo desde el punto de vista de la planificación y dirección de empresas y de sus formas análogas en el comercio tradicional: mercadillos, subastas no organizadas, etc.

Una estrategia C2C para Internet sería aquella que define un negocio cuyo objetivo es facilitar la comercialización de productos y/o servicios entre particulares, como por ejemplo eBay, sirviendo la empresa como mera intermediaria y cobrando por sus servicios. (Anteportamlatinam Valero, 2014)

Customer To Business (C2b)

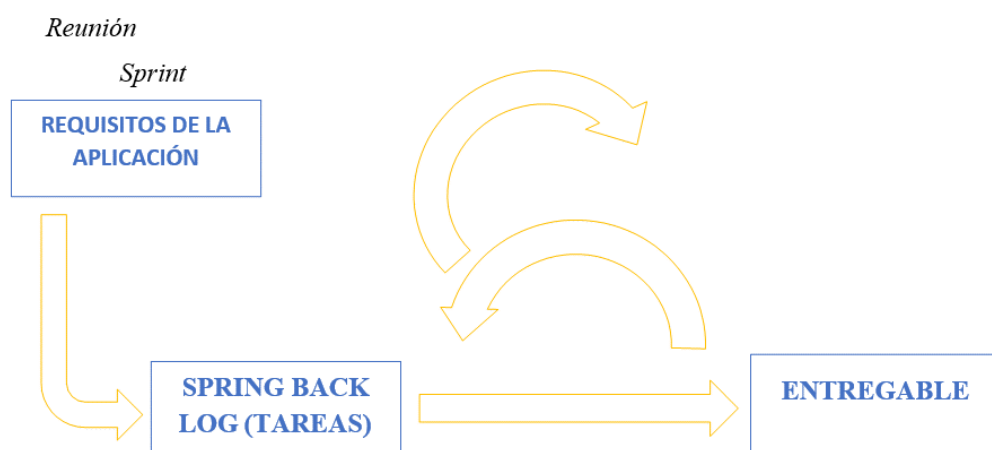
El “Customer to Business” se refiere al intercambio, mediante el soporte tecnológico de Internet, de información sobre el consumo a cambio de alguna compensación.

Son ahora los consumidores los que proveen de información de marketing a las empresas que precisan de ellas para su propia gestión comercial, a la inversa de lo que ocurría en la modalidad B2C. (Anteportamlatinam Valero, 2014)

2.2.15. Metodología de desarrollo

Scrum

Figura 10: Marco de trabajo SCRUM



Nota 13: Creación propia

De acuerdo a (Dimes, 2015), scrum es como un salvavidas para aquellas empresas que enfrentan dificultades al seguir la metodología de cascada o que ni siquiera están utilizando metodología alguna para el desarrollo de software. Scrum es un marco de referencia para crear software complejo y entregarlo a tiempo de una forma más sencilla

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), scrum se basa en la teoría de control de procesos empírica o empirismo.



El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

Transparencia

Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo. Por ejemplo:

Todos los participantes deben compartir un lenguaje común para referirse al proceso; y aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de “Terminado”.

Inspección

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

Adaptación

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores. Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación, tal y como se describen en la sección Eventos de Scrum del presente documento.



- Reunión de Planificación del Sprint (Sprint Planning Meeting)
- Scrum Diario (Daily Scrum)
- Revisión del Sprint (Sprint Review)
- Retrospectiva del Sprint (Sprint Retrospective)

El equipo SCRUM (SCRUM Team)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son auto organizados y multifuncionales. Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

El Dueño de Producto (Product Owner)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). La gestión de la lista del producto incluye:

Expresar claramente los elementos de la Lista del Producto; Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible; Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo; Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello



en lo que el equipo trabajará a continuación; Asegurar que el Equipo de Desarrollo entienda los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

El Equipo de Desarrollo (Development Team)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

- Son auto organizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente despleables;
- Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto;
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
- Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla; y,



- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El Scrum Master

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

Eventos de Scrum

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), en Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

El Sprint

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.



Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

Durante el Sprint

No se realizan cambios que puedan afectar al Objetivo del Sprint (Sprint Goal);

Los objetivos de calidad no disminuyen; y el alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

Reunión de Planificación de Sprint (Sprint Planning Meeting)

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

Objetivo del Sprint (Sprint Goal)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), el objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Es creado



durante la reunión de Planificación del Sprint. El objetivo del Sprint ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los elementos de la Lista del Producto seleccionados ofrecen una función coherente, que puede ser el objetivo del Sprint. El objetivo del Sprint puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

Scrum Diario (Daily Scrum)

El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

El Scrum Diario optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint. Cada día, el Equipo de Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo auto organizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint.

Revisión de Sprint (Sprint Review)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Lista de Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor.

Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para Sprints de un mes. Para Sprints más cortos, se reserva un tiempo proporcionalmente menor. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)



Retrospectiva de Sprint (Sprint Retrospective)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), la Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

Artefactos de Scrum

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

Lista de Producto (Product Backlog)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), la Lista de Producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única Nota de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.

La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)



Lista de Pendientes del Sprint (Sprint Backlog)

De acuerdo a (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014), la Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint.

Según se requiere nuevo trabajo, el Equipo de Desarrollo lo añade a la Lista de Pendientes del Sprint. A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan pasa a ser considerado innecesario, es eliminado. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)

Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no. (Schwaber & Sutherland, La Guía Definitiva de Scrum: Las Reglas del Juego, 2014)



2.2.16. Lenguaje de programación

Backend

PHP

Figura 11: Logo de PHP



Nota 14: (PHP, 2020)

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga (PHP, 2020).



Laravel

Figura 12: Logo Laravel



Nota 15: (Laravel, 2020)

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP. (García J. M., 2015)

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas.

Frontend

Javascript

De acuerdo a (Eguiluz, 2020), JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas



escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios

HTML

De acuerdo a (Mora, 2020), HTML es un lenguaje de marcas (etiquetas) que se emplea para dar formato a los documentos que se quieren publicar en la WWW. Los navegadores pueden interpretar las etiquetas y muestran los documentos con el formato deseado. En este capítulo se presentan los conceptos básicos y avanzados (tablas, formularios y marcos) de HTML. El capítulo analiza con una guía de estilo: pequeños consejos que pueden ayudar a lograr mejores páginas web.

CSS

De acuerdo a (Eguiluz, 2020), CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

BOOTSTRAP

De acuerdo a (Cochran, 2019), Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. A diferencia de muchos frameworks web, solo se ocupa del desarrollo front-end.

Bootstrap incluye tres carpetas: CSS, JS, FONTS, que pueden ser incluidos en la raíz de un proyecto.



- **CSS:** Incluyen archivos bootstrap.css y la versión minimizada bootstrap.min.css los cuales deben estar incluidos.
- **JS:** Incluyen los archivos bootstrap.js y la versión minimizada bootstrap.min.js este archivo contiene todas las librerías de JavaScript, lo que hace que Bootstrap sea más dinámico e interactivo.
- **FONTS:** Son los mismos archivos, sino que cambiar el formato para que se adecuen en los diferentes navegadores.

2.2.17. Tecnología

Ajax

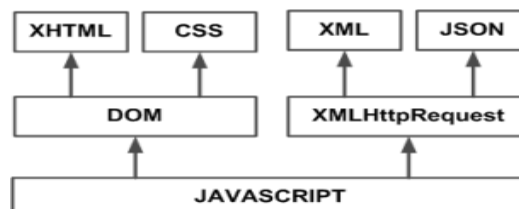
De acuerdo a (Eguiluz, 2020) , el término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML"

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Figura 13: Tecnologías agrupadas bajo el concepto de AJAX



Nota 16: (Eguiluz, 2020)



2.2.18. Microsoft Sql Server

De acuerdo a (Marques, 2011), es un Sistema de bases de datos profesional de Microsoft. Contiene una variedad de características y herramientas que se pueden utilizar para desarrollar y administrar base de datos y soluciones de todo tipo basadas en ellas.

Figura 14: Componentes de SQL SERVER



Nota 17: (Marques, 2011)

De acuerdo a (Marques, 2011), el Motor de Base de datos es el servicio principal para almacenar, procesar y proteger los datos. Asimismo, proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones de base de datos exigentes. Dependiendo del Motor de base de datos tenemos las componentes Service Broker y Replicación.

Características

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además, permite administrar información de otros servidores de datos.



T-SQL

T-SQL (Transact-SQL) es el principal medio de interacción con el Servidor, el cual permite realizar las operaciones claves en SQL Server, incluyendo la creación y modificación de esquemas de base de datos, inserción y modificación de datos en la base de datos, así como la administración del servidor como tal. Esto se realiza mediante el envío de sentencias en T-SQL y declaraciones que son procesadas por el servidor y los resultados (o errores) regresan a la aplicación cliente (Marques, 2011).

Cliente nativo de SQL

Cliente Nativo de SQL, es la biblioteca de acceso a datos para los clientes de Microsoft SQL Server versión 2005 en adelante. Implementa de forma nativa soporte para las características de SQL Server, incluyendo la ejecución de la secuencia de datos tabular, soporte para bases de datos en espejo de SQL Server, soporte completo para todos los tipos de datos compatibles con SQL Server, conjuntos de operaciones asíncronas, las notificaciones de consulta, soporte para cifrado, así como recibir varios conjuntos de resultados en una sola sesión de base de datos (Marques, 2011).

2.2.19. Servidor

Servidor en la nube

Un servidor de la nube es una potente infraestructura física o virtual que lleva a cabo el almacenamiento de procesamiento de aplicaciones e información. Los servidores en la nube se crean utilizando software de virtualización para dividir un servidor físico en varios servidores virtuales. Las organizaciones utilizan un modelo de infraestructura como servicio (IaaS) para procesar cargas de trabajo y almacenar información. Pueden utilizar una interfaz online para acceder de forma remota a las funciones del servidor virtual. (IBM, 2018)

Principales características:

- La infraestructura de computación puede ser física virtual o una combinación de las dos, dependiendo del caso de uso.
- Tiene todas las capacidades de un servidor local.



- Habilita que los usuarios procesen cargas de trabajo intensivos y almacenen volúmenes de información extensos.
- Se utiliza una API para acceder bajo demanda a servicios automáticos.
- Da a los usuarios la opción de pagar mensualmente o por consumo.

Los usuarios pueden optar por un plan de hosting compartido que se escala en función de las necesidades

2.2.20. Arquitectura modelo MVC

Cuadro comparativo arquitectura 3 capas y MVC

Figura 15: Modelo vista controlador + 3 capas

MVC	3 CAPAS
La arquitectura es triangular es decir que hay una relación entre la vista y el controlador, entre el controlador y el modelo y entre la vista y el modelo.	La arquitectura es lineal es decir que no hay una comunicación directa entre las diferentes capas.
Cada componente es independiente del otro.	Cada capa se centraliza en algún específico, además las capas van desde lo general hacia lo particular según corresponda
Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.	La información se centraliza en la aplicación ya que las capas están vinculadas.

Nota 18: (Autoría propia)

Para el desarrollo de nuestro proyecto nos hemos apoyado en la arquitectura MVC, ya que este modelo simplifica mucho el trabajo diario. El MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Models y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura.



Modelo vista controlador (MVC)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.

El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. (Universidad de Alicante, 2015)

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).



- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición *al modelo puede ser "Obtener tiempo de entrega (nueva orden de venta)".

Las vistas son responsables de:

- Recibir datos del modelo y muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

El flujo que sigue el control generalmente es el siguiente:

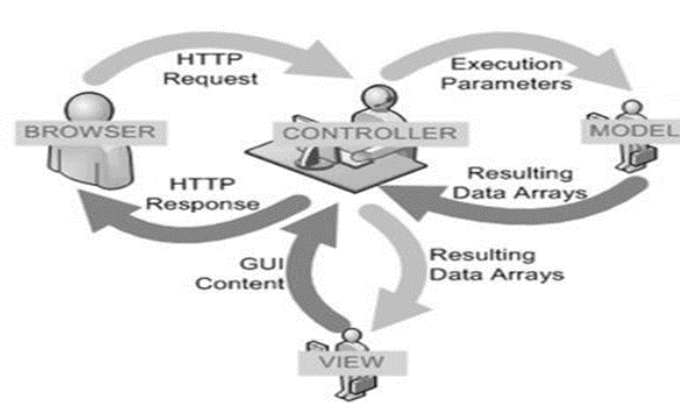
- 1) El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- 2) El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- 3) El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- 4) El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón



observador para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

- 5) De acuerdo a (Delgado, 2015), la interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Figura 16: Flujo de proceso MVC



Nota 20: Autoría propia

2.2.21. Dominio

Un Dominio es un nombre único que se utiliza para identificar tu negocio en internet, es importante saber elegir el nombre de tu sitio web (dominio), ya que este permitirá a los usuarios poder ubicarlos en el internet.

Un domino posee las siguientes partes

- Nombre: Identificador del sitio web
- Extensión: Parte final de la dirección que determina el rubro de tu sitio web.



Tipo de extensiones

- .com: Utilizadas para cualquier tipo de sitios web
- .net: Para empresas relacionadas con internet y la tecnología.
- .org: Utilizadas para organizaciones
- .info: Utilizada para sitios web de información
- .es, .eu, .mx: Extensiones que corresponden a los países, que identifica la ubicación geográfica del negocio.
- .gov: Utilizadas para páginas de los EE. UU
- .edu: Utilizadas para sitios web con fines educativos.

Figura 17: Arquitectura de Desarrollo del Software



Nota 22: (Autoría propia)

2.2.22. Herramientas complementarias

Microsoft office Excel

De acuerdo a (Corinne Hervo, 2013), excel es una aplicación desarrollada por Microsoft y distribuida en el paquete de Office para usarse en Windows o Macintosh. Presenta una interfaz intuitiva y amigable con archivos de ayuda incorporados.

Excel, es una hoja de cálculo que permite trabajar con tablas de datos, gráficos, bases de datos, macros, y otras aplicaciones avanzadas. Ayudando en el cálculo de ejercicios



aritméticos y siendo de gran utilidad diversas áreas como educación, administración, finanzas, producción, etc.

Esta herramienta nos permitió trabajar de manera ordenada cuadros descriptivos de procesos e historias de usuarios para la aplicación que se va a desarrollar.

Wamp server

El Wamp server es una aplicación que nos permite gestionar un Servidor de Páginas Web, al instalarlo también instala el motor de Base de Datos MYSQL y soporta diversas extensiones. Técnicamente es un freeware (gratis), y corre bajo el Sistema Operativo Windows para sistemas Linux es recomendable usar el Xamp.

Funcionalidades de Wamp Server son muy completas y fáciles de usar. Con Wamp Server, se puede:

- Administrar sus servicios de Apache y MySQL
- Cambiar en línea / fuera de línea (dar acceso a todo el mundo o solamente local host)
- Instalar y cambiar Apache, MySQL y PHP comunicados
- Administrar la configuración de los servidores
- Acceder a sus registros
- Acceder a sus archivos de configuración
- Crear alias

Wamp server se está utilizando en el proyecto como un entorno para desarrollo web por el cual se podrá crear nuestra aplicación web con PHP; también podemos manejar nuestra base de datos en MySql utilizándolo como servidor.

El uso de Wamp permite servir paginas HTML a internet, además de gestionar los datos en nuestra base de datos.



Capítulo III: Desarrollo, implementación o transferencia tecnológica

3.1. Desarrollo e implantación del Sistema de Información Web (SIW) “Turismo Smart Versión 2.5”

De acuerdo al ámbito situacional en los puntos anterior ya descritos dentro del operador turístico “PERU JAMUY”, la solución que se propuso fue el de desarrollar un sistema información web denominado Turismo Smart Versión 2.5 que permitió fortalecer los procesos internos de la empresa, centralizar información de los clientes, y automatizar el proceso de generación de paquetes turísticos.

El sistema información web desarrollado está basado en los procesos definidos en ámbito de los servicios turísticos: Captación y actualización de datos de los clientes y brindar servicios turísticos del mercado nacional e internacional. Ambos procesos son fundamentales en la definición del perfil de los paquetes turísticos generados por el sistema informático en web, ya que si no se cuenta con su información no se pueden desarrollar los procesos que continúan, como la definición y ejecución de programas de fidelización, gestión de la información, entre otros, los cuales se desarrollarán en futuros proyectos.

El desarrollo del proyecto se gestionó con la metodología de desarrollo ágil SCRUM, se aplicó las buenas prácticas que esta metodología plantea y se utilizó los artefactos necesarios para realizar una correcta gestión.

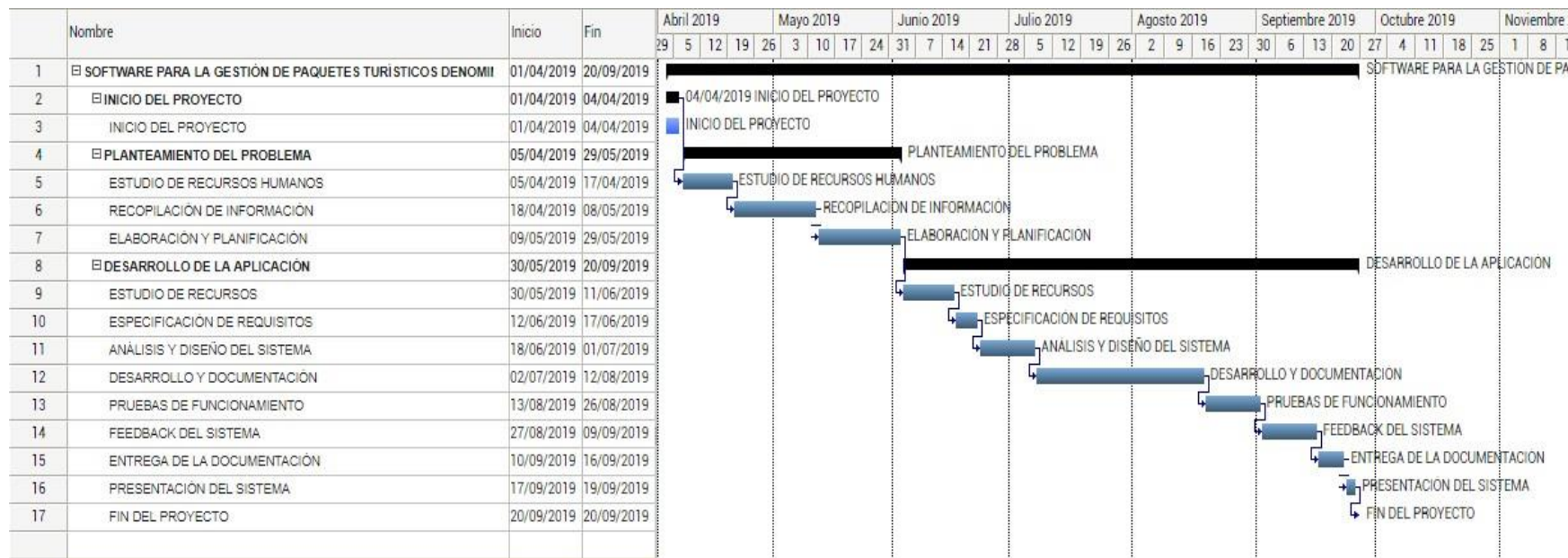
Para la implementación del sistema informático web del proyecto se utilizó todas las herramientas informáticas mencionadas en el punto anterior llamado AMBITO DE INFLUENCIA TEORICA.



3.2. Cronograma y recursos

3.2.1. Administración del proyecto – Diagrama de Gantt

Figura 18: Cronograma de Actividades



Nota 21: Creación propia

Es una herramienta grafica muy común que tiene como objetivo mostrar la estimación en tiempo de lo previsto para las tareas asignadas y las actividades por un tiempo determinado. Desde un inicio, el diagrama Gantt no relaciona la interacción entre las actividades, la posición que tiene cada tarea a lo largo del tiempo permite identificar la independencia que existe entre ellas.



3.3. Estudio de factibilidad

En esta fase de la investigación se ha tomado como criterios los estudios de factibilidad del sector técnico y financiero, cuyo objetivo es describir la viabilidad de cumplimiento del trabajo de investigación aplicativo que ya se implementó denominado sistema información web Turismo Smart Versión 2.5.

3.3.1. Factibilidad técnica

Se puede afirmar que la viabilidad técnica está garantizada para el desarrollo del Sistema de Información Web (SIW) propuesto en este proyecto.

Descripción del Hardware:

- 01 servidor en la nube:
 - 8 GB Memoria RAM
 - 100 GB Disco SSD
 - 1TB Transferencia
 - Proveedor VPS (GoDaddy)
- 03 computadores personales:
 - 12 GB Memoria RAM
 - 500 GB Disco SSD
 - Acceso a Internet
 - Sistema operativo “Windows”

Descripción del software

- Sistema Operativo: Windows Server 2016
- Servidor de Base de datos: SQL Server
- IIS 8.5
- Lenguaje de programación para desarrollar:



- PHP 7.2 (acrónimo recursivo de PHP: Hypertext Preprocessor) Es el lenguaje más usado por las empresas dado que es un código abierto de presencia web que cuenta con su propio manejador de base de datos.
- JavaScript (abreviado comúnmente JS) Es un lenguaje Front-End que permite dar dinamismo a las aplicaciones web, también es un facilitador de envío de información del cliente hacia el servidor.
- Framework Laravel 6.2 (MVC) Es un framework de código abierto desarrollado en PHP presentando un marco de trabajo en MVC esto ha permitido consolidar sus ventajas así mismo cuenta con diferentes servicios que a través de enlaces se pueden utilizar.
- PAGE: es un template, el cual contiene un conjunto de herramientas que facilitan al desarrollador utilizarlas para la creación de un proyecto web contando con formularios, botones personalizados, cuadros para enlazar, menús para navegar por la página web entre otros. Se encuentra desarrollado en HTML, CSS y algunas librerías de JavaScript
- SQL Server 2016: Es un gestor de Base de datos abierto que trabaja muy bien con el lenguaje de PHP encargado de almacenar información, gestionar las vistas, procedimientos almacenados y triggers como control de auditoría.
- Tratamiento y creación de imágenes: Las imágenes son editadas por el área administrativa que tienen a su cargo revisar las dimensiones y el tipo de archivo que son requisitos para cargarlas en la web.

3.3.2. Factibilidad económica

Para poder estimar la factibilidad económica debemos tomar como referencia el análisis de costo y beneficio que está relacionado con el uso funcional del software en la empresa hoy en día. La proyección es que los beneficios siempre sean superiores a lo invertido.

El lenguaje en el que se desarrolló el software fue PHP que su uso es de código abierto de libre licencia, así como el gestor de base de datos SQL Server 2016. Los costos destinados a la implementación son bastante bajos dado que la empresa no tendrá la necesidad de gastar en capacitaciones por que cuenta con un nivel de ayuda con un



manual en donde se describirá todo el flujo del sistema. Adicional a ello el no habrá la necesidad de utilizar una máquina para implantar el sistema web dado que el servidor estará en la nube, y cumplen con los requisitos indispensables para que el sistema pueda trabajar adecuadamente.

Tabla 6: Recurso humano para la implantación

RECURSO HUMANO			
Nombre	Cantidad	Costo unitario	Costo total
Programador	1	2,000	2,000
Capacitador	1	2,000	2,000
Soporte	1	2,000	2,000
Total, Recurso humano			6,000

Nota 23: Creación propia

Tabla 7: Recursos tecnológicos para la implantación

RECURSO TECNOLÓGICO			
Hardware			
Nombre	Cantidad	Costo unitario	Costo total
Servidor	1	2,000	2,000
Computadoras	3	1,500	4,500
TV/Monitor 50'	1	2,500	2,500
Total, Hardware			9,000
Software			
Sistema de Información Web (SIW) Turismo Smart Versión 2.5	1	10,000	10,000
Total, Software			10,000

Nota 24: Creación propia

Para la implantación del Sistema de Información Web (SIW), el costo total es de S/. 25,000.00 soles, esta inversión permitirá ser utilizado el sistema en otra MYPE del sector turismo.



3.3.3. Factibilidad operativa

El presente trabajo de investigación aplicada es factible y operativa, dado que se cuenta con el conocimiento para que las MYPES del sector turismo puedan expandir su negocio llegando a más clientes ofreciendo sus productos y servicios siendo casi imperceptible el costo para lograr ello, el proceso de implementación ya se realizó. Los dueños han manifestado la satisfacción en el funcionamiento del Sistema de Información Web (SIW) y que es sencilla y fácil de uso, también cubre todas las expectativas, solicitudes, necesidades proporcionando información de forma confiable y oportuna. Recurso Humano: Las personas involucradas para la implementación de este Sistema de Información Web (SIW) la describimos en la siguiente tabla:

Tabla 8: Recurso humano para el desarrollo

#	Cargo	Funciones
1	Analista Programador	Son los encargados de ejecutar el análisis seguidodel desarrollo cumpliendo con todas las actividades, así también brindar mantenimiento al Portal Web, vigilando que su marcha siempre sea correcta.
2	Desarrollador Web	Es el encargado de realizar el desarrollo delPortal Web de la mano con el analista.

Nota 22: Creación propia

3.4. Normas y roles del Sistema de Información Web (SIW)

3.4.1. Normas internas

Una norma interna es una manifestación de la política a la cual se debe satisfacer y expresar como una limitación o no variante al momento de hacer el análisis para el modelo del negocio. A continuación, se muestra las Reglas de Negocio del operador turístico PERU JAMUY.

- Las oficinas del operador turístico “PERU JAMUY” deben enviar información de los datos de la oficina requeridos por el sistema.



- Las oficinas del operador turístico “PERU JAMUY” deben enviar cada fin de mes lista de paquetes turísticos destacados como máximo 3 cada mes.
- Las oficinas del operador turístico “PERU JAMUY” deben cumplir con las reservas de paquetes turísticos registrados en el sistema por los clientes.
- Las oficinas del operador turístico “PERU JAMUY” tiene el compromiso de la publicidad de la página para posicionar la página que la presentación del Sistema de Información Web (SIW).

3.4.2. Definición de roles del proceso de implementación del Sistema de Información Web (SIW)

Tabla 9: Recurso humano para el desarrollo

Definición de roles del proceso de implementación del Sistema de Información Web (SIW)	
Scrum Master:	Brain Mijail Garrido Cusipaucar
Product Owner:	Pyme: “PERU JAMUY”
Equipo:	Brain Mijail Garrido Cusipaucar

Nota 23: Creación propia

Scrum master

Se encargó de la planificación, realizar seguimiento, gestionar el proyecto en todo su proceso, coordinar el grupo de desarrollo, elaborar el informe plasmando la mejora del proyecto, costos, plazos de entrega y términos de calidad.

- El Sistema de Información Web (SIW) se realizó a través de una planificación de actividades generales.
- Tiene la facultad de aceptar o rechazar lo que el trabajo de equipo da como resultado.
- Tiene como objetivo promover los valores y normas de la metodología Scrum.
- Tiene la facultad de remover dificultades.



- Se encargó de hacer seguimiento para garantizar que el equipo es productivo y funcional.
- Tiene la facultad de permitir la cooperación entre los roles y funciones.

Product Owner

Es la persona que se encargó de elaborar la lista de funcionalidades, planear cuando dará inicio cada sprint y revisar el producto cuando se termine cada sprint para concluir si se cumplió con lo requerido. EQUIPO: Sus vitales funciones son:

- Comprometerse cuando se inicia cada sprint desarrollando todas las funcionalidades solicitadas en el tiempo determinado
- Ser los responsables directos de la entrega del producto cuando se termina el sprint.
- Desarrollar el Sistema de Información Web (SIW)
- Las funciones de las áreas comprometidas en el presente Sistema de Información Web (SIW) se describen en el presente cuadro.

3.5. Fase análisis de requerimientos del Sistema de Información Web (SIW)

El operador turístico “PERU JAMUY” es una institución que ofrece la venta de un servicio turístico a sus diferentes clientes nacionales e internacionales en sus distintas oficinas, hoy en día se logró mejorar el servicio y a través de la implantación del Sistema de Información Web (SIW) Turismo Smart Versión 2.5. En esta fase de análisis y estudio se realizó mediante entrevistas verbales a todas las áreas, revisando antecedentes y visualizándose in situ, lográndose determinar los requerimientos los cuales los describimos en los siguientes puntos:

- Que la empresa cuente con un posicionamiento global a través de la web para la venta del servicio turístico en donde pueda mostrar información acerca de la empresa.
- Ofrecer servicios turísticos en línea.
- Contar con secciones en donde mostrar sus categorías y subcategorías.



- Contar con los servicios de Pago Online líderes del mercado turístico.
- Realizar constantemente promociones de servicios turísticos.
- Poder hacer seguimiento de los servicios turísticos contratados por los clientes
- Mostrar noticias relacionados a sus servicios turísticos ofertados por la empresa.
Ofrecer servicios de marketing al turista.

3.5.1. Funciones de las áreas

Tabla 10: Recurso humano actual del negocio

Área	Funciones
Área Administrativa	<p>Planificación, Organización y control de los indicadores de gestión de la empresa.</p> <ul style="list-style-type: none"> • Seguimiento de las funciones de las áreas • Reclutamiento, registro y control de personal. • Medición de indicadores de ventas de paquetes turísticos • Balance mensual de ventas de paquetes turísticos nacionales e internacionales • Balance mensual de las visitas diarias a la página. • Balances de ventas no concluidas. • Medición de indicadores de ventas como página • Balance de usuarios indirectos registrados en la página por mes. • Atención de las consultas realizadas por nuestros clientes directos e indirectos, vía teléfono o la web
Área Comercial	<p>Encargado de buscar en el mercado las oportunidades de negocio:</p> <ul style="list-style-type: none"> • Búsqueda de Alianza estratégica con empresas. • Evaluación y emisión de paquetes turísticos, cotización al cliente directo (Empresas).



-
- Modificación o reafirmación de la Pre Cotización para la aceptación final del cliente.
 - Registro de la Cotización para nuestra área contable.
 - Crear y enviar reporte para el área de sistemas de las Empresas del rubro turístico.
 - Enviar lista de servicios turísticos al área de sistemas de las Empresas que han contratado el servicio
 - Crear modelos de promociones para nuestros clientes indirectos, previo acuerdo con nuestras áreas de ventas de servicios turísticos
 - Seguimiento y envío de correo a nuestra cartera de clientes en cuanto a mencionadas promociones de paquetes turísticos.

Área Sistemas

Gestiona las incidencias o requerimientos de la empresa:

- Registro clientes directos (empresas)
- Registro de usuarios
- Registro de paquetes turísticos
- Registro de Categorías.
- Registro de Subcategorías
- Registros de Promociones de paquetes turísticos
- Mantenimiento de usuarios y roles para el control adecuado de la sección asignada.
- Capacitación a nuestros clientes directos (empresas) del manejo de su cuenta
- Reporte y control de clientes directos e indirectos
- Reporte y control de Usuarios.
- Reporte y control de Productos.
- Reporte y control de Categorías.
- Reporte y control de Subcategorías.
- Emisión del reporte de pedidos de servicios turísticos entregados y/o el status según los requerimientos.



**Área de
Contabilidad**

- Emisión de códigos de venta en el correo final que se le envía al cliente tras una venta del paquete turístico.
- Emisión de Comprobante a los clientes directos (nacionales e internacionales)
- Reporte y control de Cobranza a los clientes directos
- Registro y control de los códigos de ventas de los servicios contratados
- Pagos de mantenimiento de infraestructura (alquiler de oficina,).
- Apoyo y asesoramiento en la medición de los indicadores de ventas de los paquetes turísticos en la página en general para una debida toma de decisiones del área administrativa.
- Pagos de personal. - Control de ingresos y egresos. - Análisis de la rentabilidad y pérdidas de la empresa.

Nota 24: Creación propia

3.6. Metodología de desarrollo – SCRUM

3.6.1. Intervención metodológica

En el presente trabajo de investigación aplicada, utilizó la metodología Scrum la cual se aplicó en cada fase, así mismo se tiene previsto identificar las necesidades verídicas, poder estimar tiempos certeros, Conseguir la participación de todo el equipo de trabajo, que el desarrollo se refleje funcionalmente en el producto, a través de la retrospectiva se pueda contrastar lo avanzando al final del sprint seguido poder proponer un entregable manejándose por versiones, así obtener como resultado el producto final con esto se definen nuevos requerimientos como son el mantenimientos para darle prolongación al producto.



3.7. Fase de análisis de requerimiento

3.7.1. Requerimientos del aplicativo web

Tabla 11: Recurso humano para el desarrollo

ID	REQUERIMIENTOS FUNCIONALES	REQUERIMIENTOS NO FUNCIONALES
1.	Creación del diseño de Interfaz del Back-end Sistema de Información Web (SIW). (Administración, Cotizado)	Interfaz intuitiva con el logo de la empresa.
2.	Integración de la plantilla PAGE JavaScript con el framework Laravel	
3.	Creación de la Base de datos	Las tablas deben tener toda la información que se tiene de la empresa.
4.	Acceso al sistema (Login)	
5.	Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores).	Mantenimiento de fácil acceso e intuitivo de fácil acceso e intuitivos para los usuarios.
6.	Mantenimiento de Usuarios (Creación, Edición, Eliminación).	
7.	Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)	
8.	Mantenimiento de Servicios y Paquetes Turísticos (Creación, Edición, Eliminación).	
9.	Administración de las áreas y Staff	
10.	Cotizar Servicios	



-
11. Creación de Reportes y Control
 12. Creación del diseño de Interfaz Frontend Interfaz intuitiva con el logo de la del sistema de información Web (cliente empresa indirecto).
 13. Integración de la plantilla PAGE, JavaScript con el framework Laravel
 14. Actualización de la Base de datos del Sistema Las tablas deben tener toda la información que se tiene de la empresa.
 15. Creación de un formulario de registro del cliente. Mantenimiento de fácil acceso e intuitivo de fácil acceso e intuitivos para los usuarios.

Nota 25: Creación propia

3.7.2. Product backlog

El product backlog (pila de producto) o llamado también historias de usuario es aquí cuando el analista funcional genera un vínculo de confianza con el usuario directo para entender el desarrollo de sus actividades. Se consideró lo siguientes:

- Prioridad en el negocio (PN): Este fue medido en función al nivel: Alta, Media y Baja, el Product Owner fue quien se encargó de asignarla.
- Importancia del desarrollo (ID): Los ponderados estarán en el rango del 1 al 100 este será asignado por el Product Owner seguido de los miembros del equipo Scrum
- Tiempo Estimado (TS): Los ponderado fueron en el rango del 1 al 20 este fue asignado por el Product Owner seguido de los miembros del equipo Scrum.

La historia de usuario se dividió por módulos de esta manera podremos interpretarlo como bloques haciendo más fácil la programación las cuales son:



- Módulo de interface
- Módulo de Base de datos
- Modulo Administrador: Esta sección contendrá todas las funcionalidades que el administrador del sistema va a utilizar (Cliente Directo)
- Modulo Cliente Indirecto/Cotizador: Esta sección contendrá todas las funcionalidades por los cuales los usuarios indirectos van a interactuar con el sistema.
- Modulo página de inicio
- Módulo Login

Tabla 12: Historia de usuario HU01

HISTORIA DE USUARIO

ID: HU01 **Usuario:** Pyme: “PERU JAMUY”

Descripción de la Historia: Creación del diseño de Interfaz del Back-end Sistema de Información Web (SIW).(Administración, Cotizador)

Nivel de Prioridad: Alta **Valor del Desarrollo:** 100

Periodo Estimado: 11d **Modulo Establecido:** Página de inicio

Descripción: Se creó una interfaz donde el usuario accede a las distintas opciones de forma intuitiva y amigable

Observaciones:

Nota 26: Creación propia

Tabla 13: Historia de usuario HU02

HISTORIA DE USUARIO

ID: HU02 **Usuario:** Pyme: “PERU JAMUY”

Descripción de la Historia: Integración de la plantilla PAGE con el framework Laravel.

Nivel de Prioridad: Alta **Valor del Desarrollo:** 95



Periodo Estimado: 8d

Modulo Establecido: Página de inicio

Descripción: Se creó una interfaz donde el usuario puede visualizar el Sistema de Información Web (SIW) en distintos dispositivos de forma correcta.

Observaciones:

Nota 27: Creación propia

Tabla 14: Historia de usuario HUO3

HISTORIA DE USUARIO

ID: HU03

Usuario: Analista Funcional

Descripción de la Historia: Creación de la Base de datos

Nivel de Prioridad: Alta

Valor del Desarrollo: 85

Periodo Estimado: 8d

Modulo Establecido: Base de datos

Descripción: Las tablas muestran una nomenclatura de la empresa, así como sus datos principales para el proceso de sus actividades.

Observaciones:

Nota 28: Creación propia

Tabla 15: Historia de usuario HUO4

HISTORIA DE USUARIO

ID: HU04

Usuario: Pyme "PERU JAMUY"

Descripción de la Historia: Acceso al sistema

Nivel de Prioridad: Alta

Valor del Desarrollo:

Periodo Estimado: 8d

Modulo Establecido: Login

Descripción: Para el Login se utiliza un usuario y una contraseña el cual ya exista en la base de datos.

Observaciones:



Nota 29: Creación propia

Tabla 16: Historia de usuario HU05

HISTORIA DE USUARIO

ID: HU05

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismo y que requieren del servicio (Administración de operadores).

Nivel de Prioridad: Alta

Valor del Desarrollo: 80

Periodo Estimado: 8d

Modulo Establecido: Administrador

Descripción: El presente formulario solicita los datos principales de los clientes que desean adquirir nuestro servicio.

Observaciones:

Nota 30: Creación propia

Tabla 17: Historia de usuario HU06

HISTORIA DE USUARIO

ID: HU06

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Creación de una página con un formulario para el mantenimiento de usuarios.

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 8d

Modulo Establecido: Administrador

Descripción: El presente formulario solicita los datos principales del usuario, el rol que desempeña en la empresa. Esta cuenta se utilizará para loguear al sistema.

Observaciones:

Nota 31: Creación propia





Tabla 18: Historia de usuario HU07

HISTORIA DE USUARIO

ID: HU07

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Acceso o Mantenimiento de roles.

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 8d

Modulo Establecido: Administrador

Descripción: En el presente formulario se deberán registrar los roles que se diferenciaron por los privilegios que se den a cada uno, esto quiere decir que la visualización de las opciones del menú dependerá de los permisos que se den, para empezar, serán 3: administrador, cliente y operador.

Observaciones:

Nota 32: Creación propia

Tabla 19: Historia de usuario HU08

HISTORIA DE USUARIO

ID: HU08

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Mantenimiento de Servicios y Paquetes Turísticos

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 5d

Modulo Establecido: Administrador

Descripción: El usuario podrá registrar los datos del servicio turístico que quiera adquirir.

El usuario podrá cargar o escoger los paquetes turísticos representados por imágenes.

Se considera los campos como imagen, precio, descripción, fecha.

Observaciones: Puede marcar las opciones de destacado.

Nota 33: Creación propia



Tabla 20: Historia de usuario HU09

HISTORIA DE USUARIO

ID: HU09

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Administración de las áreas y Staff

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 5d

Modulo Establecido: Administrador

Descripción:

- El usuario accede a la información del Sistema de Información Web (SIW)
- El usuario visualiza las visitas al fanpage del Sistema de Información Web (SIW)

Observaciones:

Nota 34: Creación propia

Tabla 21: Historia de usuario HU10

HISTORIA DE USUARIO

ID: HU10

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Cotizar Servicios.

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 5d

Modulo Establecido: Administrador

Descripción: El usuario puede cotizar o administrar a través de proceso, programa, vista y reporte en el fanpage del Sistema de Información Web (SIW).

Observaciones:

Nota 35: Creación propia



Tabla 22: Historia de usuario HU11

HISTORIA DE USUARIO

ID: HU11

Usuario: Pyme “PERU JAMUY”

Descripción de la Historia: Creación de Reportes y Control

Nivel de Prioridad: Alta

Valor del Desarrollo: 75

Periodo Estimado: 5d

Modulo Establecido: Administrador

Descripción:

- Usuarios (clientes directos): Puede visualizar todos los usuarios con la siguiente información: nombre, email, password, fecha ingreso, rol, oficina
- Roles: Se puede visualizar todos los usuarios con la siguiente información: nombre, estado, fecha ingresa.
- Servicios turísticos: Se puede visualizar todos los servicios turísticos ofrecidos en el Sistema de Información Web (SIW) con la siguiente información: nombre, descripción, estado, fecha ingreso.
- Clientes: Puede visualizar todos los clientes con la siguiente información: nombre, apellido, DNI, email, password, fecha de cumpleaños, teléfono, celular, dirección, estado, fecha ingreso.
- Cotizador: Puede visualizar todas las cotizaciones de los paquetes turísticos reservados por un Cotizador.
- Reserva de Paquetes Turísticos: Se podrá visualizar todas las reservas de los paquetes turísticos, mostrando la siguiente información: nombre del emisor, cantidad, total, estado, oficina, fecha reserva, hora reserva, dirección, teléfono, nombre del receptor o cliente, apellido paterno, apellido materno, mensaje, fecha ingreso.
- Compras, Proveedores asignados, Caja, Seguimiento de caja, Liquidación y producción, Productividad

Observaciones:

Nota 36: Creación propia



Tabla 23: Historia de usuario HU12

HISTORIA DE USUARIO

ID: HU12

Usuario: Cliente

Descripción de la Historia: Creación del diseño de Interfaz del Front-end del Sistema de Información Web (SIW).

Nivel de Prioridad: Alta

Valor del Desarrollo: 100

Periodo Estimado: 5d

Modulo Establecido: Página de inicio

Descripción: El cliente al ingresar al Sistema de Información Web (SIW) puede visualizar una atractiva página donde podrá visualizar las promociones de paquetes turísticos vigentes, el número de visita al sitio web, de acuerdo a su nivel de acceso puede ingresar a las distintas pestañas con las que cuenta el sistema. Puede visualizar a un bloque bastante llamativo de servicios turísticos destacados por su precio o novedad.

Observaciones:

Nota 39: Creación propia

Tabla 24: Historia de usuario HU13

HISTORIA DE USUARIO

ID: HU13

Usuario: Cliente

Descripción de la Historia: Integración de la plantilla PAGE con el framework Laravel

Nivel de Prioridad: Alta

Valor del Desarrollo: 95

Periodo Estimado: 3d

Modulo Establecido: Página de inicio

Descripción: Se creará una interfaz donde el usuario pueda visualizar el Sistema de Información Web (SIW) en distintos dispositivos de forma correcta

Observaciones:

Nota 39: Creación propia



Tabla 25: Historia de usuario HU14

HISTORIA DE USUARIO

ID: HU14

Usuario: Analista Funcional

Descripción de la Historia: Actualización de la Base de datos del Sistema

Nivel de Prioridad: Alta

Valor del Desarrollo: 95

Periodo Estimado: 3d

Modulo Establecido: Página de inicio

Descripción: Las tablas deben mostrar nomenclatura de la empresa, así como sus datos principales para el proceso de sus actividades.

Observaciones:

Nota 40: Creación propia

Tabla 26: Historia de usuario HU15

HISTORIA DE USUARIO

ID: HU15

Usuario: Cliente

Descripción de la Historia: Creación de un formulario de registro del cliente.

Nivel de Prioridad: Alta

Valor del Desarrollo: 90

Periodo Estimado: 3d

Modulo Establecido: Página de inicio

Descripción: El cliente deberá registrarse como cliente de nuestro Sistema de Información Web (SIW) ingresando sus nombres, apellidos, DNI, email, contraseña, fecha de nacimiento, teléfono, celular y dirección o su persona jurídica.

Observaciones:

Nota 41: Creación propia



Tabla 27: Product backlog

PRODUCT BACKLOG	
Usuarios	Historias de usuarios
Clientes	<ul style="list-style-type: none"> • Creación de un formulario de registro del cliente. • Creación de un formulario de logueo.
Encargado de administrativa y cotización	<ul style="list-style-type: none"> • Creación de Reportes y Control • Cotizar Servicios
Encargado del área de sistemas	<ul style="list-style-type: none"> • Integración de la plantilla PAGE, JavaScript con el framework Laravel. • Creación de la base datos • Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). • Creación de una página con un formulario para el mantenimiento de usuarios.

Nota 43: Creación propia

3.8. Release backlog (lista de historia de usuario por orden de importancia)

Tabla 28: Release backlog

ID	HISTORIA DE USUARIOS	PRIORIDAD	TIEMPO ESTIMADO	IMPORTANCIA
1	Creación del diseño de Interfaz del Back-end Sistema de Información Web (SIW). (Administración, Cotizador)	Alta	11d	100
2	Integración de la plantilla PAGE JavaScript con el framework Laravel	Alta	8d	100



3	Creación de la Base de datos	Alta	8d	95
4	Acceso al sistema (Login)	Alta	8d	90
5	Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores).	Alta	8d	85
6	Mantenimiento de Usuarios (Creación, Edición, Eliminación).	Alta	8d	85
7	Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)	Alta	8d	85
8	Mantenimiento de Servicios y Paquetes Turísticos (Creación, Edición, Eliminación).	Alta	5d	85
9	Administración de las áreas y Staff	Alta	5d	85
10	Cotizar Servicios.	Alta	5d	85
11	Creación de Reportes y Control	Alta	5d	85
12	Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	Alta	5d	85
13	Integración de la plantilla PAGE, JavaScript con el framework Laravel	Alta	3d	80
14	Actualización de la Base de datos del Sistema	Alta	3d	75



15	Creación de un formulario de registro del cliente.	Alta	3d	70
----	--	------	----	----

Nota 44: Creación propia

3.9. Definición de los sprint

Se puede definir la velocidad en que se realiza el desarrollo de cada uno de los Sprint según la importancia que tiene cada una de las historias de usuario seguido del tiempo que se estimó en el trabajo del equipo de Scrum para el desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5 y la dedicación al mismo.

Tabla 29: Definición de los sprint

Equipo SCRUM	Jornada laboral	Horas de trabajo al proyecto por semana	Semana de Trabajo por mes	Total, de horas	Total, de días laborables para el Sistema
Brain Mijail Garrido Cusipaucar	8 horas	40 horas	8 semanas	320 horas	48 días
Brain Mijail Garrido Cusipaucar	8 horas	40 horas	8 semanas	320 horas	48 días
Tiempo que se estimó en el desarrollo del Sistema de Información Web (SIW)					96 días

Nota 45: Creación propia

El tiempo estimado dado como entrega del Sistema de Información Web (SIW) Turismo Smart Versión 2.5 y las horas asignadas que estarán dentro del programa de trabajo se esperan tener algunos obstáculos y entretenimientos pero que están dentro de las propuestas dentro del desarrollo del sistema, este escenario lo tiene presente el Product Owner por esa razón se compromete con un 90% como tiempo estimado que el mismo va a realizar.

Velocidad estimada del Sprint = Días Hombre Disponible X Factor de Dedicación



36 = 30 X 90%

Conforme a la rapidez calculada para el cumplimiento de cada Sprint y considerando el nivel de jerarquía definido en cada ítem la historia de usuario se continúa con el siguiente paso que es agrupar las mismas e ir separando por bloques basándose en los tiempos, considerando una cantidad por cada Sprint para el proyecto, dando el siguiente resultado:

Tabla 30: Estimación del tiempo para el sprint 01

Sprint Nro. 01				
Modulo	Historia de Usuario	Prioridad	Importancia	Estimación de tiempo
MPI	Creación del diseño de Interfaz del Back-end Sistema de Información Web (SIW). (Administración, Cotizador)	Alta	100	11 días
MPI	Integración de la plantilla PAGE Javascript con el framework Laravel	Alta	100	8 días
MBD	Creación de la Base de datos	Alta	95	8 días
ML	Acceso al sistema (Login)	Alta	90	8 días
Tiempo estimado para el Sprint Nro. 01				35 días

Nota 46: Creación propia



Tabla 31: Estimación del tiempo para el sprint 02

Sprint Nro. 02				
Modulo	Historia de Usuario	Prioridad	Importancia	Estimación de tiempo
MA	Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores).	Alta	85	8d
MA	Mantenimiento de Usuarios (Creación, Edición, Eliminación).	Alta	85	8d
MA	Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)	Alta	85	8d
MA	Mantenimiento de Servicios y Paquetes Turísticos (Creación, Edición, Eliminación).	Alta	85	5d
Tiempo estimado para el Sprint Nro. 02				29 días

Nota 47: Creación propia



Tabla 32: Estimación del tiempo para el sprint 03

Sprint Nro. 03				
Modulo	Historia de Usuario	Prioridad	Importancia	Estimación de tiempo
MA	Administración de las áreas y Staff	Alta	85	5d
MA	Cotizar Servicios.	Alta	85	5d
MA	Creación de Reportes y Control	Alta	85	5d
MPI	Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	Alta	85	5d
Tiempo estimado para el Sprint Nro. 03				20 días

Nota 48: Creación propia

Tabla 33: Estimación del tiempo para el sprint 04

Sprint Nro. 04				
Modulo	Historia de Usuario	Prioridad	Importancia	Estimación de tiempo
MPI	Integración de la plantilla PAGE, JavaScript con el framework Laravel	Alta	80	3d
MBD	Actualización de la Base de datos del Sistema	Alta	75	3d
MPI	Creación de un formulario de registro del cliente.	Alta	70	3d
Tiempo estimado para el Sprint Nro. 04				9 días

Nota 49: Creación propia



3.10. Planificación de los sprint

Tabla 34: Planificación del sprint 01

Sprint Nro. 01	
Fecha de inicio	30-05-2019
Fecha de culminación	19-07-2019
Revisión del progreso	Según la programación el último o primer día de la semana se realizó una revisión. Las fechas por revisar con las siguientes: <ul style="list-style-type: none">- 08-06-2019- 15-06-2019- 22-06-2019- 29-06-2019- 06-07-2019- 13-07-2019
Tarea asignada al desarrollador	<ul style="list-style-type: none">• Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW)• Integración de la plantilla PAGE con el framework Laravel• Creación de la Base de datos• Acceso al sistema (Login)

Nota 50: Creación propia



Tabla 35: Planificación del sprint 02

Sprint Nro. 02	
Fecha de inicio	19-07-2019
Fecha de culminación	30-08-2019
Revisión del progreso	Según la programación el último o primer día de la semana se realizó una revisión. Las fechas a revisar con las siguientes: <ul style="list-style-type: none">- 27-07-2019- 03-08-2019- 10-08-2019- 17-08-2019- 24-07-2019
Tarea asignada al desarrollador	<ul style="list-style-type: none">• Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismo y que requieren del servicio (Administración de operadores).• Mantenimiento de Usuarios (Creación, Edición, Eliminación).• Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)• Mantenimiento de Servicios y Paquetes Turísticos (Creación, Edición, Eliminación).

Nota 51: Creación propia



Tabla 36: Planificación del sprint 03

Sprint Nro. 03

Fecha de inicio 30-08-2019

Fecha de culminación 27-09-2019

Revisión del progreso del Según la programación el último o primer día de la semana se realizó una revisión. Las fechas a revisar con las siguientes:

- 06-09-2019
- 13-09-2019
- 20-09-2019

Tarea asignada al desarrollador

- Administración de las áreas y Staff
- Cotizar Servicios.
- Creación de Reportes y Control
- Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).

Nota 52: Creación propia

Tabla 37: Planificación del sprint 04

Sprint Nro. 04

Fecha de inicio 27-09-2019

Fecha de culminación 25-10-2019

Revisión del progreso del Según la programación el último o primer día de la semana se realizó una revisión. Las fechas a revisar con las siguientes:

- 04-10-2019
- 11-10-2019
- 18-10-2019

Tarea asignada al desarrollador

- Información de la empresa, áreas, staff
- Integración de la plantilla PAGE, JavaScript con el framework Laravel
- Actualización de la Base de datos del Sistema
- Creación de un formulario de registro del cliente.

Nota 53: Creación propia



3.11. Sprint BurnDown chart

Ser realizado un cuadro llamado Taskboard (Cuadro de tareas) de desarrollo donde se inició el Sistema de Información Web (SIW) el cual incluyó todas las historias de usuario y la condición establecida desde el inicio por cada uno de los Sprint.

Tabla 38: Taskboard inicial de desarrollo

Inicio: 30-05-2019		Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Fin: 27-10-2019				
	Historia de Usuario	Pendiente	En curso	Hecho
Sprint Nro. 01	<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 	X		
	<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 	X		
	<ul style="list-style-type: none"> Creación de la Base de datos 	X		
	<ul style="list-style-type: none"> Acceso al sistema (Login) 	X		
	<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 	X		
Sprint Nro. 02	<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 	X		
	<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección 	X		



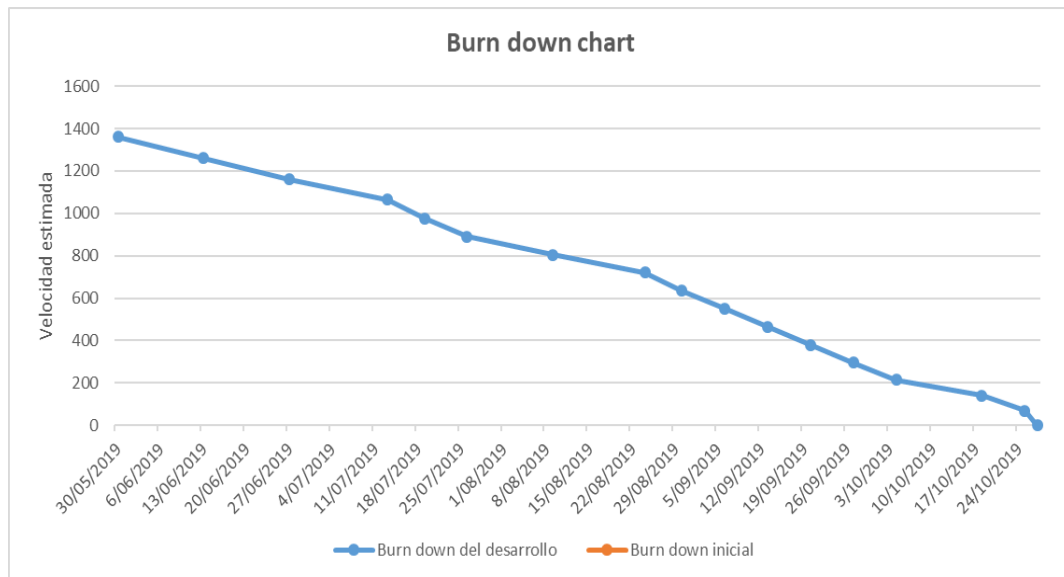
	asignada (Creación, Edición, Eliminación)	
Sprint Nro. 03	• Mantenimiento de Servicios y Paquetes	X
	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X
	• Actualización de la Base de datos del Sistema	X
	• Creación de un formulario de registro del cliente.	X

Nota 54: Creación propia

A continuación, se muestra la figura BurnDown Chart que dio inicio al Sistema de Información Web (SIW) Turismo Smart Versión 2.5 en la cual se observa la velocidad aproximada del desarrollo del sistema.



Figura 19: Gráfico de Trabajo (BurnDown Chart). Semana N° 0



Nota 55: Creación propia

3.12. Desarrollo del sistema

Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 1 de la Semana 01, 02 teniendo como historia de usuario “Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW)” se puso en marcha.

Tabla 39: Taskboard semana 01, 02

	Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
	Fin: 13-06-2019			
	01, 02			
	Historia de Usuario	Pendiente	En curso	Hecho
Sprint Nro. 01	• Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW)		X	
	• Integración de la plantilla PAGE con el framework Laravel		X	



Sprint Nro. 02	• Creación de la Base de datos	X
	• Acceso al sistema (Login)	X
	• Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores).	X
	• Mantenimiento de Usuarios (Creación, Edición, Eliminación).	X
	• Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)	X
	• Mantenimiento de Servicios y Paquetes	X
Sprint Nro. 03	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X

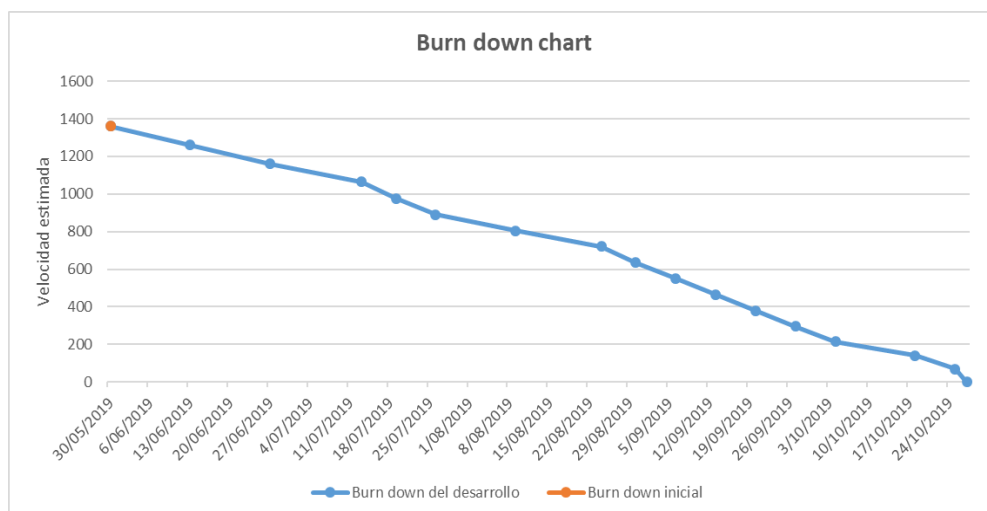


Sprint Nro. 04	<ul style="list-style-type: none"> Integración de la plantilla PAGE, JavaScript con el framework Laravel 	X
	<ul style="list-style-type: none"> Actualización de la Base de datos del Sistema 	X
	<ul style="list-style-type: none"> Creación de un formulario de registro del cliente. 	X

Nota 56: Creación propia

En la siguiente Figura se muestra el adelanto de las semanas N° 01, 02, las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo".

Figura 20: Gráfico de Trabajo (BurnDown Chart). Semana N° 01, 02

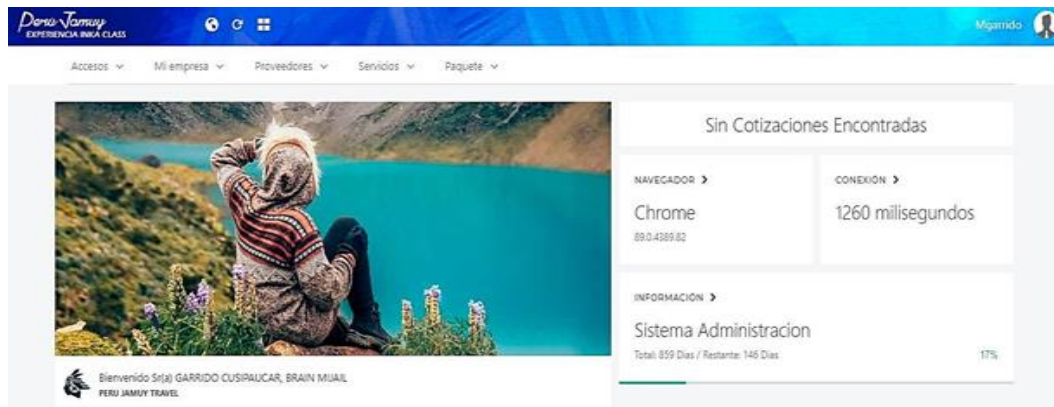


Nota 57: Creación propia

Semana 01, 02 a la fecha 13/06/2019

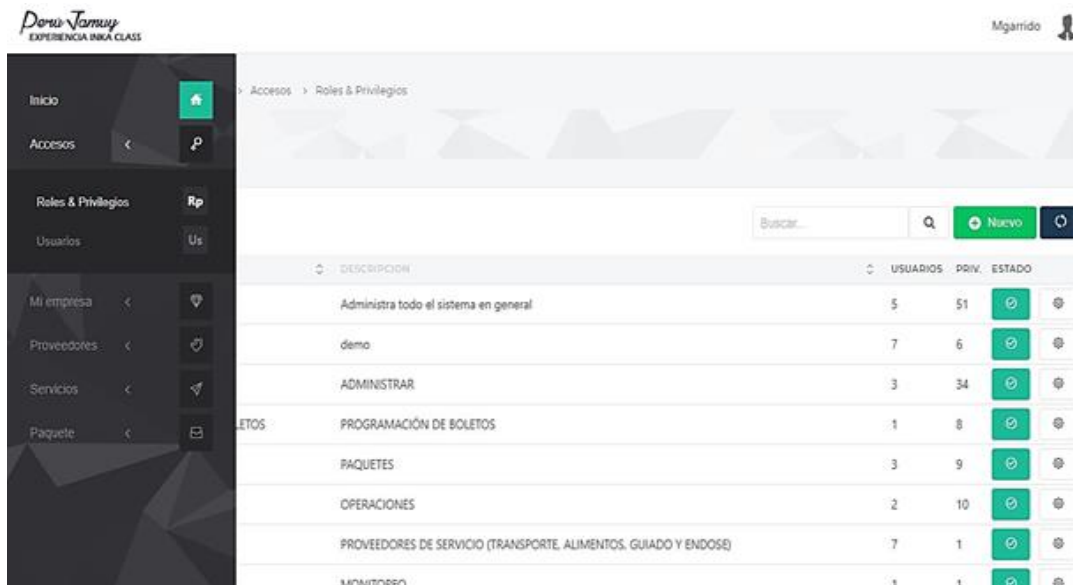
Se muestra la historia de usuario “Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) Turismo Smart Versión 2.5” y “Integración de la plantilla PAGE con el framework Laravel” se encuentra finalizada.

Figura 21: Diseño de Interfaz del Back-end e integración de plantilla



Nota 56: Creación propia

Figura 22: Integración de plantilla framework Page



Nota 57: Creación propia

Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 01 de la Semana 2 teniendo como historia de usuario “Creación del diseño de Interfaz del Back-end del Portal” e “Integración de la plantilla PAGE con el framework Laravel” se encuentra finalizada y la “Creación de la Base de datos” se puso en marcha.



Tabla 40: Taskboard semana 01, 02

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 01, 02		Fin: 13-06-2019	Pendiente	En curso	Hecho
		Historia de Usuario			
Sprint Nro. 01		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos 		X	
		<ul style="list-style-type: none"> Acceso al sistema (Login) 		X	
Sprint Nro. 02		<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 	X		
		<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 	X		
		<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 	X		



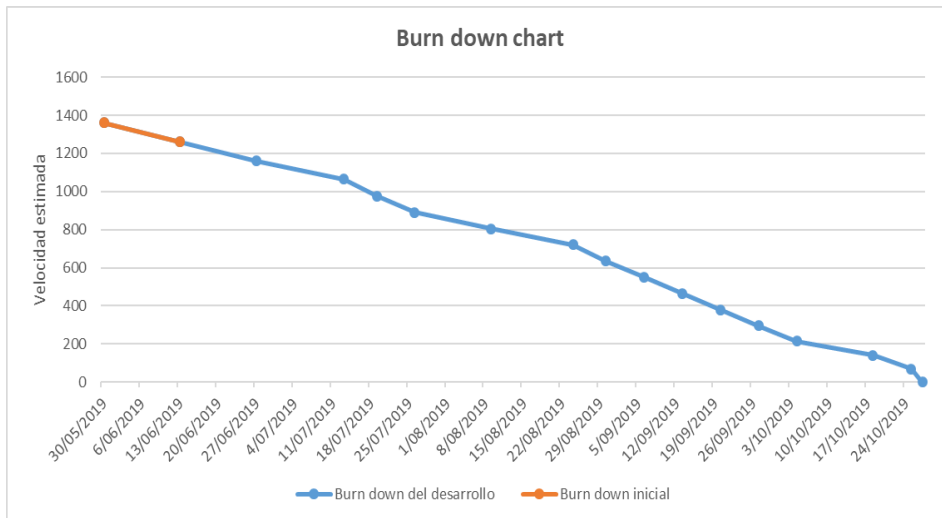
Sprint Nro. 03	• Mantenimiento de Servicios y Paquetes	X
	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X
	• Actualización de la Base de datos del Sistema	X
	• Creación de un formulario de registro del cliente.	X

Nota 58: Creación propia

La siguiente imagen presenta el adelanto de la semana N°01, 02 nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo".



Figura 23: Gráfico de Trabajo (BurnDown Chart). Semana N° 01, 02



Nota 59: Creación propia

Tabla 41: Tabla funcional semana 01, 02

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 01	VERSION DE EJECUCION	PF-MPI01
		FECHA EJECUCION	13/06/2019
TAREA:	<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) Integración de la plantilla PAGE con el framework Laravel 	MODULO DEL SISTEMA	MPI

Descripción del caso de prueba: El siguiente paso es efectuar pruebas con relación al diseño del Administrador e integración con PAGE y Laravel.

1. CASO DE PRUEBA

- a. Precondiciones



- Navegador Chrome y Firefox
- Hacer uso de una Pc, Tablet y Celular

b. Pasos de la prueba

- Validar que cada uno de los módulos mantengan un estándar.
- Validar que las opciones de ingreso sean accesibles y amigables.
- Validar con el DevTools (Depurador de Chrome) las etiquetas de integración con PAGE y el framework Laravel.

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los módulos mantienen un estándar.	X		Se validó satisfactoriamente.	X	
Los módulos son accesibles y amigables	X		Se validó satisfactoriamente.	X	
El administrador presenta las etiquetas de integración.		X	Se validó que existen las etiquetas de PAGE y las del Framework Laravel	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

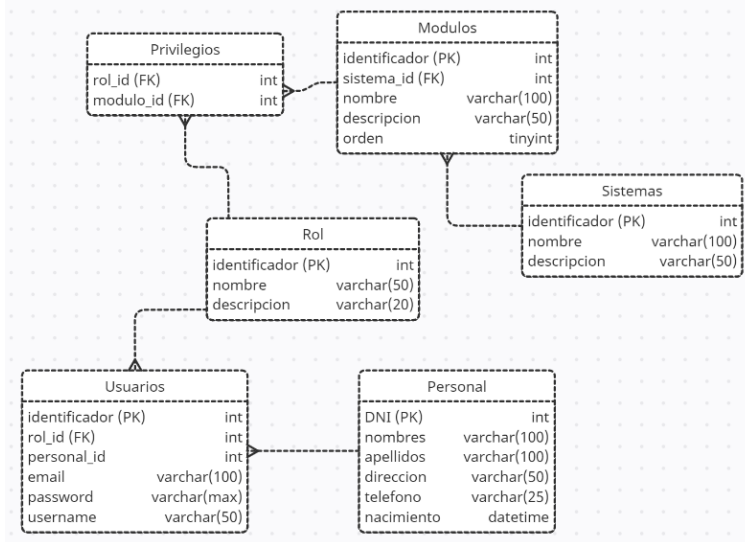
Nota 60: Creación propia



Semana 03, 04 a la fecha 27/06/2019

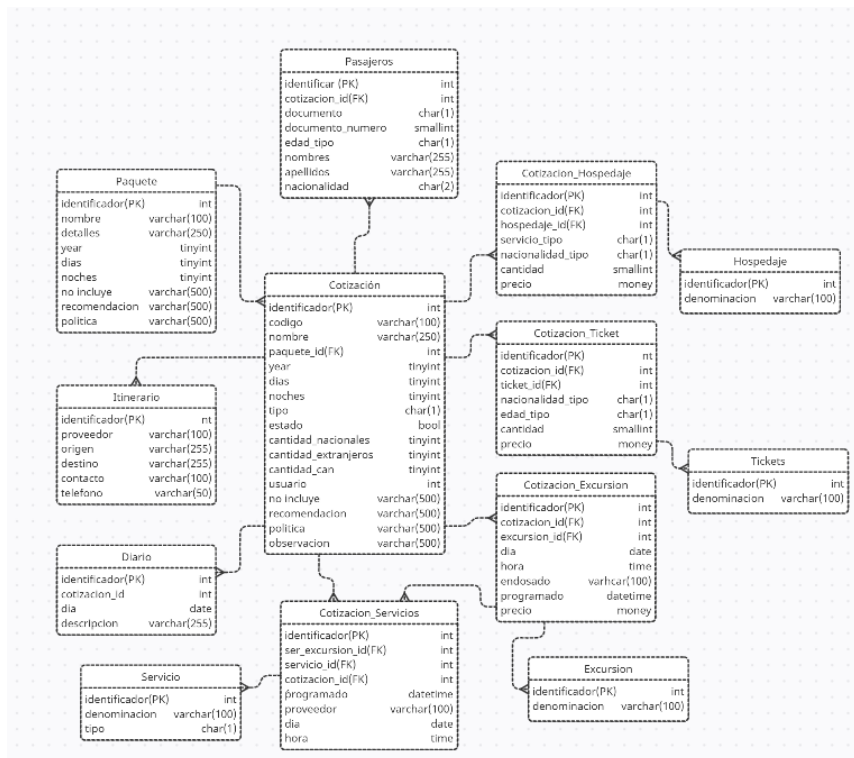
Se muestra la historia de usuario “Creación de la Base de datos” se encuentra finalizada.

Figura 24: Creación de la Base de datos (Proceso de accesos)



Nota 61: Creación propia

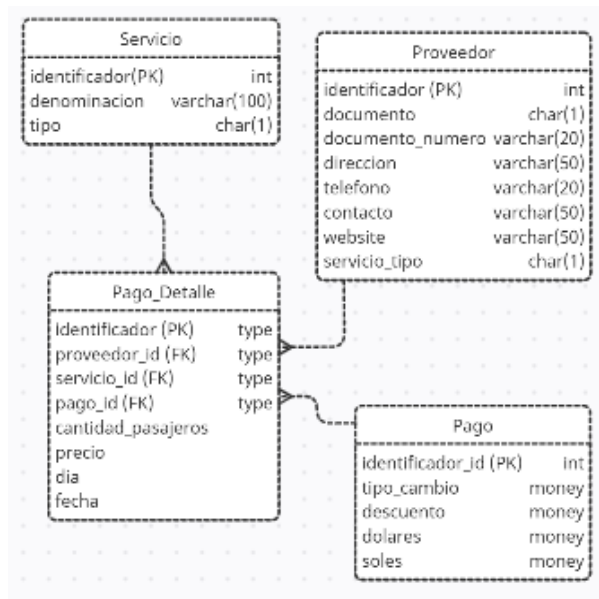
Figura 25: Creación de la Base de datos (Proceso de cotización)



Nota 62: Creación propia



Figura 26: Creación de la Base de datos (Proceso de asignación y pago)



Nota 61: Creación propia

Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 01 de la Semana 03, 04 teniendo como historia de usuario “Creación de la Base de datos” se encuentra finalizada y el “Acceso al sistema (Login)” el cual se puso en curso.

Tabla 42: Taskboard semana 03, 04

	Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5
Semana 03, 04	Fin: 27-06-2019	
	Historia de Usuario	Pendiente En curso Hecho
Sprint Nro. 01	• Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW)	X
	• Integración de la plantilla PAGE con el framework Laravel	X
	• Creación de la Base de datos	X



	<ul style="list-style-type: none"> • Acceso al sistema (Login) 		X
Sprint Nro. 02	<ul style="list-style-type: none"> • Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 	X	
	<ul style="list-style-type: none"> • Mantenimiento de Usuarios (Creación, Edición, Eliminación). 	X	
	<ul style="list-style-type: none"> • Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 	X	
	<ul style="list-style-type: none"> • Mantenimiento de Servicios y Paquetes 	X	
Sprint Nro. 03	<ul style="list-style-type: none"> • Administración de las áreas y Staff 	X	
	<ul style="list-style-type: none"> • Cotizar Servicios. 	X	
	<ul style="list-style-type: none"> • Creación de Reportes y Control 	X	
	<ul style="list-style-type: none"> • Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto). 	X	



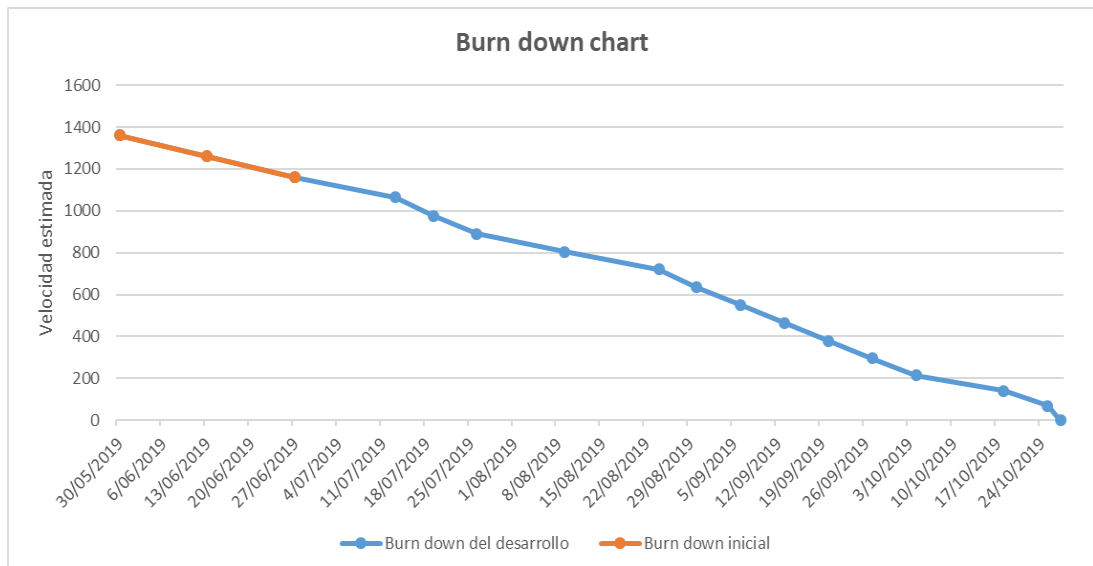
Sprint Nro. 04

- Integración de la plantilla PAGE, JavaScript con el framework Laravel X
- Actualización de la Base de datos del Sistema X
- Creación de un formulario de registro del cliente. X

Nota 62: Creación propia

La siguiente imagen presenta el adelanto de la semana N° 03,04, nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo".

Figura 27: Gráfico de Trabajo (BurnDown Chart). Semana N° 03, 04



Nota 63: Creación propia



Tabla 43: Tabla funcional semana 03, 04

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 02	VERSION DE EJECUCION	PF-MBD01
		FECHA EJECUCION	27/06/2019
TAREA:	<ul style="list-style-type: none"> Creación de la Base de datos. 	MODULO DEL SISTEMA	MBD

Descripción del caso de prueba: El siguiente paso es efectuar pruebas con relación a la carga de datos, la validación de los campos de almacenamiento y las relaciones existentes en la BD.

1. CASO DE PRUEBA

- a. Precondiciones
 - Acceso a la base de datos
 - Datos pre cargados.
- b. Pasos de la prueba
 - Registro de datos individual por tablas
 - Ejecución de Select simples y masivos según la base de datos existente.
 - Verificar que las relaciones en la base de datos sean las correctas.

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los módulos mantienen un estándar.	X		Se validó satisfactoriamente.	X	
Los módulos son accesibles y amigables	X		Se validó satisfactoriamente.	X	
El administrador presenta las etiquetas de integración.	X		Se validó que existen las etiquetas de PAGE y las del Framework Laravel	X	

- c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA



Defectos y desviaciones	Veredicto	
Observaciones:	X PASO	FALLO
	Probador	
	Firma:	
	Nombre:	
	Fecha:	

Nota 64: Creación propia

Semana 05 a la fecha 13/07/2019

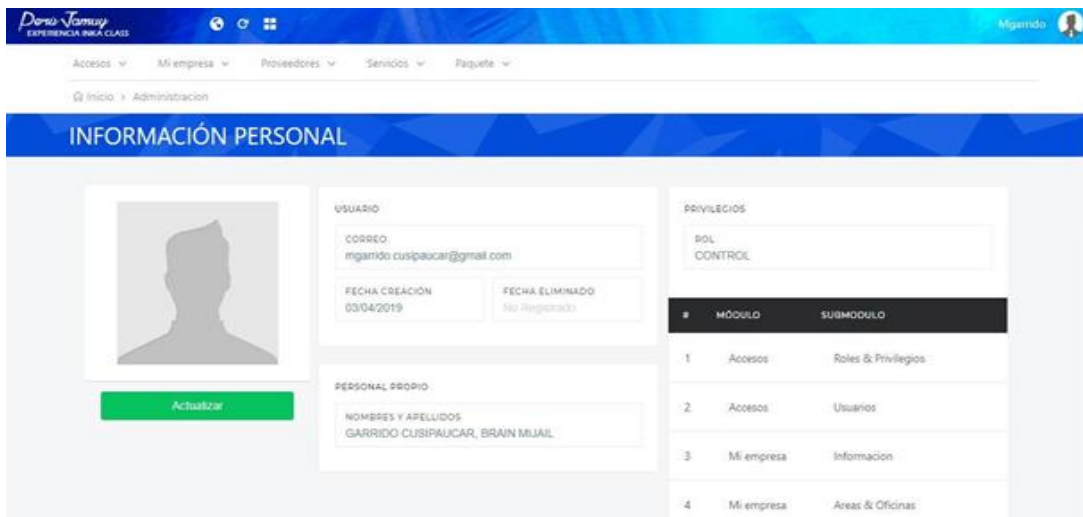
Se muestra la historia de usuario “Acceso al sistema (Login)” se encuentra finalizada.

Figura 28: Acceso al sistema (Login)



Nota 65: Creación propia

Figura 29: Credenciales de Acceso al sistema (Login)



Nota 66: Creación propia



Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 01 de la Semana 5 teniendo como historia de usuario “Acceso al sistema (Login)” se encuentra finalizada y la “Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores)” se puso en marcha.

Tabla 44: Taskboard semana 05

Semana 05		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
		Fin: 13-07-2019	Pendiente	En curso	Hecho
Sprint Nro. 01	Historia de Usuario				
	• Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW)				X
	• Integración de la plantilla PAGE con el framework Laravel				X
	• Creación de la Base de datos				X
	• Acceso al sistema (Login)				X
Sprint Nro. 02	• Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores).			X	
	• Mantenimiento de Usuarios (Creación, Edición, Eliminación).		X		



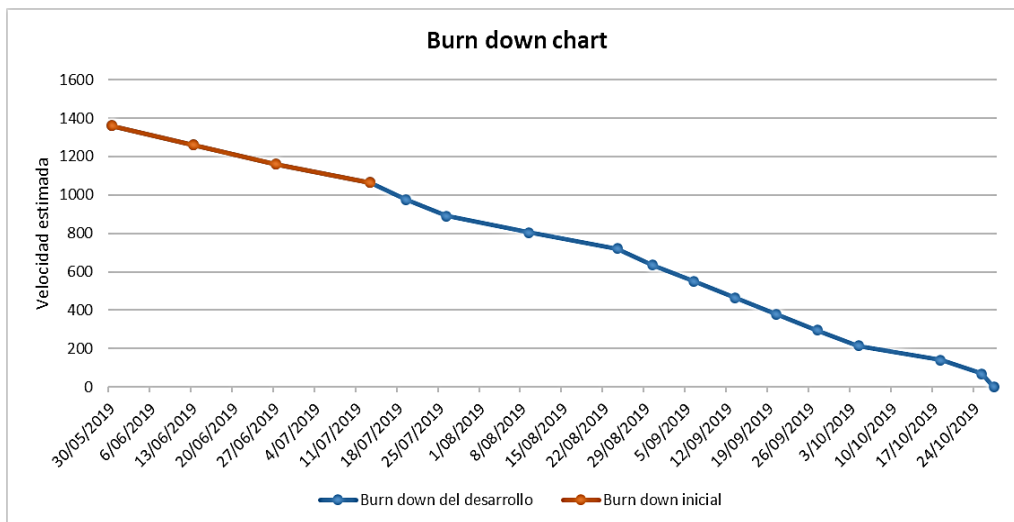
Sprint Nro. 03	• Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)	X
	• Mantenimiento de Servicios y Paquetes	X
	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X
	• Actualización de la Base de datos del Sistema	X
	• Creación de un formulario de registro del cliente.	X

Nota 62: Creación propia

La siguiente imagen presenta el adelanto de la semana N°5, nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo".



Figura 30: Gráfico de Trabajo (BurnDown Chart). Semana N° 05



Nota 63: Creación propia

Tabla 45: Tabla funcional semana 05

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 03	VERSION DE EJECUCION	PF- ML01
		FECHA EJECUCION	13/07/2019
TAREA:	<ul style="list-style-type: none"> Acceso al sistema (Login) 	MODULO DEL SISTEMA	ML

Descripción del caso de prueba: El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados y los mensajes de respuesta que muestra.

1. CASO DE PRUEBA

a. Precondiciones

- Ninguna

b. Pasos de la prueba

- Ingresar datos no válidos para validar campos
- Validar que el acceso funcione.



RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Bienvenido al sistema Usuario o Contraseña Incorrectos	X		Carga Satisfactoria Mostrar la consulta solicitada	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
Observaciones:	X PASO	FALLO
	Probador	
	Firma:	
	Nombre:	
	Fecha:	

Nota 64: Creación propia

Semana 06, 07 a la fecha 26/07/2019

Se muestra la historia de usuario “Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores)” se encuentra finalizada.

Figura 31: Administración de operadores (clientes directos)

RAZON SOCIAL	RUC	TITULAR	TELEFONO	CORREO	DIRECCION
SUNSHINE INTERNATIONAL S.A.C.	2055587671	BAI FEI	98232111111	gerencia@sunshine.com	CALLAS GARZAS NORTE NRO. 316 URB. LIMATAMBO LIMA - LIMA - SAN ISIDRO
QUALITY VACATIONS / LYMA SERVICIOS TURISTICOS S. A. C.	20537605031	PAZ CALLE MARIELA ADALGISA	95414555	gerencia@qualityvacations.com	AV. JOSE LARCO NRO. 812 INT. 105 (ESQUINA CON SAN MARTIN) LIMA - LIMA - MIRAFLORES
PERU INKASICO SAC - SERVICIOS TURISTICOS MULTIPLES	20517613593	VICTOR ORLANDO SALDAÑA HUJAMANGAYO	952411245	gerencia@peruinkasico.com	CALENRIQUE PALACIOS NRO. 360 DPTO. 512
DIKA AVENTURE TRAVEL EIRL	20492438473	MIRANDA VELASQUEZ CARLOS ANTONIO	447 7812 / 243 6443	info@dikatraveltravel	CALGRIMALDO DEL SOLAR NRO. 236 INT. 101 LIMA - LIMA - MIRAFLORES
PJ LUXURY	2060050992	JOSE MIGUEL ARONI LOPE	986064212	gerencia@perujamytavel.com	AV. EL SOL 948 OF. 302

Nota 65: Creación propia



Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 02 de la Semana 06, 07 teniendo como historia de usuario “Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores)” se encuentra finalizada. Y las historias de usuario como Mantenimiento de Usuarios (Creación, Edición, Eliminación), Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación), Mantenimiento de Servicios y Paquetes, se pusieron en marcha.

Tabla 46: Taskboard semana 06, 07

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 06,07		Fin: 26-07-2019			
		Historia de Usuario	Pendiente	En curso	Hecho
Sprint Nro. 01		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos 			X
		<ul style="list-style-type: none"> Acceso al sistema (Login) 			X
		<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X
Sprint Nro. 02					



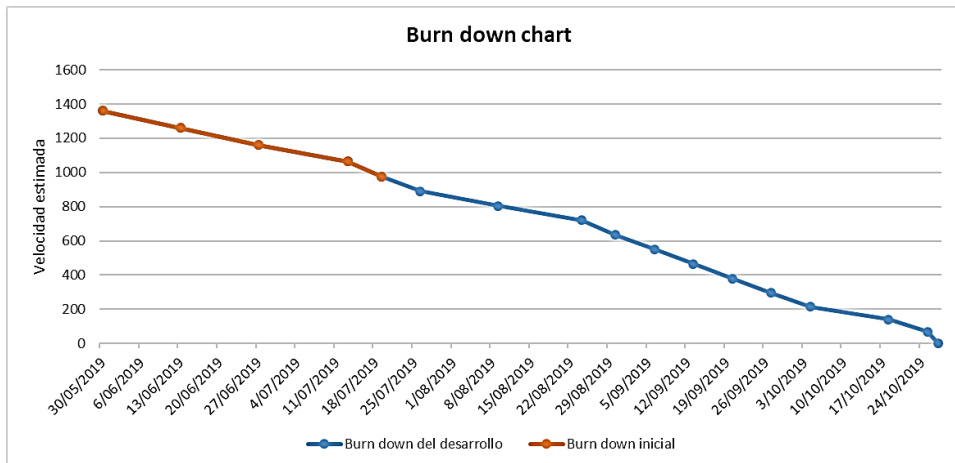
Sprint Nro. 03	<ul style="list-style-type: none"> • Mantenimiento de Usuarios (Creación, Edición, Eliminación). 	X
	<ul style="list-style-type: none"> • Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 	X
	<ul style="list-style-type: none"> • Mantenimiento de Servicios y Paquetes 	X
	<ul style="list-style-type: none"> • Administración de las áreas y Staff 	X
	<ul style="list-style-type: none"> • Cotizar Servicios. 	X
	<ul style="list-style-type: none"> • Creación de Reportes y Control 	X
	<ul style="list-style-type: none"> • Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto). 	X
Sprint Nro. 04	<ul style="list-style-type: none"> • Integración de la plantilla PAGE, JavaScript con el framework Laravel 	X
	<ul style="list-style-type: none"> • Actualización de la Base de datos del Sistema 	X
	<ul style="list-style-type: none"> • Creación de un formulario de registro del cliente. 	X

Nota 62: Creación propia

La siguiente imagen presenta el adelanto de la semana N°5, nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo".



Figura 32: Gráfico de Trabajo (BurnDown Chart). Semana N° 06, 07



Nota 63: Creación propia

Tabla 47: Tabla funcional semana 06,07

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 04	VERSION DE EJECUCION	PF- MI02
		FECHA EJECUCION	26/07/2019

TAREA:	<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 	MODULO DEL SISTEMA	ML
--------	--	--------------------	----

Descripción del caso de prueba: El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, duplicidad de los clientes directos o empresas, editar empresa y eliminar empresa.



1. CASO DE PRUEBA

a. Precondiciones

- Cliente directo o empresa existente en la base de datos

b. Pasos de la prueba

- Validación de los objetos en el Formulario de registro de los clientes directos o empresas.
- Validar que se puedan editar los objetos en el Formulario de los clientes directos o empresas ya existentes.
- Validar que sea posible eliminar un objeto al tener un cliente directo o empresa seleccionada

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos	X		Muestra mensaje con valores no son permitidos.	X	
La empresa ya se encuentra registrada	X		La empresa ya existe.	X	
Los datos han sido actualizados	X		Los datos han sido actualizados.		
La empresa ha sido eliminada.	X		Se cambia de estado al cliente directo o empresa.		

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

Nota 64: Creación propia



Semana 08, 09 a la fecha 24/08/2019

Se muestra la historia de usuario “Mantenimiento de Usuarios” (Creación, Edición, Eliminación).

Figura 33: Administración de usuarios

PERSONAL / EMPRESA	TIPO	EMAIL	ROL	CREADO	ESTADO
GARRIDO CUSIPAUCAR, BRAIN MIJAL USUARIO: GARRIDO	PERSONAL PROPIO	mgarrido.cuspauca@gmail.com	CONTROL	2019-04-03	[Edit] [Delete]
SAO PAULO BRASIL, SIMONE USUARIO: SIMENA...	PERSONAL PROPIO	jimena@gmail.com	DEMO	2019-05-17	[Edit] [Delete]
ARONI LOPE, JESUS MANUEL USUARIO: JESUS MANUEL A.	PERSONAL PROPIO	jesus.aroni@gmail.com	CONTROL	2019-05-23	[Edit] [Delete]
BLAS ALFARO, VERONICA USUARIO: VERONICA	PERSONAL PROPIO	veronica@perujamuytravel.com	ADMINISTRACION	2019-06-12	[Edit] [Delete]
GARRIDO CONDOIRL, MELISA USUARIO: MELISA	PERSONAL PROPIO	melissa@perujamuy.com.pe	PAQUETES	2019-06-17	[Edit] [Delete]
JUAN DE DIOS CONDOIRL, EMILY USUARIO: EMILY JUAN DE DIOS	PERSONAL PROPIO	econdoir@perujamuy.com	DEMO	2019-06-20	[Edit] [Delete]
SIMONE CRISTINA, DAVID USUARIO: SIMONE CRISTINA DAVID	PERSONAL MAYORISTA	simone@sanpaulviagens.com.br	DEMO	2019-08-20	[Edit] [Delete]

Nota 65: Creación propia

Se muestra la historia de usuario “Mantenimiento de Roles” para el control adecuado de la sección asignada (Creación, Edición, Eliminación)

Figura 34: Administración de Roles

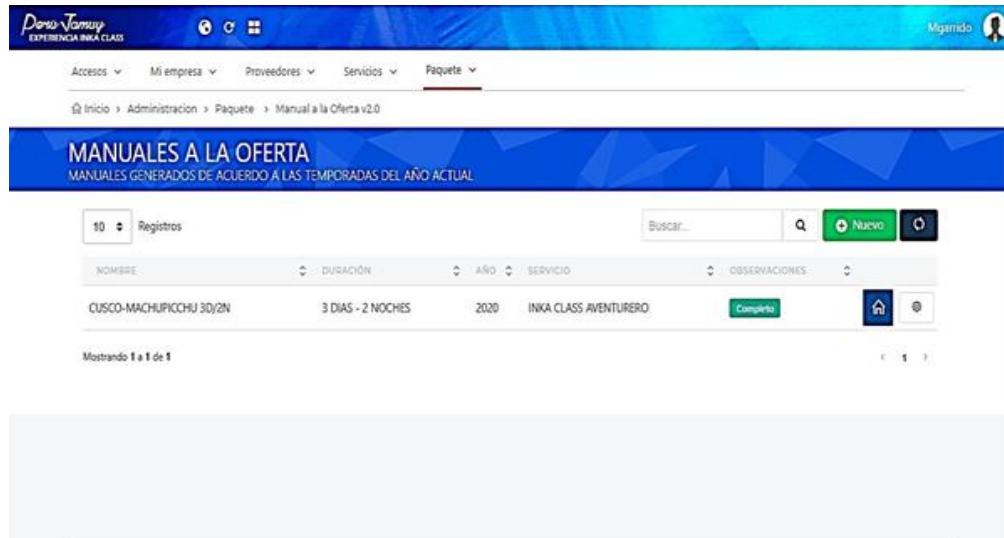
NOMBRE	DESCRIPCION	USUARIOS	PRIV.	ESTADO
CONTROL	Administra todo el sistema en general	5	51	[Edit] [Delete]
DEMO	demo	7	6	[Edit] [Delete]
ADMINISTRACION	ADMINISTRAR	3	34	[Edit] [Delete]
PROGRAMACION DE BOLETOS	PROGRAMACIÓN DE BOLETOS	1	8	[Edit] [Delete]
PAQUETES	PAQUETES	3	9	[Edit] [Delete]
OPERACIONES	OPERACIONES	2	10	[Edit] [Delete]
PROVEEDOR	PROVEEDORES DE SERVICIO (TRANSPORTE, ALIMENTOS, GUIADO Y ENDOSE)	7	1	[Edit] [Delete]
MONITOREO	MONITOREO	1	1	[Edit] [Delete]

Nota 66: Creación propia

Se muestra la historia de usuario “Mantenimiento de Servicios y Paquetes”.



Figura 35: Administración de Paquetes turísticos



Nota 67: Creación propia

Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 02 de la Semana 08, 09 teniendo como historias de usuario “Mantenimiento de Usuarios” (Creación, Edición, Eliminación), “Mantenimiento de Roles” para el control adecuado de la sección asignada (Creación, Edición, Eliminación), y “Mantenimiento de Servicios y Paquetes” se encuentran finalizadas. Se puso en marcha la historia de usuario Información de la Empresa, áreas, Staff, y Mantenimiento de Cotizador y Administrador.

Tabla 48: Taskboard semana 08, 09

	Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5
Semana 08, 09	Fin: 24-08-2019	
	Historia de Usuario	Pendiente En curso Hecho
Sprint Nro. 01	<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 	X



Sprint Nro. 02	<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
	<ul style="list-style-type: none"> Creación de la Base de datos 			X
	<ul style="list-style-type: none"> Acceso al sistema (Login) 			X
	<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X
	<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 			X
	<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 			X
Sprint Nro. 03	<ul style="list-style-type: none"> Mantenimiento de Servicios y Paquetes 			X
	<ul style="list-style-type: none"> Administración de las áreas y Staff 		X	
	<ul style="list-style-type: none"> Cotizar Servicios. 		X	
	<ul style="list-style-type: none"> Creación de Reportes y Control 	X		
	<ul style="list-style-type: none"> Creación del diseño de Interfaz Front-end del 	X		



Sprint Nro. 04

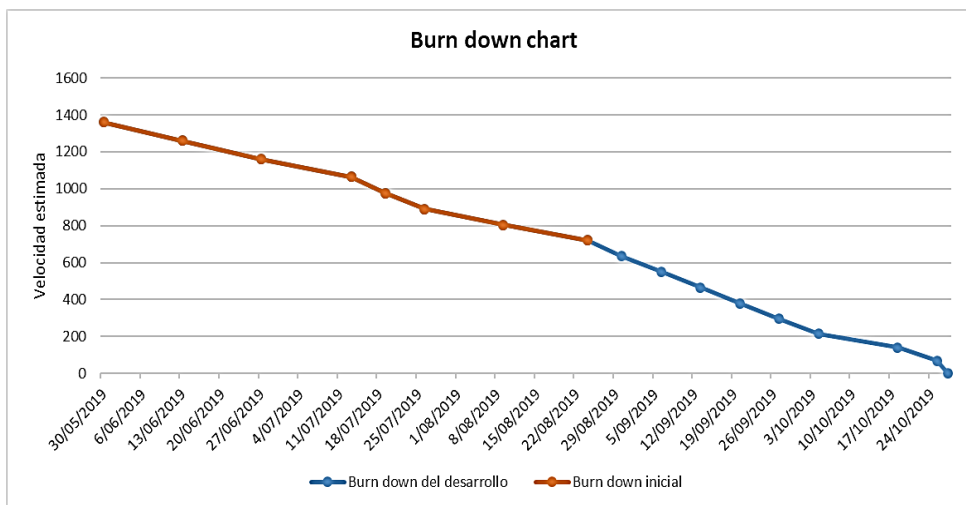
sistema de información Web (cliente indirecto).

- Integración de la plantilla PAGE, JavaScript con el framework Laravel X
- Actualización de la Base de datos del Sistema X
- Creación de un formulario de registro del cliente. X

Nota 68: Creación propia

La siguiente imagen presenta el adelanto de la semana N° 08, 09, se puede observar que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo" estimado observando que los tiempos comprometidos para el desarrollo actividades sean lo que uno espera.

Figura 36: Gráfico de Trabajo (BurnDown Chart). Semana N° 08,09



Nota 69: Creación propia



Tabla 49: Tabla funcional semana 08, 09

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 05	VERSION DE EJECUCION	PF- ML02
		FECHA EJECUCION	28/08/2019

TAREA:

- Mantenimiento de Usuarios (Creación, Edición, Eliminación).
- Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación)
- Mantenimiento de Servicios y Paquetes

MODULO DEL SISTEMA MA

Descripción del caso de prueba:

- El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, duplicidad de Usuarios, editar, eliminar Rol.
- El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, duplicidad de Roles, editar Rol y eliminar Rol.
- El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, duplicidad de Paquetes turísticos, editar, eliminar el paquete turístico.

1. CASO DE PRUEBA

a. Precondiciones



- Cliente directo o empresa existente en la base de datos.

b. Pasos de la prueba

- Validación de los objetos en el Formulario de registro de Roles, Usuarios y Registro de paquetes turísticos de la empresa.
- Validar que se puedan editar los objetos en el Formulario de Roles, Usuarios y paquetes turísticos ya existentes.
- Validar que sea posible eliminar un objeto al tener un Role, un usuario, y paquetes turísticos seleccionados.

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos.	X		Muestra mensaje con valores no son permitidos.	X	
El Rol, usuario y el paquete turístico ya se encuentra registrado	X		El rol, el usuario, el paquete turístico ya existe.	X	
Los datos han sido actualizados.	X		Los datos han sido actualizados.	X	
El Rol ha sido eliminado	X		Se cambia de estado al cliente directo o empresa	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

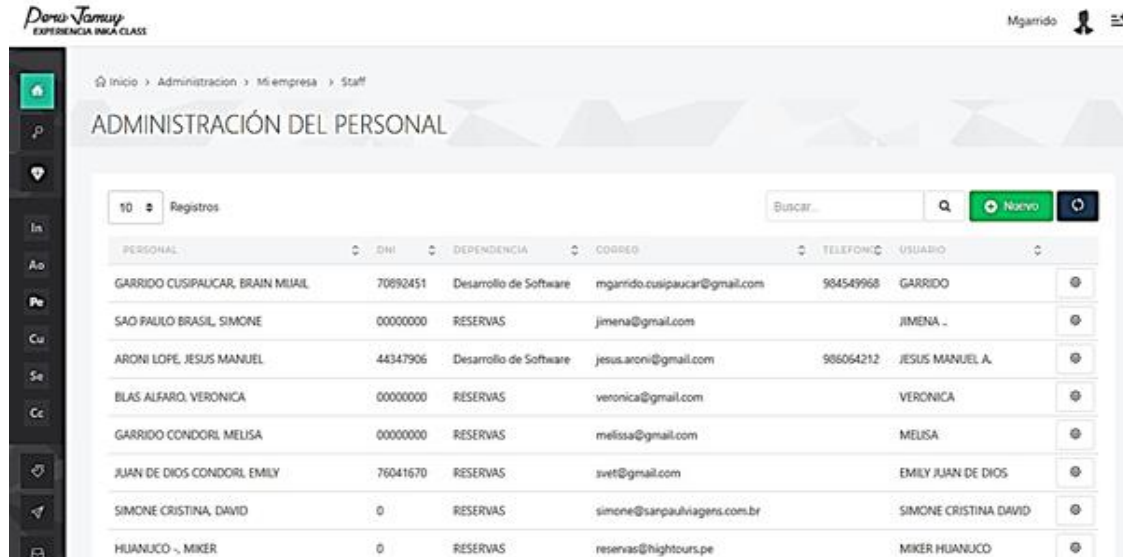
Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

Nota 70: Creación propia

Semana 10,11 a la fecha 20/09/2019

Se muestra la historia de usuario Administración de las áreas y staff.

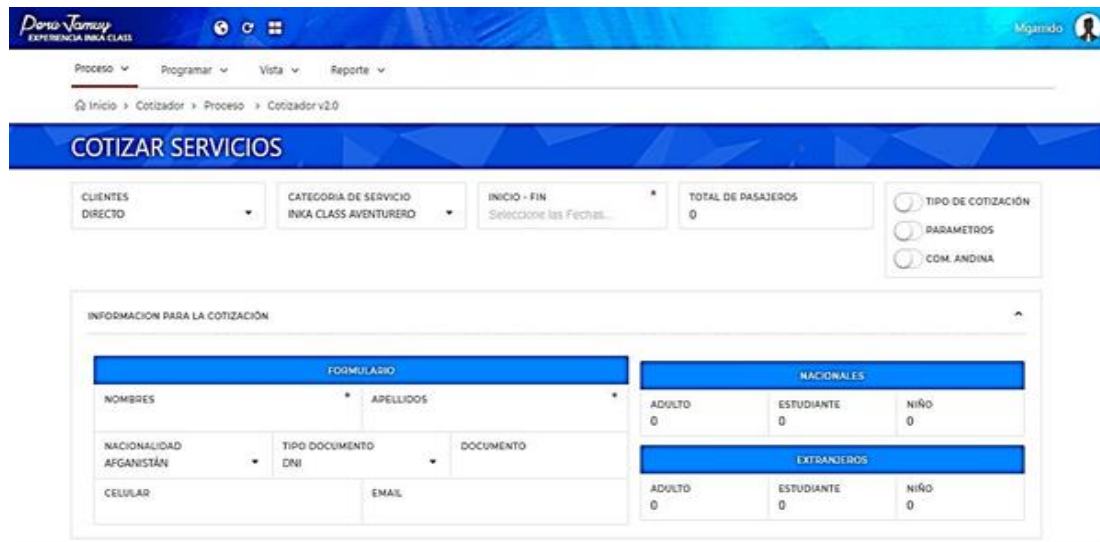
Figura 37: Administración de Personal Staff



Nota 71: Creación propia

Se muestra la historia de usuario Cotizar Servicios.

Figura 38: Cotizar servicios



Nota 72: Creación propia

Seguidamente se presenta el Scrum Taskboard que muestra el estado en que se encuentra el Sprint 03 de la Semana 10,11 teniendo como historia de usuario Administración de las áreas y Staff, “Cotizar Servicios”, se encuentra finalizada. Y se puso en marcha las



historias de usuario “Creación de Reportes y Control” y Creación del diseño de Interfaz Front-end del Sistema de Información Web (SIW).

Tabla 50: Taskboard semana 10, 11

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 10, 11		Fin: 20-09-2019			
		Historia de Usuario	Pendiente	En curso	Hecho
Sprint Nro. 01		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos 			X
		<ul style="list-style-type: none"> Acceso al sistema (Login) 			X
Sprint Nro. 02		<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X
		<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 			X
		<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección asignada 			X



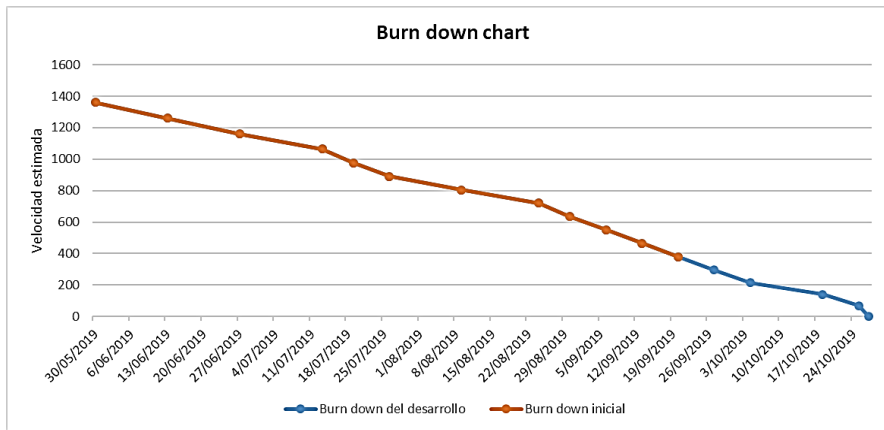
	(Creación, Edición, Eliminación)		
Sprint Nro. 03	• Mantenimiento de Servicios y Paquetes		X
	• Administración de las áreas y Staff		X
	• Cotizar Servicios.		X
	• Creación de Reportes y Control		X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).		X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X	
	• Actualización de la Base de datos del Sistema	X	
	• Creación de un formulario de registro del cliente.	X	

Nota 73: Creación propia

Seguidamente la figura muestra el adelanto de la Semana 10, 11, nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo" estimado, observando que los tiempos estimados para el desarrollo actividades sean los correctos.



Figura 39: Gráfico de Trabajo (BurnDown Chart). Semana N° 10, 11



Nota 74: Creación propia

Figura 40: Tabla funcional semana 10, 11

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 06	VERSION DE EJECUCION	PF- MA03
		FECHA	20/09/2019
TAREA:	<ul style="list-style-type: none"> Administración de las áreas y Staff Cotizar Servicios. 	MODULO DEL SISTEMA	MA

- Descripción del caso de prueba:**
- El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, de las áreas de la empresa y su personal
 - El siguiente paso es efectuar pruebas con relación a la validación de los campos cuando hay datos errados, duplicidad de la cotización de Paquetes turísticos, editar, eliminar el paquete turístico.

1. CASO DE PRUEBA

- Precondiciones



- Cliente directo o empresa existente en la base de datos.

b. Pasos de la prueba

- Validación de los objetos en el Formulario de registro de áreas, personal de la empresa.
- Validar que sea posible eliminar, editar un objeto, al tener un paquete turístico seleccionado.

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos	X		Muestra mensaje con valores no son permitidos.	X	
Los datos han sido actualizados.	X		Los datos han sido actualizados.	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

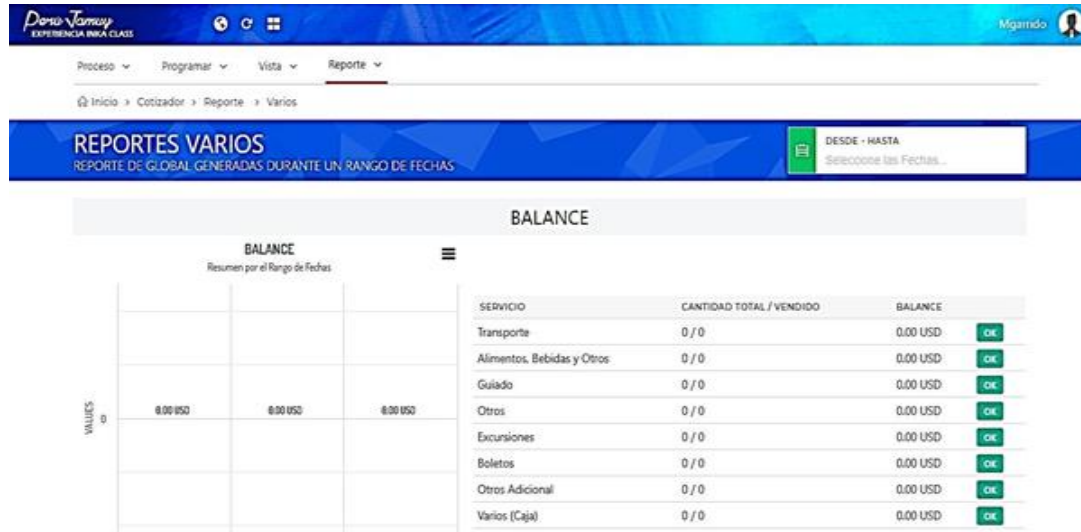
Nota 75: Creación propia



Semana 12, 13 a la fecha 04/10/2019

Seguidamente se muestra la historia de usuario Creación de Reportes y Control.

Figura 41: Reportes y control



Nota 76: Creación propia

Seguidamente se muestra la historia de usuario Creación del diseño de Interfaz Front-end del Sistema de Información Web (SIW) (cliente indirecto).

Figura 42: Front-end



Nota 77: Creación propia

Seguidamente, se presenta el Scrum Taskboard que muestra el estado en que se encuentra el Sprint 03 de la Semana 12,13 donde se concluyó las historias de usuario de



“Creación de Reportes y Control”, “Creación del diseño de Interfaz Front-end del Sistema de Información Web (SIW) (cliente indirecto). Además, se puso en marcha las historias de usuario “Integración de la plantilla PAGE, JavaScript con el framework Laravel” y “Actualización de la Base de datos del Sistema”

Tabla 51: Taskboard semana 12, 13

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 12, 13		Fin: 04-10-2019			
		Historia de Usuario	Pendiente	En curso	Hecho
Sprint Nro. 01		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos 			X
		<ul style="list-style-type: none"> Acceso al sistema (Login) 			X
Sprint Nro. 02		<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X
		<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 			X



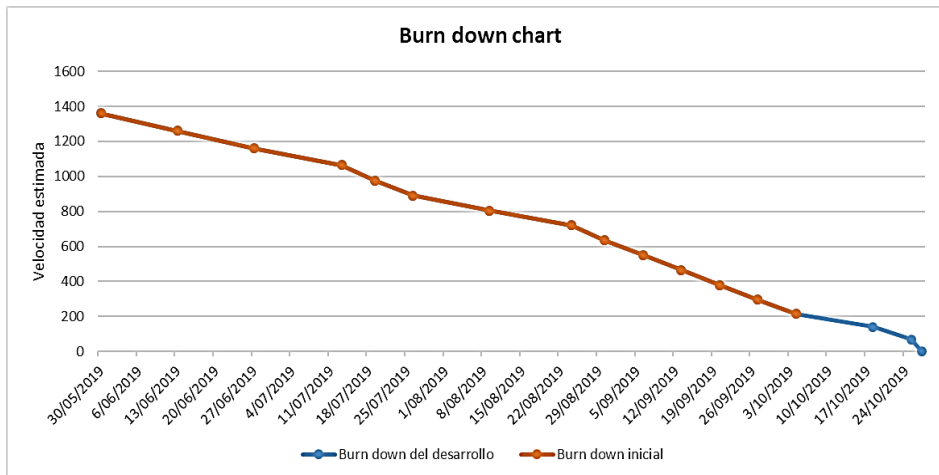
Sprint Nro. 03	<ul style="list-style-type: none"> • Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 	X
	<ul style="list-style-type: none"> • Mantenimiento de Servicios y Paquetes 	X
	<ul style="list-style-type: none"> • Administración de las áreas y Staff 	X
	<ul style="list-style-type: none"> • Cotizar Servicios. 	X
	<ul style="list-style-type: none"> • Creación de Reportes y Control 	X
	<ul style="list-style-type: none"> • Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto). 	X
Sprint Nro. 04	<ul style="list-style-type: none"> • Integración de la plantilla PAGE, JavaScript con el framework Laravel 	X
	<ul style="list-style-type: none"> • Actualización de la Base de datos del Sistema 	X
	<ul style="list-style-type: none"> • Creación de un formulario de registro del cliente. 	X

Nota 76: Creación propia

En la siguiente presentación se muestra el adelanto de la Semana Nro. 12 y 13, verificamos que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo" estimado, observando que los tiempos estimados para el desarrollo actividades sean los correctos.



Figura 43: Gráfico de Trabajo (BurnDown Chart). Semana N° 12,13



Nota 77: Creación propia

Tabla 52: Tabla funcional semana 12, 13

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 07	VERSION DE EJECUCION	PF- MA04
		FECHA	04/10/2019
		EJECUCION	

TAREA:	<ul style="list-style-type: none"> Creación de reportes de control Creación del diseño de interfaz Front-end del Sistema de Información Web (SIW) (cliente indirecto). 	MODULO DEL SISTEMA	MAP
--------	--	--------------------	-----

Descripción del caso de prueba:

- Lo siguiente es efectuar pruebas con relación a las funciones de las consultas de información solicitadas por el usuario



- El paso consecutivo es realizar pruebas con relación a las visitas que los usuarios realizan al Sistema de Información Web (SIW).

1. CASO DE PRUEBA

a. Precondiciones

- Requerimientos ingresados.

b. Pasos de la prueba

- Mostrar los requerimientos pendientes
- Mostrar la página principal para el usuario que visite al sistema información web.

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos	X		Muestra mensaje con valores no son permitidos.	X	
Los datos han sido actualizados.	X		Los datos han sido actualizados.	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

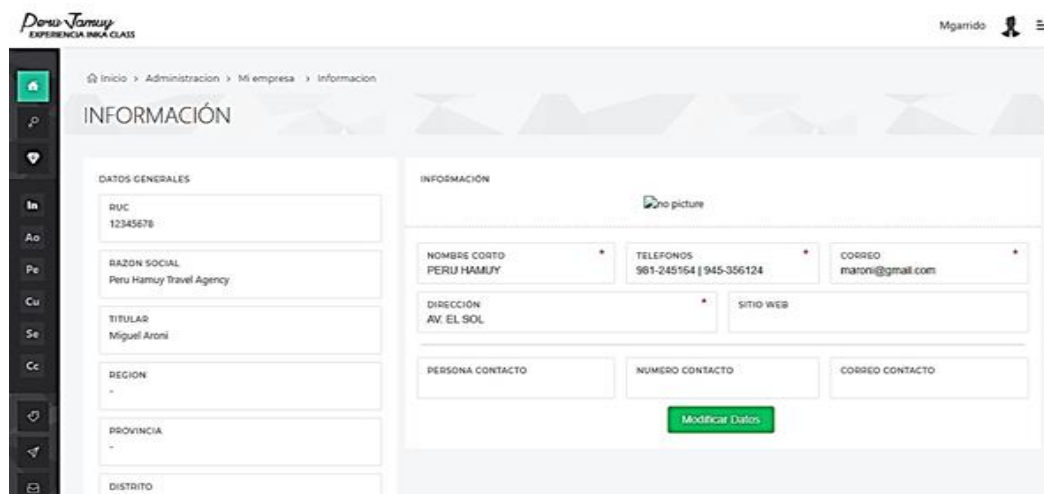
Nota 75: Creación propia



Semana 14,15 a la fecha 25/10/2019

Se presenta la historia de usuario integración de la plantilla PAGE, JavaScript con el framework Laravel.

Figura 44: Integración de la plantilla



Nota 76: Creación propia

Se presenta la historia de usuario. Actualización de la Base de Datos del Sistema.

Figura 45: Categorización de cuentas actualizar BD



Nota 77: Creación propia

Se presenta el Scrum Taskboard que muestra el estado en que se encuentra el Sprint 04 de la Semana 14,15 donde se concluyó las historias de usuario de “Integración de la plantilla PAGE con el framework Laravel” y “Actualización de la Base de datos del



Sistema. Además, se inició las historias de usuario “Creación de un formulario de registro del cliente.

Tabla 53: Taskboard semana 14, 15

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 14, 15		Fin: 25-10-2019	Pendiente	En curso	Hecho
		Historia de Usuario			
Sprint Nro. 01		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos 			X
		<ul style="list-style-type: none"> Acceso al sistema (Login) 			X
Sprint Nro. 02		<ul style="list-style-type: none"> Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X
		<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 			X
		<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección asignada 			X



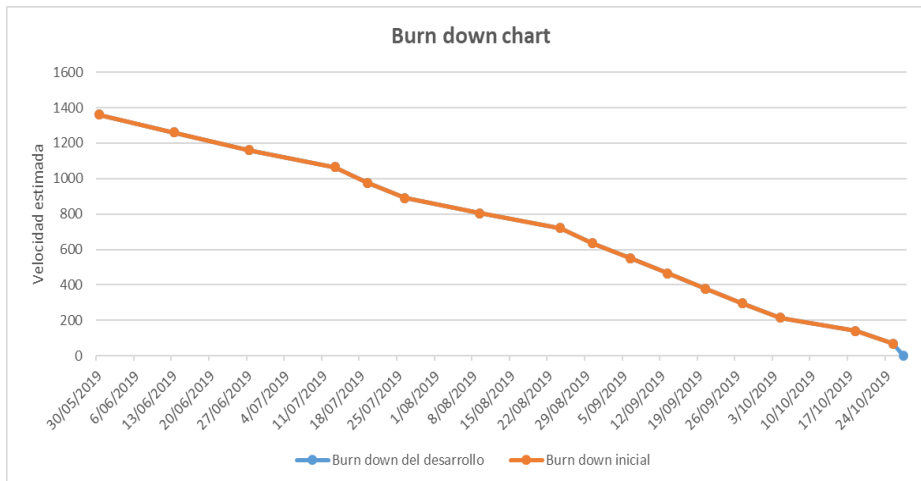
	(Creación, Edición, Eliminación)	
Sprint Nro. 03	• Mantenimiento de Servicios y Paquetes	X
	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X
	• Actualización de la Base de datos del Sistema	X
	• Creación de un formulario de registro del cliente.	X

Nota 76: Creación propia

La siguiente Figura muestra el avance de la Semana Nro. 14 y 15, visualizamos las actividades realizadas en el "BurnDown" se aproxima al grafico esperado "BurnDown del desarrollo" estimado, observando que los tiempos comprometidos para el desarrollo actividades sean lo que uno espera.



Figura 46: Gráfico de Trabajo (BurnDown Chart). Semana N° 14,15



Nota 77: Creación propia

Tabla 54: Tabla funcional semana 14, 15

PRUEBA FUNCIONAL

PRUEBA N°.	Prueba de funcionalidad N° 08	VERSION DE EJECUCION	PF- MBD03
		FECHA EJECUCION	25/10/2019
TAREA:	<ul style="list-style-type: none"> Integración de la plantilla PAGE, JavaScript con el framework Laravel Actualización de la Base de datos del Sistema 	MODULO DEL SISTEMA	MBD
Descripción del caso de prueba:	<ul style="list-style-type: none"> El siguiente paso efectúa pruebas con relación al diseño del Administrador e integración con PAGE y Laravel. 		



- Actualización de la información de las cuentas y objetos de la base de datos del sistema información web.

1. CASO DE PRUEBA

a. Precondiciones

- Requerimientos establecidos, datos ingresados.

b. Pasos de la prueba

- Mostrar los requerimientos establecidos
- Mostrar la página principal para el usuario que visite al sistema información web

RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos	X		Muestra mensaje con valores no son permitidos.	X	
Los datos han sido actualizados.	X		Los datos han sido actualizados.	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

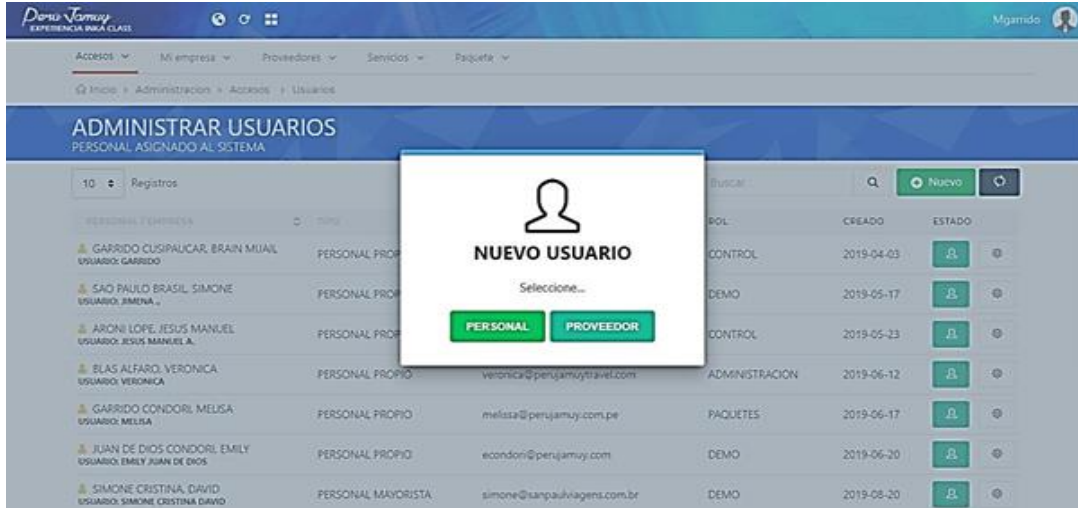
Nota 78: Creación propia



Semana 16 a la fecha 27/10/2019

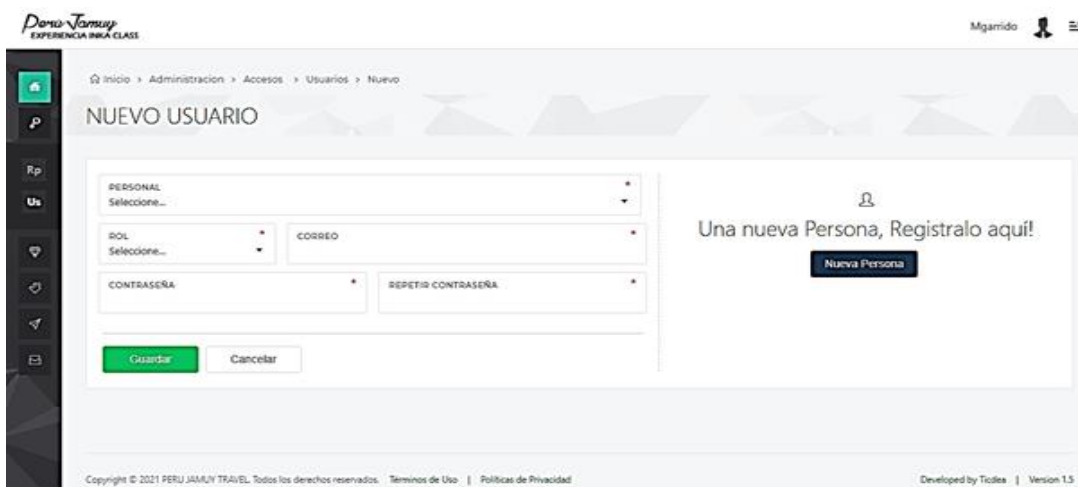
Se muestra la historia de usuario “Creación de un formulario de registro del cliente”

Figura 47: Ingreso al registro del cliente



Nota 79: Creación propia

Figura 48: Registrar al cliente



Nota 80: Creación propia

Seguidamente, se muestra el Scrum Taskboard que visualiza el estado del Sprint 04 de la Semana 16 donde se concluyó la historia de usuario de “Creación de un formulario de registro del cliente”.



Tabla 55: Taskboard semana 14, 15

		Inicio: 30-05-2019	Nombre: Desarrollo del Sistema de Información Web (SIW) Turismo Smart Versión 2.5		
Semana 16		Fin: 27-10-2019	Pendiente	En curso	Hecho
Sprint Nro. 01	Historia de Usuario				
		<ul style="list-style-type: none"> Creación del diseño de Interfaz del Back-end del Sistema de Información Web (SIW) 			X
		<ul style="list-style-type: none"> Integración de la plantilla PAGE con el framework Laravel 			X
		<ul style="list-style-type: none"> Creación de la Base de datos Acceso al sistema (Login) Creación de una página con un formulario para el registro de los clientes directos que pertenecen al rubro turismos y que requieren del servicio (Administración de operadores). 			X X X
Sprint Nro. 02		<ul style="list-style-type: none"> Mantenimiento de Usuarios (Creación, Edición, Eliminación). 			X
		<ul style="list-style-type: none"> Mantenimiento de Roles para el control adecuado de la sección asignada (Creación, Edición, Eliminación) 			X



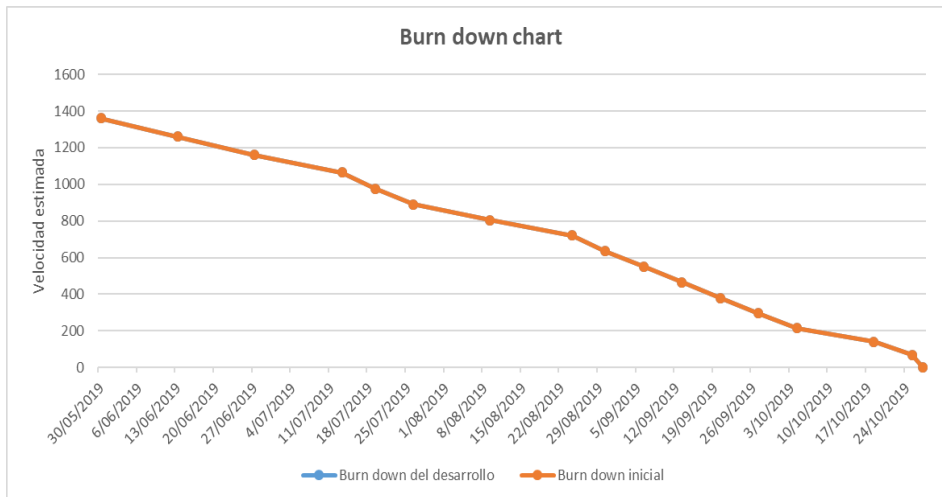
Sprint Nro. 03	• Mantenimiento de Servicios y Paquetes	X
	• Administración de las áreas y Staff	X
	• Cotizar Servicios.	X
	• Creación de Reportes y Control	X
	• Creación del diseño de Interfaz Front-end del sistema de información Web (cliente indirecto).	X
Sprint Nro. 04	• Integración de la plantilla PAGE, JavaScript con el framework Laravel	X
	• Actualización de la Base de datos del Sistema	X
	• Creación de un formulario de registro del cliente.	X

Nota 81: Creación propia

La siguiente imagen presenta el adelanto de la semana Nro.16, nos podemos dar cuenta que las actividades realizadas en el "BurnDown" se aproxime al grafico esperado "BurnDown del desarrollo" estimado observando que los tiempos comprometidos para el desarrollo actividades sean lo que uno espera.



Figura 49: Gráfico de Trabajo (BurnDown Chart). Semana N° 16



Nota 82: Creación propia

Tabla 56: Tabla funcional semana 16

PRUEBA FUNCIONAL

PRUEBA N°. Prueba de funcionalidad N° 09 VERSION DE EJECUCION PF- MBD04
FECHA 27/10/2019
EJECUCION

TAREA: • Creación de un MODULO DEL MBD formulario para el SISTEMA registro del cliente.

Descripción del caso de prueba: El siguiente paso es efectuar pruebas con relación al registro de un usuario nuevo o editar uno ya existente.

1. CASO DE PRUEBA

a. Precondiciones

- Repositorio o base de datos activa

b. Pasos de la prueba

- Realizar el registro de un nuevo usuario
- Mostrar la página con un nuevo usuario registrado



RESPUESTA ESPERADA DE LA APLICACIÓN	COINCIDE		RESPUESTA DEL SISTEMA	COINCIDE	
	SI	NO		SI	NO
Los valores ingresados no son permitidos	X		Muestra mensaje con valores no son permitidos.	X	
Los datos han sido actualizados.	X		Los datos han sido actualizados.	X	

c. Post condiciones

Si aplica

2. RESULTADOS DE LA PRUEBA

Defectos y desviaciones	Veredicto	
	X PASO	FALLO
Observaciones:	Probador	
	Firma:	
	Nombre:	
	Fecha:	

Nota 83: Creación propia

3.13. Arquitectura e implementación

En el desarrollo de la arquitectura e implementación Sistema de Información Web (SIW) Turismo Smart Versión 2.5 se utilizó el estilo de desarrollo Modelo-vista-controlador (MVC). El autor que hemos citado como referencia ha afirmado lo siguiente, “Modelo-vista-controlador (MVC) sus siglas en inglés, model-view-controller (modelo-vista-controlador) - es un patrón arquitectónico que nos permite desarrollar aplicaciones, manteniendo separada la lógica de negocios de las vistas, utilizando un “controlador” como conector (o intermediario) entre ambas” (Bahit, 2012, pág. 7) La arquitectura que se hace mención nos habla de capas y estas son 3:

3.13.1. Capa Presentación

- Aquí se lee toda la información que se obtiene en el formulario del cliente y esta es enviada a una ruta dentro del mismo proyecto es decir una capa proceso.
- Recepción de toda la información recibida por la capa proceso para poder alimentar a una vista la cual es mostrada a un usuario final.



- Aquí se almacena todas las vistas que contiene el proyecto.
- El usuario final tiene los permisos para la visualización de las vistas destinadas a su proceso en la presente capa.

3.13.2. Capa Proceso (Controlador)

- Recepción de datos de entrada enviadas desde la capa presentación.
- Después de un conjunto de procedimientos se envía datos a la capa modelo si esto es necesario para que esto los procede y retorne un nuevo valor si esto es necesario.
- Envía datos si esto es necesario a la capa presentación.

3.13.3. Capa Modelo

- Se realiza operaciones con la base de datos
- Se realiza los procesos transaccionales como grabar, editar y eliminar
- Retorna datos
- Da seguridad a la integridad de los datos



Capítulo IV: Resultados

Después del desarrollo, implementación y pruebas del Sistema de Información Web (SIW) Turismo Smart Versión 2.5, se puso de inmediato en producción ya que la empresa lo requería. Todo esto se aceleró gracias al uso una metodología de desarrollo fácil y ágil denominada Scrum el cual permitió alcanzar el objetivo planteado en la tesis de investigación aplicada. Además de ello se observó los siguientes resultados:

4.1. Comprobación de la prospectiva

- El desarrolló del Sistema de Información Web (SIW) Turismo Smart Versión 2.5 permite agilizar los procesos digitales de las áreas y staff de la empresa de turismo, sincronizando el funcionamiento a través de un solo Sistema de Información Web (SIW).
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 trabaja ininterrumpidamente en un dominio y hosting comprado por la empresa. Además, el acceso a este es a través del internet desde cualquier Host debidamente acreditado.
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 realiza la integración de información principalmente en el ámbito de cotizaciones de Paquetes turísticos y la reserva de servicios solicitados por los clientes directos de la empresa.
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5. ha sufrido diversas actualizaciones a través del tiempo según lo solicitado por dueño de la empresa. Estas versiones comenzaron en la versión 1.0 y ahora se encuentra en la versión 2.5.
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 es un producto comercial capaz de adaptarse al funcionamiento de cualquier operador turístico que quiera utilizarlo.

4.2. Cumplimiento de los objetivos

En el planteamiento de objetivo general de la tesis de investigación de tipo aplicada se propuso Desarrollar e implantar un sistema información web que permita gestionar



paquetes turísticos, integrando los procesos funcionales del operador turístico “PERU JAMUY”. Objetivo que se logró cumplir ya que se implanto en la empresa el Sistema de Información Web (SIW) Turismo Smart Versión 2.5 el cual está en funcionamiento.

Al concretar el objetivo general se pudo también lograr alcanzar los objetivos específicos ya que estos contribuyen directamente en el desarrollo del software tales como:

- Analizar los procesos funcionales que permitan gestionar paquetes turísticos.
- Diseñar e implementar la solución o software que involucre las siguientes funcionalidades: Módulo de gestión de paquetes turísticos, organización de procesos internos, clientes.
- Desplegar la solución en las instancias definidas por la organización para la empresa y el cliente.

4.3. Contribuciones (impacto)

- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 está alojada en un dominio privado y disponible las 24 horas del día para el uso del personal y staff de la empresa. Como está en un hosting web produce un servicio de información constante de todas las solicitudes de información de los procesos almacenados en el dominio a través de la consulta a una base de datos en tiempo real. Se puede acceder desde cualquier dispositivo electrónico como celular, Tablet, computadora, etc., proporcionando información comercial estratégica en cualquier momento y en cualquier lugar.
- La infraestructura del software fue desarrollada con una metodología ágil, donde se establecen procesos definidos, el cual permite la documentación del software en desarrollo, obteniendo un plano funcional a través de etapas en el tiempo y procesos de evaluación de módulos.
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 es funcional con cualquier navegador utilizado en un computador, Tablet, etc.



- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 ofrece módulos adaptables de crecimiento de acuerdo con la empresa de servicios turísticos donde se despliega
- El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 no necesita instalación, solo necesita un proceso de conexión adaptable al TCP/IP, en versión 4 o 6.
- El sistema El Sistema de Información Web (SIW) Turismo Smart Versión 2.5 cuenta con el mismo enfoque que el primer antecedente nacional, diferenciándose en la metodología utilizada y en la tecnología utilizada, y que se concluyó en la implementación del software en la entidad de turismo.
- Los antecedentes nacionales, como internacionales, son parte del Sistema de Información Web (SIW) Turismo Smart Versión 2.5, en la cual se tomaron las recomendaciones para la optimización de los procesos.



Glosario

Acceso: Operación de identificación de un dato, un registro o una entidad que pertenece a un fichero. Proceso necesario para acceder a una instrucción almacenada en memoria para ejecutarla.

Automatización: Sistema tecnológico basado en la ingeniería y la informática, que proporciona una optimización de los procesos productivos mediante la regulación automática (autorreguladores).

Bit: El bit es la unidad mínima de información empleada en informática, en cualquier dispositivo digital, o en la teoría de la información. Con él, podemos representar dos valores cualesquiera, como verdadero o falso, abierto o cerrado, blanco o negro, norte o sur, masculino o femenino, rojo o azul, etc. Basta con asignar uno de esos valores al estado de "apagado" (0), y el otro al estado de "encendido" (1).

Byte: Es una unidad de información digital equivalente a cuatro bits (binary digit, dígito binario) original y posteriormente como estándar se adoptó que 1 byte equivale a ocho bits. La palabra byte proviene de bite, que significa mordisco, como la cantidad más pequeña de datos que un ordenador podía "morder" a la vez. El símbolo de byte es un B mayúscula, para distinguir de bit, cuyo símbolo es b minúscula. El byte se utiliza generalmente en las áreas de informática y telecomunicaciones, en esta última se denomina comúnmente octeto, que proviene del francés octet, derivado del latín octo y del griego okto, que significa ocho, diferenciando así el byte de 8 bits de otros bytes con diferente equivalencia de bits.

Cloud Storage: El almacenamiento en la nube, del inglés cloud storage, es un modelo de almacenamiento de datos basado en redes de computadoras, ideado en los años 1960, donde los datos están alojados en espacios de almacenamiento virtualizados, por lo general aportados por terceros.

Embebido: Un sistema embebido o empotrado (integrado, incrustado) es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas, frecuentemente en un sistema de computación en tiempo real. Al contrario de lo que ocurre con los ordenadores de propósito general (como por ejemplo una computadora personal o PC) que están diseñados



para cubrir un amplio rango de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas.

Formulario: En Internet y sistemas informáticos, los formularios pueden ser llenados de forma online con validación de datos, y son muy útiles para las encuestas, registración de usuarios, ingreso a sistemas, suscripciones, etc. Los formularios por Internet son llamados formularios web, y generalmente son hechos a través de etiquetas HTML, aunque también existen otros medios como Flash, Java, etc.

IDE: Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Iteración: En programación, cuando el bloque de instrucciones de un bucle se ejecuta, se dice que se ha producido una iteración.

Jquery: Es una biblioteca multiplataforma de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

Módulo: Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

MVC: Es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Open source: Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

Product Owner: El rol del Product Owner, puede venir de parte del cliente o dentro de la empresa misma, depende. Generalmente no se aconseja que el Product Owner sea parte también



del equipo de desarrollo, o el Scrum Master mismo, sus intereses se pueden ver enfrentados, pero esto puede variar según el caso. Por defecto, probablemente venga de parte del cliente.

Product backlog: El Product Backlog es el conjunto de requisitos funcionales y no funcionales que debe cumplir el producto una vez entregado. No se requiere que esté completo al momento de su creación, basta con definir aquellos requisitos que se conozcan en su momento y alentar a su crecimiento continuo o su modificación.

Proceso: Es una mínima unidad de ejecución de una tarea determinada, siguiendo un conjunto de instrucciones que ha sido especificado, utilizando un recurso del sistema y partiendo de la base de un estado determinado, para lo cual se obtiene lógicamente un resultado afín al deseado.

Registro: En un fichero de bases de datos, cada uno de los elementos en que se divide. Contiene los datos de un elemento de los descritos en el fichero, dividiendo la información en campos. Generalmente es el contenido de un formulario completo del formato escogido para la recogida de datos.

Reporte: Son informes que organizan y exhiben la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios.

RUP: El Rational Unified Process o Proceso Unificado de Racional. Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo que es enfocada hacia “diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal.

Scrum: El Scrum es un proceso de la Metodología Ágil que se usa para minimizar los riesgos durante la realización de un proyecto, pero de manera colaborativa. Entre las ventajas se encuentran la productividad, calidad y que se realiza un seguimiento diario de los avances del proyecto, logrando que los integrantes estén unidos, comunicados y que el cliente vaya viendo los avances.



Script: Un script, archivo de órdenes, archivo de procesamiento por lotes o, cada vez más aceptado en círculos profesionales y académicos, guion es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guion. El uso habitual de los guiones es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los intérpretes de órdenes sean a la vez intérpretes de este tipo de programas.

Sprint: Sprint es el nombre que va a recibir cada uno de los ciclos o iteraciones que vamos a tener dentro de dentro de un proyecto Scrum.

SDK: (Software Development Kit - Kit de desarrollo de software o devkit). Un SDK es un conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software, estructura de software, plataforma de hardware, sistema de computadora, consulta de videojuego, sistema operativo o similar.

Servidor: En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos.

SGBD: (Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

Sprint: Es una división de un conjunto de tareas componen un proyecto en un periodo de tiempo. Normalmente los Sprints suelen ser de 15 a 30 días). En cada sprint se busca poder tener una parte del proyecto que sea “entregable” para poder ver la evolución del mismo hasta llegar al producto final.

Sistema de Información Web (SIW): Un sistema de información web tiene como principal objetivo la gestión, y administración de los datos e información que lo componen publicada en el internet por el protocolo HTTP o HTTPs.



Trazabilidad: Un conjunto de procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto, o lote de productos a lo largo de la cadena de suministros, en un momento dado y a través de unas herramientas determinadas

Webservice: Un servicio web (en inglés, web service o web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web.

Turismo: comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos al de su entorno habitual, por un período de tiempo consecutivo inferior a un año, con fines de ocio, por negocios y por otros motivos turísticos.

Aplicación Web: herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador.

Perú: País de Sudamérica que abarca una sección del bosque del Amazonas y Machu Picchu, una antigua ciudad inca en las alturas de los Andes.

Paquete turístico: Se le considera un conjunto de servicios debido a que el momento de su operación cada programa involucra servicios tales como alojamiento, transporte, alimentos, bebidas, guiado, zonas arqueológicas y otros.



Conclusiones

En la elaboración del presente trabajo de investigación de tipo desarrollo aplicado se ha llegado a las siguientes conclusiones:

1. Se analizo los procesos funcionales que permiten gestionar los paquetes turísticos
2. Se determino los requerimientos para la implementación del Sistema de Información Web (SIW).
3. Se diseño la estructura funcional del del Sistema de Información Web (SIW)
4. Se documento e implemento el Sistema de Información Web (SIW)
5. Se testeo los resultados producidos por el Sistema de Información Web (SIW)
6. Se desplago la solución en las instancias definidas por la organización para la empresa y el cliente.



Recomendaciones

1. Se recomienda analizar los procesos funcionales en otras empresas del rubro turístico y así establecer mejoras en los procesos de gestión de información.
2. Se sugiere obtener más requerimientos y así agilizar los procesos y mejorar el Sistema de Información Web (SIW).
3. Se sugiere realizar una reingeniería a la estructura funcional del Sistema de Información Web (SIW), para que esta sea robusta.
4. Se recomienda mejorar la documentación para que sea interactiva y de fácil entendimiento, adicionalmente que el Sistema de Información Web (SIW) sea implementado con nuevas tecnologías para que así pueda ser utilizado en otras plataformas.
5. Se sugiere testear el Sistema de Información Web (SIW) con mas afluencia de usuarios para mejorar la disponibilidad del sistema.
6. Se sugiere desplegar la solución en nuevas empresas del rubro turístico para obtener y mejorar la funcionalidad del sistema.



Referencias

- Acdx. (2019). Digital Signature. *Digital Signature*.
- Achour, M. (2013). Manual de PHP. E.E.U.U.
- Affiliates, O. C. (2016). *Workbench Reference Manual*. Obtenido de <http://dev.mysql.com/>
- Anexo-06. (s.f.). *Código fuente la herramienta de monitoreo y control de datos*.
- Angel Cobo, P. G. (2005). *PHP Y MySQL*. Obtenido de <https://www.editdiazdesantos.com/wwwdat/pdf/9788479787066.pdf>
- Anteportamlatinam Valero, J. (2014). *Relevancia del E-Comerce para la empresa actual*. Soria: Universidad de Valladolid.
- Baeza, J. P. (2009). *Manual Arduino*. Obtenido de <https://rua.ua.es/dspace/bitstream/10045/11833/1/arduino.pdf>
- Bahit, E. (14 de Enero de 2012). *Aplicaciones MVC Modulares en PHP*. Obtenido de Aplicaciones MVC Modulares en PHP: [http://46.101.4.154/Libros/Aplicaciones%20MVC%20Modulares%20en%20PHP%20\(taller%20antiguo\).pdf](http://46.101.4.154/Libros/Aplicaciones%20MVC%20Modulares%20en%20PHP%20(taller%20antiguo).pdf)
- Bilicki, E. &. (2016).
- Bizagi Limited. (2016). *Bizagi BPM Suite*. Obtenido de Solución para Gestión de procesos de negocio (BPM): <http://www.bizagi.com>
- Cabanillas, A. R. (2010). *DISEÑO DE UN PROTOTIPO DE UN SISTEMA DE TRAZABILIDAD DE GANADO USANDO RFID*. Lima - Perú.
- Cádiz, U. d. (2018). *Comenzando con arduino*. Obtenido de [http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANEXO S_2011_2012/22232441_310201212102.pdf](http://www.uca.es/recursos/doc/Unidades/Unidad_Innovacion/Innovacion_Docente/ANEXO_S_2011_2012/22232441_310201212102.pdf)
- Casillas Santillan, L., Ginesta, M., & Perez Mora, O. (2009). *Bases de Datos en Mysql*. Barcelona: www.uoc.edu.
- Castilla, D. (2014). *A Hybrid Approach Using RUP and Scrum as a Software Development Strategy*. Florida.



- Chan, N. (2005). *Circuitos Turisticos, Programacion y Cotizacion*. Buenos Aires: Ediciones Turisticas.
- Chawla, K. (. (1998). *Composite materials*. New York: Springer.
- Choque, J. C. (2019).
- Cochran, D. (2019). *Twitter Bootstrap Web Development How-To*. Packt Publishing.
- corbacho-perez. (s.f.). *anexo* .
- Corcuera, P. (s.f.). *Programacion Arduino*. Obtenido de <https://personales.unican.es/corcuerp/progucont/slides/Arduino.pdf>
- CÓRDOVA FORERO, J. A. (2014). *IMPLEMENTACIÓN DE UN SISTEMA DE MATRÍCULAS Y PAGOS PARA EL CENTRO DE INFORMÁTICA DE LA UNIVERSIDAD CÉSAR VALLEJO*. Lima.
- Córdova Zamora, M. (2000). *Estadística descriptiva e inferencial* (Cuarta ed.). Lima, Perú: Moshera S.R.L. Recuperado el 21 de 09 de 2017
- Deitel, P. J. (2008). (2008). *Java: como programar*. Pearson educacion. Obtenido de <https://www.java.com/es/>
- Delgado, W. M. (2015). *ASP.NET MVC 6 - UNA GUÍA INTRODUCCTORIA*. Obtenido de <https://leanpub.com/aspnetmvc6-unaguaintroductoria>
- Dimes, T. (2015). *Conceptos Basicos de SCRUM*. Babelcube.
- Eguiluz, J. (2020). *Uniwebsidad*. Obtenido de Introducción a JavaScript: <https://uniwebsidad.com/>
- Fernández Alarcón, V. (2006). *Desarrollo de Sistemas de Información*.
- Figuerola, C., Zazo Rodríguez, A., & Alonso Berrocal, J. (s.f.). *Firma Digital*. Salamanca: Universidad de Salamanca.
- Gallego, R., & Sanz, J. C. (2001). Diccionario Akal del color. En R. Gallego, & J. C. Sanz.
- García, A. A. (Diciembre de 2007). *Desarrollo de herramientas web de gestión docente* . Obtenido de <http://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf>



- García, J. M. (11 de 12 de 2015). ARSYS. Obtenido de ¿Qué es Laravel?: <https://www.arsys.es/>
- Gardey, J. P. (2012). Definición de temperatura. Mexico.
- Gauchat, J. D. (2012). Barcelona: MARCOMBO S.A.
- Gauchat, J. D. (2016). Obtenido de <https://gutl.jovenclub.cu/wp-content/uploads/2013/10/El+gran+libro+de+HTML5+CSS3+y+Javascrrip.pdf>
- Gianni, R. d. (2011). *El libro del BPM 2011*.
- Gómez Vieites, A. (2010). *Sistemas de Información en las Empresas*.
- gonzalez gisela, c. c. (s.f.). *Calor y temperatura*. Obtenido de <http://www.elgeometra.com.ar/fisica/trabajos%20practicos/2010/B/CALOR.pdf>
- Group, The PHP. (10 de 05 de 2018). *PHP*. Obtenido de <http://php.net/manual/es/preface.php>
- Herederero, C. d. (2018). *Dirección y Gestión de los Sistemas de Información en la Empresa*. ESIC Editorial.
- Hernández, S., Mancilla, D., Narvaez, C., & Iregui, M. (2013). Caso de estudio para validar la representación del conocimiento en aplicaciones web centradas en el usuario por medio de contenido multimedia enriquecido.
- Huamán, C. E. (2015). *Reyna*.
- IBM. (2018). *IBM Cloud Platform*. Obtenido de <https://www.ibm.com/>
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *El proceso unificado de desarrollo de software*. Madrid: Pearson Educación S.A.
- Jiménez González, Y., & Vega Castro, J. (2011). *Que es Protocolo SET*. Arequipa: Universidad de la Salle.
- Jiménez, R. E. (8 de Enero de 2015). *Metodologías Ágiles de Desarrollo de Software Aplicadas a la*. Obtenido de <https://core.ac.uk/download/pdf/80296686.pdf>
- Kerzner. (2013). Dashboard diseños.
- kerzner. (2013). Metodología para el diseño de Dashboard. En kerzner, *Metodología para el diseño de Dashboard*.



Kniberg, H. (2007). *Scrum y XP desde las trincheras. Como hacemos Scrum*. Nueva York: C4Media, editor de InfoQ.com. .

Laravel. (2020). *Laravel*. Obtenido de Laravel: <https://laravel.com/>

Laudon, K. C. (2012). *Sistemas de informacion Gerencial. 12va*. Mexico: Pearson Educación.

López, J. L. (1994). Definicion Turismo-Turista. *Papers de Turisme*, 77-94.

López, M. (2009). Proceedings of theet al "Wireless Monitoring of the pH".

LUXÓMETRO 1335 . (s.f.). Obtenido de http://www.crana.org/themed/crana/files/docs/033/042/medidorniveliluminacion_3.pdf

M., C. M. (2006). *RFID: El código de barras inteligente para Bibliotecas*. Santiago - Chile.

Marakas, J. A. (2006). *Sistemas de Información Gerencial*.

Maria Loureiro, J. P. (julio de 2017). *Taller robotica libre con arduino*. Obtenido de Arduino y el open Source: <http://educalab.es/documents/10180/640047/TallerRoboticaLibreArduino.pdf/c77adbf4-606a-4f4e-acd4-11630927b5a4>

Marques, M. P. (2011). *SQL Server 2008 R2 Motor de Base de datos y administracion*. Madrid: RC Libros.

Materiales de laboratorio. (2017). Obtenido de <https://materialeslaboratorio.com/luxometro/>

Medidas de a luz. (s.f.). Obtenido de <http://www2.fisica.unlp.edu.ar/materias/FEIII/OLD/2012/IntensidadLuminosa.pdf>

Meneses, C. P. (2012). *DISEÑO DE UN SISTEMA DE CONTROL DE ACTIVOS PARA EL ALMACÉN DE ELECTRÓNICA DE LA PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ UTILIZANDO RFID*. Lima - Perú.

Mera, H. M. (2013). *MERA, HEINER MAHECHA*.

Mestras, J. P. (s.f.). *PHP Aplicaciones web/sistemas web*. Obtenido de <https://www.fdi.ucm.es/profesor/jpavon/web/33-PHP.pdf>

Microsoft. (2012). *Microsoft Developer Network*. Obtenido de Documentación: [https://msdn.microsoft.com/es-es/library/fx6bk1f4\(v=vs.100\).aspx](https://msdn.microsoft.com/es-es/library/fx6bk1f4(v=vs.100).aspx)



Microsoft. (2016). *Developer Network*. Obtenido de <https://msdn.microsoft.com/es-es/library/bb545450.aspx>

Mincetur. (2016). *Reglamento de Agencias de Viajes y Turismo*. Lima: El Peruano.

Mogetta, A., Audisio, J., & Casco, D. (2014). *Informática III – Internet – Intranet - Extranet*. Escuela Superior de Comercio “Lib. Gral. San Martín” (UNR): Departamento de Tecnologías de la Información y las Comunicaciones.

Mora, S. L. (2002). *Programación de Aplicaciones Web: Historia, Principios Básicos y Clientes Web*. Alicante: Editorial Club Universitario.

Murphey, R. (Agosto de 2013). *Fundamentos de jQuery*. Obtenido de <http://librojquery.com/>

Ofimática Profesional - Corinne Hervo. (2013). *Microsoft Excel 2013*. Barcelona: Ediciones ENI.

Oracle. (2016). *NetBeans IDE*. Obtenido de https://netbeans.org/index_es.html

Oracle. (2016). *Oracle Documentación*. Obtenido de <http://www.oracle.com/technetwork/es/database/enterprise-edition/documentation/index.html>

PBWORK. (2008). *Cursos de Ingeniería de Software*. Obtenido de <https://ingsw.pbworks.com/f/Ciclo+de+Vida+del+Software.pdf>

Pérez, J. E. (17 de Diciembre de 2010). Obtenido de http://www.jesusda.com/docs/ebooks/introduccion_ajax.pdf

Pérez, J. E. (17 de Diciembre de 2010). *Introducción a XHTML*. Obtenido de http://www.jesusda.com/docs/ebooks/introduccion_xhtml.pdf

PHP. (2020). *PHP*. Obtenido de ¿Qué es PHP?: <https://www.php.net/>

Quispe, B. E. (2017). Sistema de información para la gestión de reserva de paquetes turísticos en la agencia de viaje RAP TRAVEL. CUSCO, PERU: UNIVERSIDAD ANDINA DEL CUSCO.

Ralph M. Stair, George W. Reynolds. (2001). *Principios de Sistema de Información. Enfoque administrativo*. Madrid: Ediciones Paraninfo.

Rancel, M. R. (2013). *Aprende a programar*. Obtenido de ww.aprenderaprogramar.com



ROMO, P. O. (2016).

Saavedra, I. N. (2009). *Sistema de Información via web para la promoción y administración de servicios caso: EMISTUR*. La Paz.

Salazar Caraballo, L. (2013). *Asuntos de la Ingeniería de software*. Obtenido de <http://www.gazafatonarioit.com>

Sanchez, J. (2015). *MySQL guía rápida*. Obtenido de <http://www.cartagena99.com/recursos/programacion/apuntes/mysql.pdf>

Sanchez, J. A. (2008). *Sistema de control de acceso con RFID*. México D.F., México.

SAS, DOINTECH. (2015). *DOINTECH SAS, Automatización, seguridad y control*. Obtenido de <http://www.dointech.com.co/control-de-acceso.html>

Schwaber, K., & Sutherland, J. (Julio de 2013). *Scrum guides*. Obtenido de La guía de Scrum: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Schwaber, K., & Sutherland, J. (2014). *La Guía Definitiva de Scrum: Las Reglas del Juego*.

SIGMA ELECTRÓNICA LTDA. (2016). *SIGMA ELECTRÓNICA LTDA*. Obtenido de <http://www.sigmaelectronica.net/sl500l-p-1971.html>

SOFTENG. (2016). *SOFTENG*. Obtenido de Metodología Scrum para desarrollo de software - aplicaciones complejas: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>

Sparx Systems. (2016). *Sparx Systems*. Obtenido de Enterprise Architect - Herramienta de diseño UML: <http://www.sparxsystems.com.ar/products/ea.html>

Trigas Gallego, M. (s.f.). *Gestión de proyectos Informáticos*. Obtenido de Metodología SCRUM: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>

Universidad de Alicante. (2015). *Servicio de Informática ASP.NET MVC 3 Framework*. Obtenido de Modelo vista controlador (MVC): <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

zona, M. (septiembre de 2014). *Manual de arduino programacion y conceptos basicos*.





Anexos

❖ **Anexo 01:** Ficha de observación. de sprint: historia usuaria, tarea, prueba

FICHA HISTORIA USUARIO:

Historia de Usuario	
ID	HUXX
NOMBRE	
PRIORIDAD	
RIESGO	
DESCRIPCION	
VALIDACION	

FICHA TAREAS

TAREA	
ID	TXX
Historia del usuario	
ESTADO	
DESCRIPCION	

FICHA PRUEBAS

Nombre de caso de prueba:	Iniciar herramienta <u>Dashboard</u>
Entradas:	
Condiciones:	



❖ Anexo 02: Ficha de observación. de sprint: historia usuaria, tarea, prueba



❖ Anexo 03: PERU JAMUY – Personal que trabajan en planta y campo





❖ Anexo 04: PERU JAMUY – Personal realizando los paquetes ofrecidos



❖ Anexo 05: PERU JAMUY – Personal de campo





➤ ANEXO 06: Código fuente – Controlador Manual a la oferta

```
<?php

namespace App\Http\Controllers\Administrar\Paquete;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Response;

use App\PACombinacion;
use App\PADescripcion;
use App\PAExcursion;
use App\PAHospedaje;
use App\PANoches;
use App\PArmado;
use App\PAServicio;
use App\PATicket;
use App\Servicio;
use App\SHospedajeServicio;
use App\TipoTransporte;
use App\VPaquete;

use Barryvdh\DomPDF\Facade as PDF;

class ManualController extends Controller
{
    public function __construct()
    {
        $this->middleware('login');
        $this->middleware('privilege:admpaqmanual');
    }
}
```



```
public function index(Request $request)
{
    $servicio = Servicio::where('cuenta_id', $this->getCuenta())->get();
    return view('administrar.paquete.manual')->with('servicio', $servicio);
}

public function resumen(Request $request, $id)
{
    // Nos bajaremos cualquier operador seleccionado, para trabajar con los precios de la cuenta actual
    $request->session()->forget('cuenta_cambio');

    $tipotransporte = TipoTransporte::select('id', 'nombre')->get();

    $paquete = VPaquete::where('cuenta_id', $this->getCuenta())->findOrFail($id);
    $todo = PArmado::select('no_incluye', 'politica', 'recomendacion', 'detalles', 'noches')->findOrFail($id);

    $capacidad = SHospedajeServicio::select('capacidad')->where('cuenta_id', $this->getCuenta())->orderBy('capacidad')->groupBy('capacidad')->get();

    # variables auxiliares
    $res_hos = collect(); $res_tic = collect(); $res_exc = collect();
    $res_ser = collect(); $res_noc = collect(); $res_des = collect();
    $res_destino = collect(); $res_combinacion = collect();

    $res_servicio_excursion = collect();

    $contador = 0;

    // alojamiento
    if($paquete->alojamientos > 0) {
```




```
$hospedaje = PAHospedaje::with('hospedaje')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($hospedaje as $item) {
    $contador++;
    $res_hos->push([
        'id' => $contador,
        'hospedaje_id' => $item->proveedor_id,
        'nombre' => $item->hospedaje->nombre,
        'costos' => $item->hospedaje->precios,
        'distrito_id' => $item->hospedaje->distrito_id,
        'distrito_name' => $item->hospedaje->ubicacion->generado,
        'categoria' => $item->hospedaje->categoria,
        'checkin' => $item->hospedaje->checkin ?? 'No Especifica',
        'checkout' => $item->hospedaje->checkout ?? 'No Especifica',
    ]);
}

$combinacion = PACombinacion::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
$res_destino = $this->GenerarDestinos($res_hos);

foreach($combinacion as $index => $item) $res_combinacion->push(json_decode($item->combinacion, true));
}

//ticket
if($paquete->boletos > 0) {
    $ticket = PATicket::with('ticket')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($ticket as $item) {
        $costos_nac = 0; $costos_ext = 0;
        foreach ($item->ticket->precios as $data) {
            if ($data->nombre == 'NACIONAL') {
                $costos_nac = $data->adulto;
            }
        }
    }
}
```



```
break;
}
}
foreach ($item->ticket->precios as $data) {
if ($data->nombre == 'EXTRANJERO') {
$costos_ext = $data->adulto;
break;
}
}
$contador++;
$res_tic->push([
'id' => $contador,
'boleto_id' => $item->ticket_id,
'dolares' => $item->ticket->dolares ?? '0',
'nombre' => $item->ticket->generado,
'detalle' => $item->ticket->zonas_name,
'costos_nac' => number_format($costos_nac, 2, '.', ''),
'costos_ext' => number_format($costos_ext, 2, '.', ''),
]);
}
}
//excursion
if($paquete->excursiones > 0) {
$excursion = PAExcursion::with(['servicio.servicio_exc' => function($query) {
$query->with('servicio_sel_tran.tipotransporte')->with('servicio_sel_vari.varios')->with('servicio_sel_guia');
}])->with(['excursion' => function($query) {
$query->with('servicio')->with('endosar')->withCount('endosar')->withCount('servicio');
}])->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($excursion as $item) {
$costos_nac = 0; $costos_ext = 0;
```



```
foreach ($item->excursion->costos as $index => $data) {
if ($index == 0) {
$costos_nac = $data->nacional;
$costos_ext = $data->extranjero;
break;
}
}
//servicios
$res_exc_ser = collect();
foreach($item->servicio as $data) {
if($data->servicio_exc) {
$data->servicio_exc->vehiculo_id = $data->vehiculo_id ?? '0';
if($data->servicio_exc->tipo == '0') {
$costos_nac_s = 0;
$costos_ext_s = 0;

$estatus = true;
foreach($data->servicio_exc->servicio_sel_tran->precios as $index => $ytem) {
$split = explode(' | ', $ytem->nombre);
if($split[1] == $data->servicio_exc-
>vehiculo_id && $split[0] == 'NACIONAL' && $estatus) {
$costos_nac_s = $data->servicio_exc->servicio_sel_tran->tipotransporte-
>variable == '1' ? $ytem->adulto : $ytem->autorizado;
}
}
if($split[1] == $data->servicio_exc-
>vehiculo_id && $split[0] == 'EXTRANJERO' && $estatus) {
$costos_ext_s = $data->servicio_exc->servicio_sel_tran->tipotransporte-
>variable == '1' ? $ytem->adulto : $ytem->autorizado;
}
}
}
}
}
$data->servicio_exc->costos_nac = $costos_nac_s;
$data->servicio_exc->costos_ext = $costos_ext_s;
}
```



```
$res_exc_ser->push($data->servicio_exc);
}
}
$contador++;
$res_servicio_excursion->push([
'version' => 'excursion',
'id' => $contador,
'orden' => $item->orden,
'excursion_id' => $item->excursion_id,
'nombre' => $item->excursion->generado,
'costos_nac' => $item->excursion-
>endosar_count != '0' ? number_format($costos_nac, 2, '.', '') : $item-
>excursion->precios,
'costos_ext' => $item->excursion-
>endosar_count != '0' ? number_format($costos_ext, 2, '.', '') : $item-
>excursion->precios,
'endosado' => $item->excursion->endosar_count,
'servicios' => $res_exc_ser,
'dia' => $item->dia,
]);
}
}
//servicio
if($paquete->servicios > 0) {
$servicio = PAservicio::with('servicio_sel_tran')->with('servicio_sel_vari')-
>with('servicio_sel_guia')->where('paquete_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();
foreach($servicio as $item ) {
// transporte
if($item->tipo == '0') {
$costos_nac_s = 0;
$costos_ext_s = 0;
$estatus = true;
```



```
// Privado && Variable
if($item->servicio_sel_tran->compartido == '0' && $item->servicio_sel_tran->tipotransporte->variable != '1') {
if($item->vehiculo_id == null) {
$costos_nac_s = 0; $costos_ext_s = 0;
} else {
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
$split = explode(' | ', $ytem->nombre);
if($split[1] == $item->vehiculo_id && $split[0] == 'NACIONAL' && $estatus) {
$costos_nac_s = $item->servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
}
if($split[1] == $item->vehiculo_id && $split[0] == 'EXTRANJERO' && $estatus) {
$costos_ext_s = $item->servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
}
}
}
} else {
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
$split = explode(' | ', $ytem->nombre);
if($split[0] == 'NACIONAL' && $estatus) {
$costos_nac_s = $item->servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
}
if($split[0] == 'EXTRANJERO' && $estatus) {
$costos_ext_s = $item->servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
$estatus = FALSE;
}
}
}
}
```



```
$contador++;  
$res_servicio_excursion->push([  
  'version' => 'servicio',  
  'id' => $contador,  
  'orden' => $item->orden,  
  'servicio_id' => $item->servicio_id,  
  'compartido' => $item->servicio_sel_tran->compartido,  
  'precios' => $item->servicio_sel_tran->precios,  
  'nombre' => $item->servicio_sel_tran->generado,  
  'dia' => $item->dia,  
  'variable' => $item->servicio_sel_tran->tipotransporte->variable,  
  'costos_nac' => $item->costo_nacional ?? $costos_nac_s,  
  'costos_ext' => $item->costo_extranjero ?? $costos_ext_s,  
  'vehiculo_id' => $item->vehiculo_id ?? '0',  
  'tipo' => $item->tipo,  
]);  
}  
// varios  
if($item->tipo == '1') {  
  $costo_name = 0;  
  foreach($item->servicio_sel_vari->precios as $index => $ytem) {  
    if ($index == 0) {  
      $costo_name = $ytem->adulto;  
      break;  
    }  
  }  
}  
  
$contador++;  
$res_servicio_excursion->push([  
  'version' => 'servicio',  
  'id' => $contador,  
  'orden' => $item->orden,
```



```
'servicio_id' => $item->servicio_id,  
'compartido' => '0',  
'precios' => $item->servicio_sel_vari->precios,  
'nombre' => $item->servicio_sel_vari->generado,  
'dia' => $item->dia,  
'variable' => '0',  
'costos_nac' => $costo_name,  
'costos_ext' => $costo_name,  
'tipo' => $item->tipo,  
]);  
}  
  
// guiado  
if($item->tipo == '3') {  
$costo_name = 0;  
foreach($item->servicio_sel_guia->precios as $index => $ytem) {  
if ($index == 0) {  
$costo_name = $ytem->autorizado;  
break;  
}  
}  
  
$contador++;  
$res_servicio_excursion->push([  
'version' => 'servicio',  
'id' => $contador,  
'orden' => $item->orden,  
'servicio_id' => $item->servicio_id,  
'compartido' => $item->servicio_sel_guia->compartido,  
'precios' => $item->servicio_sel_guia->precios,  
'nombre' => $item->servicio_sel_guia->generado,  
'dia' => $item->dia,  
'variable' => '0',
```



```
'costos_nac' => $costo_name,  
'costos_ext' => $costo_name,  
'tipo' => $item->tipo,  
]);  
}  
}  
}  
  
//noches  
if($paquete->noches > 0) {  
$noches = PANoches::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();  
foreach($noches as $item) {  
$res_noc->push([  
'id' => $item->identificador,  
'cont' => $item->variable,  
'cant' => $item->cantidad,  
]);  
}  
}  
  
//descripcion  
if($paquete->descripciones > 0) {  
$descripcion = PADescripcion::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();  
foreach($descripcion as $item) {  
$res_des->push([  
'dia' => $item->dia,  
'descripcion' => $item->descripcion,  
]);  
}  
}
```




```
return view('administrar.paquete.manual_resumen')->with('paquete', $paquete)->with('todo', $todo)
->with('capacidad', $capacidad)->with('tipotransporte', $tipotransporte)
->with('contador', $contador)
->with('res_hos', $res_hos)->with('res_tic', $res_tic)
->with('res_exc', $res_exc)->with('res_ser', $res_ser)
->with('res_servicio_excursion', $res_servicio_excursion->sortBy('orden'))
->with('res_noc', $res_noc)->with('res_des', $res_des)
->with('res_combinacion', $res_combinacion)->with('res_destino', $res_destino->unique('distrito_id'));
}

protected function GenerarDestinos($grupal)
{
$destino = collect();
foreach($grupal as $data) {
$destino->push([
'distrito_id' => $data["distrito_id"],
'distrito_name' => $data["distrito_name"],
]);
}
return $destino;
}

public function imprimir(Request $request, $id, $tipo)
{
$usuario = $request->user();
$ahora = now();
$totalnacional = 0; $totalextranjero = 0;

$resumen = VPaquete::where('cuenta_id', $this->getCuenta())->findOrFail($id);
```



```
$modelo = PArmado::withCount('alojamiento')->withCount('ticket')->withCount('noches')->withCount('descripcion')->withCount('excursiones')->withCount('servicios')->withCount('adicional')->where('cuenta_id', $this->getCuenta())->find($id);

$capacidad = SHospedajeServicio::select('capacidad')->where('cuenta_id', $this->getCuenta())->orderBy('capacidad')->groupBy('capacidad')->get();

$margen = (100 - floatval($modelo->cantidad)) / 100;

$incentivo = $modelo->incentivo ?? 10;

# Variables Auxiliares
$incluye = collect();
$allservicio = collect();
$itinerario = collect();
$adicionales = collect();
$allalojamiento = array();

$allcombinacion = array();

$tipo_transportes = 0;$tipo_alimentos = 0;$tipo_guiado = 0;

# Descripcion
if($modelo->descripcion_count > 0) {
    $descripcion = PADescripcion::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    $itinerario = $descripcion->sortBy('dia');
}

# Ticket
if($modelo->ticket_count > 0) {
    $ticket = PATicket::with('ticket')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
}
```



```
foreach($ticket as $item) {
$costos_nac = 0; $costos_ext = 0;
foreach ($item->ticket->precios as $data) {
if ($data->nombre == 'NACIONAL') {
$totalnacional += $item->ticket->dolares == '0' ? (floatval($data->adulto) / floatval($modelo->tipo_cambio ?? 3.20)) : floatval($data->adulto);
break;
}
}
foreach ($item->ticket->precios as $data) {
if ($data->nombre == 'EXTRANJERO') {
$totalextranjero += $item->ticket->dolares == '0' ? (floatval($data->adulto) / floatval($modelo->tipo_cambio ?? 3.20)) : floatval($data->adulto);
break;
}
}
}
}
# Servicios
if($modelo->servicios_count > 0) {
$servicio = PAservicio::with('servicio_sel_tran')->with('servicio_sel_vari')->with('servicio_sel_guia')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($servicio as $item ) {
$nombre = ''; $variable = 0;
if($item->tipo == '0') {
$tipo_transportes++;
$nombre = $item->servicio_sel_tran->nombre .' | '.($item->servicio_sel_tran->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO') . ($item->vehiculo_id == NULL || $item->vehiculo_id == '0' ? '' : (' => ' . $item->vehiculo_id));
if(!$item->servicio_sel_tran->tipotransporte->variable) {
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
```



```
if (strrpos($ytem->nombre, chr('NACIONAL')) >= 0 && strrpos($ytem->nombre, chr($item->vehiculo_id ?? '0')) >= 0) {
$totalnacional += floatval($ytem->autorizado);
break;
}
}
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
if ( strrpos($ytem->nombre, chr('EXTRANJERO')) >= 0 && strrpos($ytem->nombre, chr($item->vehiculo_id ?? '0')) >= 0) {
$totalextranjero += floatval($ytem->autorizado);
break;
}
}
} else {
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
if ($ytem->nombre == 'NACIONAL') {
$totalnacional += $item->costo_nacional ?? floatval($ytem->adulto);
break;
}
}
foreach($item->servicio_sel_tran->precios as $index => $ytem) {
if ($ytem->nombre == 'EXTRANJERO') {
$totalextranjero += $item->costo_extranjero ?? floatval($ytem->adulto);
break;
}
}
}
}
if($item->tipo == '1') {
$tipo_alimentos++;
$nombre = $item->servicio_sel_vari->nombre . ' | ' . $item->servicio_sel_vari->varios->nombre . ' | ' . ($item->servicio_sel_vari->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO');
}
```



```
foreach($item->servicio_sel_vari->precios as $index => $ytem) {
if ($index == 0) {
$totalnacional += floatval($ytem->adulto == 0 ? 0 : $ytem->adulto);
$totalextranjero += floatval($ytem->adulto == 0 ? 0 : $ytem->adulto);
break;
}
}
}
if($item->tipo == '3') {
$tipo_guiado++;
$nombre = $item->servicio_sel_guia->nombre .' | ' .($item->servicio_sel_guia->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO');
foreach($item->servicio_sel_guia->precios as $index => $ytem) {
if ($index == 0) {
$totalnacional += floatval($ytem->autorizado == 0 ? 0 : $ytem->autorizado);
$totalextranjero += floatval($ytem->autorizado == 0 ? 0 : $ytem->autorizado);
break;
}
}
}
}
}
}
# Excursiones
if($modelo->excursiones_count > 0) {
$excursion = PAExcursion::with(['servicio.servicio_exc' => function($query) {
$query->with('servicio_sel_tran.tipotransporte')-
>with('servicio_sel_vari.varios')->with('servicio_sel_guia');
}])->with(['excursion' => function($query) {
$query->with('servicio')->with('endosar')->withCount('endosar')-
>withCount('servicio');
}])->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($excursion as $item) {
```



```
if($item->excursion->endosar_count == 0) {
foreach($item->servicio as $data) {
if($data->servicio_exc) {
if($data->servicio_exc->tipo == '0') {
$tipo_transportes++;
if($data->servicio_exc->servicio_sel_tran->tipotransporte->variable) {
foreach($item->servicio_exc->servicio_sel_tran->precios as $index => $ytem) {
if ($ytem->nombre == 'NACIONAL') {
$totalnacional += floatval($ytem->adulto);
break;
}
}
}
foreach($item->servicio_exc->servicio_sel_tran->precios as $index => $ytem) {
if ($ytem->nombre == 'EXTRANJERO') {
$totalextranjero += floatval($ytem->adulto);
break;
}
}
} else {
foreach($data->servicio_exc->servicio_sel_tran->precios as $index => $ytem) {
if (strpos($ytem->nombre, chr('NACIONAL')) >= 0 && strpos($ytem->nombre, chr($data->vehiculo_id)) >= 0) {
$totalnacional += floatval($ytem->autorizado);
break;
}
}
}
foreach($data->servicio_exc->servicio_sel_tran->precios as $index => $ytem) {
if ( strpos($ytem->nombre, chr('EXTRANJERO')) >= 0 && strpos($ytem->nombre, chr($data->vehiculo_id)) >= 0) {
$totalextranjero += floatval($ytem->autorizado);
break;
}
}
}
```



```
}  
}  
}  
if($data->servicio_exc->tipo == '1') {  
    $tipo_alimentos++;  
    foreach($data->servicio_exc->servicio_sel_vari->precios as $index => $ytem) {  
        if ($index == 0) {  
            $totalnacional += ($ytem->adulto == 0 ? 0 : $ytem->adulto);  
            $totalextranjero += $ytem->adulto == 0 ? 0 : $ytem->adulto;  
            break;  
        }  
    }  
}  
if($data->servicio_exc->tipo == '3') {  
    $tipo_guiado++;  
    foreach($data->servicio_exc->servicio_sel_guia->precios as $index => $ytem) {  
        if ($index == 0) {  
            $totalnacional += $ytem->autorizado == 0 ? 0 : $ytem->autorizado;  
            $totalextranjero += $ytem->autorizado == 0 ? 0 : $ytem->autorizado;  
            break;  
        }  
    }  
}  
} else {  
    foreach ($item->excursion->costos as $index => $data) {  
        if ($index == 0) {  
            $totalnacional += $data->nacional;  
            $totalextranjero += $data->extranjero;  
            break;  
        }  
    }  
}
```



```
}  
}  
}  
}  
# Alojamiento  
if($modelo->alojamiento_count > 0) {  
$combinacion = PACombinacion::where('paquete_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
$hospedaje = PAHospedaje::with('hospedaje')->where('paquete_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
if($hospedaje) {  
$alojamientos = collect();  
$contador = 0;  
foreach($hospedaje as $item) {  
$contador++;  
$noches_seleccionadas = PANOches::select('cantidad')-  
>where('identificador', $item->hospedaje->distrito->id)-  
>where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->first();  
$alojamientos->push((object)[  
'id' => $contador,  
'hospedaje_id' => $item->proveedor_id,  
'nombre' => $item->hospedaje->nombre . ' ('.$item->hospedaje->categoria.'*)',  
'costos' => $item->hospedaje->precios,  
'distrito_name' => $item->hospedaje->distrito->nombre,  
'noches' => $noches_seleccionadas->cantidad ?? '0',  
]);  
}  
//dd($alojamientos);  
foreach($combinacion as $index => $item) {  
$_nombres = array(); $_precios = array();  
foreach(json_decode($item->combinacion, true) as $ytem) {
```




```
foreach($alojamientos as $ztem) {
if($ztem->hospedaje_id == $ytem) {
array_push($_nombres, $ztem->nombre);
foreach($ztem->costos as $w => $wtem) {
foreach($wtem as $x => $xtem) {
$montopasajero = ($w == 'nacional' ? $totalnacional : $totalextranjero);
$nuevomonto = (isset($_precios[$w][$xtem["capacidad"]]) ? $_precios[$w][$xtem["capacidad"]] + floatval($xtem["monto"]) : floatval($xtem["monto"]) * (intval($ztem->noches) / intval($xtem["capacidad"])));
$_precios[$w][$xtem["capacidad"]] = $nuevomonto;
}
}
}
}
}

if(count($_precios) > 0) array_push($allalojamiento, array('nombre' => implode(' | ', $_nombres), 'precios' => $_precios));
}
}
}

if($tipo == "PDF") {
$pdf = PDF::loadView('administrar.paquete.manual_imprimir', compact('usuario', 'ahora', 'modelo', 'itinerario', 'resumen', 'tipo_transportes', 'tipo_alimentos', 'tipo_guiado', 'capacidad', 'allalojamiento', 'totalnacional', 'totalextranjero', 'margen', 'incentivo'));
return $pdf->download('Paquete_'.time().'.pdf');
} elseif($tipo == "WORD") {
$headers = array(
"Content-type"=>"text/html",
"Content-Disposition"=>"attachment;Filename=".$resumen->nombre.".doc"
);
```



```
$texto_cada_dia = "";
if($resumen->descripciones > 0) {
$descripcion = PADescripcion::where('paquete_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
foreach($descripcion as $item) {
$contenido = "";
if($item->descripcion != '<p><br></p>' && $item->descripcion != NULL && $item-
>descripcion != '') {
$contenido = $item->descripcion;
} else {
$contenido = "Día Libre";
}

$texto_cada_dia .= "<h3>DIA ".(intval($item->dia)+1)."</h3>
<p>".$contenido."</p>";
}
}

$tabla_contenido = "";
if(count($allalojamiento) > 0) {
$cantidad_noches = PANoches::where('paquete_id', $id)->sum('cantidad');
$cabecera_tabla = '';
foreach($capacidad as $index => $item) $cabecera_tabla .= '<th>'.$item-
>tuple.'WB</th>';

$tabla_contenido = '<table border="1" width="100%">
<thead>
<tr>
<th rowspan="2">Hoteles + Servicios</th>
<th colspan="'.count($capacidad).'">Nacional</th>
<th colspan="'.count($capacidad).'">Extranjero</th>
</tr>
```



```
<tr>'. $cabecera_tabla.''. $cabecera_tabla.'  
</tr>  
</thead>  
<tbody>';  
foreach($allalojamiento as $item) {  
$tabla_contenido .= '<tr><td>'. $item["nombre"]. '</td>';  
foreach($item["precios"] as $y => $ytem) {  
foreach($ytem as $ztem) {  
if(floatval($ztem) > 0) {  
$nuevovalor = floatval($ztem) + floatval($y == 'nacional' ? $totalnacional : $  
totalextranjero);  
$tabla_contenido .= '<td>$ '. number_format(round(($nuevovalor/$margen) + $ince  
tivo, 0), 0, '.', ''). '</td>';  
} else {  
$tabla_contenido .= '<td>$ 0.00</td>';  
}  
}  
}  
$tabla_contenido .= '</tr>';  
}  
$tabla_contenido .= '</tbody></table>';  
} else {  
$tabla_contenido = '<table border="1" width="100%">  
<thead>  
<tr>  
<th></th>  
<th>Nacional</th>  
<th>Extranjero</th>  
</tr>  
</thead>  
  
<tfoot>
```



```
<tr>
<td>PRECIO TOTAL</td>
<td>$ '.number_format(round(($totalnacional / $margen) + $incetivo, 0), 0, '.',
, '').'</td>
<td>$ '.number_format(round(($totalextranjero / $margen) + $incetivo, 0), 0, '
.', '').'</td>
</tr>
</tfoot>
</table>';
}

$content = '<html>
<head><meta charset="utf-8"></head>
<body>
<h1>' . $resumen->nombre . '</h1>
<p>' . $modelo->detalles . '</p>
<h2>Itinerario</h2>
' . $texto_cada_dia . '
<h2>Incluye</h2>
<ul>
' . ($resumen->alojamientos > 0 ? '<li>Alojamientos - ' . $cantidad_noches . ' Noche(s) y ' . $cantidad_noches . ' Desayuno(s)</li>' : '').'
' . ($resumen->boletos > 0 ? '<li>Boletos Turisticos</li>' : '').'
' . ($resumen->servicios > 0 && $tipo_transportes > 0 ? '<li>Transportes</li>' : '').'
' . ($resumen->servicios > 0 && $tipo_alimentos > 0 ? '<li>Alimentos</li>' : '').'
' . ($resumen->servicios > 0 && $tipo_guiado > 0 ? '<li>Guiado Profesional</li>' : '').'
<li>Asistencia Permanente</li>
</ul>
' . ($modelo->no_incluye != '<p><br></p>' ? '<h2>No Incluye</h2>' . $modelo->no_incluye : '').'
```



```
<h2>Presupuesto</h2>
'.$tabla_contenido.'
</body>
</html>';

        return Response::make($content,200, $headers);
    }

}
}
```

➤ **ANEXO 07:** Código fuente – Controlador Armado de paquetes

```
<?php

namespace App\Http\Controllers\Administrar\Paquete;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Barryvdh\DomPDF\Facade as PDF;

use App\Servicio;
use App\PArmado;
use App\SHospedajeServicio;
use App\PAHospedaje;
use App\PATicket;
use App\PAExcursion;
use App\PAServicio;
use App\URegion;
use App\PANoches;
use App\PADescripcion;
use App\PAAdicional;
use App\PACombinacion;
use App\TipoTransporte;
```



```
class ArmadoController extends Controller
{
    public function __construct()
    {
        $this->middleware('login');
        $this->middleware('privilege:admpaqarmado');
    }

    public function index(Request $request)
    {
        $servicio = Servicio::where('cuenta_id', $this->getCuenta())->get();
        return view('administrar.paquete.armado')-
>with('servicio', $servicio);
    }

    public function resumen(Request $request, $id)
    {
        $request->session()->forget('cuenta_cambio');

        $servicio = Servicio::where('cuenta_id', $this->getCuenta())->get();
        $region = URegion::all();

        $tipotransporte = TipoTransporte::select('id', 'nombre')->get();

        $paquete = PArmado::withCount('alojamiento')->withCount('ticket')-
>withCount('noches')->withCount('descripcion')
        ->withCount('excursiones')->withCount('servicios')-
>withCount('adicional')->where('cuenta_id', $this->getCuenta())->find($id);

        $capacidad = SHospedajeServicio::select('capacidad')-
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-
>groupBy('capacidad')->get();

        # variables auxiliares
```



```
$res_hos = collect();
$res_tic = collect();
$res_exc = collect();
$res_ser = collect();
$res_noc = collect();
$res_des = collect();
$res_upd = collect();

$res_destino = collect(); $res_combinacion = collect();

$contador = 0;
// hospedajes

if($paquete->alojamiento_count > 0) {
    $hospedaje = PAHospedaje::with('hospedaje')-
>where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($hospedaje as $item) {
        $contador++;
        $res_hos->push([
            'id' => $contador,
            'hospedaje_id' => $item->proveedor_id,
            'nombre' => $item->hospedaje->nombre,
            'costos' => $item->hospedaje->precios,
            'distrito_id' => $item->hospedaje->distrito_id,
            'distrito_name' => $item->hospedaje->ubicacion->generado,
            'categoria' => $item->hospedaje->categoria,
            'checkin' => $item->hospedaje-
>checkin ?? 'No Especifica',
            'checkout' => $item->hospedaje-
>checkout ?? 'No Especifica',
        ]);
    }
}
```



```
$combinacion = PACombinacion::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($combinacion as $index => $item) {
        if($index == 0) $res_destino = $this->GenerarDestinos(json_decode($item->combinacion, true), $res_hos);
        $res_combinacion->push(json_decode($item->combinacion, true));
    }
}
//ticket
if($paquete->ticket_count > 0) {
    $ticket = PATicket::with('ticket')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($ticket as $item) {
        $costos_nac = 0; $costos_ext = 0;
        foreach ($item->ticket->precios as $data) {
            if ($data->nombre == 'NACIONAL') {
                $costos_nac = $data->adulto;
                break;
            }
        }
        foreach ($item->ticket->precios as $data) {
            if ($data->nombre == 'EXTRANJERO') {
                $costos_ext = $data->adulto;
                break;
            }
        }
        $contador++;
        $res_tic->push([
            'id' => $contador,
            'boleto_id' => $item->ticket_id,
            'dolares' => $item->ticket->dolares ?? '0',
```




```
        'nombre' => $item->ticket->generado,  
        'detalle' => $item->ticket->zonas_name,  
        'costos_nac' => number_format($costos_nac, 2, '.', ''),  
        'costos_ext' => number_format($costos_ext, 2, '.', ''),  
    ]);  
    }  
}  
  
//excursion  
if($paquete->excursiones_count > 0) {  
    $excursion = PAExcursion::with(['servicio.servicio_exc' => function($query) {  
        $query->with('servicio_sel_tran.tipotransporte')->  
>with('servicio_sel_vari.varios')->with('servicio_sel_guia');  
    }])->with(['excursion' => function($query) {  
        $query->with('servicio')->with('endosar')->  
>withCount('endosar')->withCount('servicio');  
    }])->where('paquete_id', $id)->where('cuenta_id', $this->  
>getCuenta())->get();  
    foreach($excursion as $item) {  
        $costos_nac = 0; $costos_ext = 0;  
        foreach ($item->excursion->costos as $index => $data) {  
            if ($index == 0) {  
                $costos_nac = $data->nacional;  
                $costos_ext = $data->extranjero;  
                break;  
            }  
        }  
    }  
    //servicios  
    $res_exc_ser = collect();  
    foreach($item->servicio as $data) {  
        if($data->servicio_exc) {  
            $data->servicio_exc->vehiculo_id = $data->  
>vehiculo_id ?? '0';  
        }  
    }  
}
```



```
        if($data->servicio_exc->tipo == '0') {
            $costos_nac_s = 0;
            $costos_ext_s = 0;

            $estatus = true;
            foreach($data->servicio_exc->servicio_sel_tran-
>precios as $index => $ytem) {
                $split = explode(' | ', $ytem->nombre);
                if($split[1] == $data->servicio_exc-
>vehiculo_id && $split[0] == 'NACIONAL' && $estatus) {
                    $costos_nac_s = $data->servicio_exc-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;
                }
                if($split[1] == $data->servicio_exc-
>vehiculo_id && $split[0] == 'EXTRANJERO' && $estatus) {
                    $costos_ext_s = $data->servicio_exc-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;
                }
            }
            $data->servicio_exc->costos_nac = $costos_nac_s;
            $data->servicio_exc->costos_ext = $costos_ext_s;
        }
        $res_exc_ser->push($data->servicio_exc);
    }
}
$contador++;
$res_exc->push([
    'id' => $contador,
    'excursion_id' => $item->excursion_id,
    'nombre' => $item->excursion->generado,
```



```
        'costos_nac' => $item->excursion-
>endosar_count != '0' ? number_format($costos_nac, 2, '.', '') : $item-
>excursion->precios,
        'costos_ext' => $item->excursion-
>endosar_count != '0' ? number_format($costos_ext, 2, '.', '') : $item-
>excursion->precios,
        'endosado' => $item->excursion->endosar_count,
        'servicios' => $res_exc_ser,
        'dia' => $item->dia,
    ]);
    }
}
//servicio
if($paquete->servicios_count > 0) {
    $servicio = PAservicio::with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')-
>where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($servicio as $item) {
        // transporte
        if($item->tipo == '0') {
            $costos_nac_s = 0;
            $costos_ext_s = 0;
            $estatus = true;
            // Privado && Variable
            if($item->servicio_sel_tran->compartido == '0' && $item-
>servicio_sel_tran->tipotransporte->variable != '1') {
                if($item->vehiculo_id == null) {
                    $costos_nac_s = 0; $costos_ext_s = 0;
                } else {
                    foreach($item->servicio_sel_tran-
>precios as $index => $ytem) {
                        $split = explode(' | ', $ytem->nombre);
                        if($split[1] == $item-
>vehiculo_id && $split[0] == 'NACIONAL' && $estatus) {
```



```
                $costos_nac_s = $item->servicio_sel_tran-
>tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
            }
            if($split[1] == $item-
>vehiculo_id && $split[0] == 'EXTRANJERO' && $estatus) {
                $costos_ext_s = $item->servicio_sel_tran-
>tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
            }
        }
    }
} else {
    foreach($item->servicio_sel_tran-
>precios as $index => $ytem) {
        $split = explode(' | ', $ytem->nombre);
        if($split[0] == 'NACIONAL' && $estatus) {
            $costos_nac_s = $item->servicio_sel_tran-
>tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
        }
        if($split[0] == 'EXTRANJERO' && $estatus) {
            $costos_ext_s = $item->servicio_sel_tran-
>tipotransporte->variable == '1' ? $ytem->adulto : $ytem->autorizado;
            $estatus = FALSE;
        }
    }
}

$contador++;
$res_ser->push([
    'id' => $contador,
    'servicio_id' => $item->servicio_id,
    'compartido' => $item->servicio_sel_tran->compartido,
    'precios' => $item->servicio_sel_tran->precios,
    'nombre' => $item->servicio_sel_tran->generado,
```



```
        'dia' => $item->dia,
        'variable' => $item->servicio_sel_tran-
>tipotransporte->variable,
        'costos_nac' => $costos_nac_s,
        'costos_ext' => $costos_ext_s,
        'vehiculo_id' => $item->vehiculo_id ?? '0',
        'tipo' => $item->tipo,
    ]);
}
// varios
if($item->tipo == '1') {
    $costo_name = 0;
    foreach($item->servicio_sel_vari-
>precios as $index => $ytem) {
        if ($index == 0) {
            $costo_name = $ytem->adulto;
            break;
        }
    }

    $contador++;
    $res_ser->push([
        'id' => $contador,
        'servicio_id' => $item->servicio_id,
        'compartido' => '0',
        'precios' => $item->servicio_sel_vari->precios,
        'nombre' => $item->servicio_sel_vari->generado,
        'dia' => $item->dia,
        'variable' => '0',
        'costos_nac' => $costo_name,
        'costos_ext' => $costo_name,
        'tipo' => $item->tipo,
```



```
    ]);  
  }  
  // guiado  
  if($item->tipo == '3') {  
    $costo_name = 0;  
    foreach($item->servicio_sel_guia-  
>precios as $index => $ytem) {  
      if ($index == 0) {  
        $costo_name = $ytem->autorizado;  
        break;  
      }  
    }  
  }  
  
  $contador++;  
  $res_ser->push([  
    'id' => $contador,  
    'servicio_id' => $item->servicio_id,  
    'compartido' => $item->servicio_sel_guia->compartido,  
    'precios' => $item->servicio_sel_guia->precios,  
    'nombre' => $item->servicio_sel_guia->generado,  
    'dia' => $item->dia,  
    'variable' => '0',  
    'costos_nac' => $costo_name,  
    'costos_ext' => $costo_name,  
    'tipo' => $item->tipo,  
  ]);  
  }  
}  
}  
}  
  
//noches  
if($paquete->noches_count > 0) {
```



```

    $noches = PANOches::where('paquete_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());

    foreach($noches as $item) {
        $res_noc->push([
            'id' => $item->identificador,
            'cont' => $item->variable,
            'cant' => $item->cantidad,
        ]);
    }

    //dump($res_noc);
}

//descripcion
if($paquete->descripcion_count > 0) {
    $descripcion = PADescripcion::where('paquete_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
    foreach($descripcion as $item) {
        $res_des->push([
            'dia' => $item->dia,
            'descripcion' => $item->descripcion,
        ]);
    }
}

//adicional
if($paquete->adicional_count > 0) {
    $servicio = PAA adicional::with('servicio_sel_vari')-
>where('paquete_id', $id)->where('cuenta_id', $this->getCuenta()->get());
    foreach($servicio as $item ) {
        // varios
        $costo_name = 0;
        foreach($item->servicio_sel_vari-
>precios as $index => $ytem) {
            if ($index == 0) {
                $costo_name = $ytem->adulto;
            }
        }
    }
}

```



```
                break;
            }
        }

        $contador++;
        $res_upd->push([
            'id' => $contador,
            'servicio_id' => $item->servicio_id,
            'vehiculo_id' => '0',
            'compartido' => $item->servicio_sel_vari->compartido,
            'precios' => $item->servicio_sel_vari->precios,
            'nombre' => $item->servicio_sel_vari->generado,
            'dias' => explode(',',$item->dias),
            'variable' => '0',
            'costos_nac' => $costo_name,
            'costos_ext' => $costo_name,
            'tipo' => '1',
        ]);
    }
}

if($paquete)
    return view('administrar.paquete.armado_resumen')
        ->with('paquete', $paquete)->with('servicio', $servicio)-
>with('region', $region)
        ->with('capacidad', $capacidad)->with('contador', $contador)
        ->with('res_hos', $res_hos)->with('res_tic', $res_tic)
        ->with('res_exc', $res_exc)->with('res_ser', $res_ser)
        ->with('res_noc', $res_noc)->with('res_des', $res_des)
        ->with('res_upd', $res_upd)-
>with('tipotransporte', $tipotransporte)
```




```
        ->with('res_combinacion', $res_combinacion)-  
>with('res_destino', $res_destino);  
        else  
            return back();  
    }  
  
    protected function GenerarDestinos($primera_combinacion, $grupal)  
    {  
        $destino = collect();  
        foreach($primera_combinacion as $item) {  
            foreach($grupal as $data) {  
                if($item == $data["hospedaje_id"]) {  
                    $destino->push([  
                        'distrito_id' => $data["distrito_id"],  
                        'distrito_name' => $data["distrito_name"],  
                    ]);  
                }  
            }  
        }  
        return $destino;  
    }  
  
    public function imprimir(Request $request, $id)  
    {  
        $usuario = $request->user();  
        $ahora = now();  
        $totalnacional = 0; $totalextranjero = 0;  
  
        $modelo = PArmado::withCount('alojamiento')->withCount('ticket')-  
>withCount('noches')->withCount('descripcion')  
            ->withCount('excursiones')->withCount('servicios')-  
>withCount('adicional')->where('cuenta_id', $this->getCuenta())->find($id);
```



```
        $capacidad = SHospedajeServicio::select('capacidad')->where('cuenta_id', $this->getCuenta())->orderBy('capacidad')->groupBy('capacidad')->get();

        $margen = (1 + (floatval($modelo->cantidad) / 100));
        $incentivo = $modelo->incentivo ?? 10;

        # Variables Auxiliares
        $incluye = collect();
        $allservicio = collect();
        $itinerario = collect();
        $adicionales = collect();
        $allalojamiento = array();

        $allcombinacion = array();

        # Descripcion
        if($modelo->descripcion_count > 0) {
            $descripcion = PADescripcion::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
            $itinerario = $descripcion->sortBy('dia');
        }

        # Ticket
        if($modelo->ticket_count > 0) {
            $ticket = PATicket::with('ticket')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
            foreach($ticket as $item) {
                $costos_nac = 0; $costos_ext = 0;
                foreach ($item->ticket->precios as $data) {
                    if ($data->nombre == 'NACIONAL') {
                        $totalnacional += (floatval($data->adulto) / floatval($modelo->tipo_cambio ?? 3.10));
                        break;
                    }
                }
            }
        }
    }
}
```



```
    }
    foreach ($item->ticket->precios as $data) {
        if ($data->nombre == 'EXTRANJERO') {
            $totalextranjero += (floatval($data->adulto) / floatval($modelo->tipo_cambio ?? 3.10));
            break;
        }
    }
}
}

# Servicios
if($modelo->servicios_count > 0) {
    $servicio = PAservicio::with('servicio_sel_tran')->with('servicio_sel_vari')->with('servicio_sel_guia')->where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
    foreach($servicio as $item) {
        $nombre = ''; $variable = 0;
        if($item->tipo == '0') {
            $nombre = $item->servicio_sel_tran->nombre . ' | ' . ($item->servicio_sel_tran->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO') . ($item->vehiculo_id == NULL || $item->vehiculo_id == '0' ? '' : (' => ' . $item->vehiculo_id));
            if(!$item->servicio_sel_tran->tipotransporte->variable) {
                foreach($item->servicio_sel_tran->precios as $index => $ytem) {
                    if (strrpos($ytem->nombre, chr('NACIONAL')) >= 0 && strrpos($ytem->nombre, chr($item->vehiculo_id ?? '0')) >= 0) {
                        $totalnacional += floatval($ytem->autorizado);
                        break;
                    }
                }
            }
            foreach($item->servicio_sel_tran->precios as $index => $ytem) {
```



```
                if ( strrpos($ytem-
>nombre, chr('EXTRANJERO')) >= 0 && strrpos($ytem->nombre, chr($item-
>vehiculo_id ?? '0')) >= 0) {
                    $totalextranjero += floatval($ytem-
>autorizado);
                    break;
                }
            }
        } else {
            foreach($item->servicio_sel_tran-
>precios as $index => $ytem) {
                if ($ytem->nombre == 'NACIONAL') {
                    $totalnacional += floatval($ytem->adulto);
                    break;
                }
            }
            foreach($item->servicio_sel_tran-
>precios as $index => $ytem) {
                if ($ytem->nombre == 'EXTRANJERO') {
                    $totalextranjero += floatval($ytem->adulto);
                    break;
                }
            }
        }
    }
    if($item->tipo == '1') {
        $nombre = $item->servicio_sel_vari-
>nombre . ' | ' . $item->servicio_sel_vari->varios->nombre . ' | ' . ($item-
>servicio_sel_vari->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO');
        foreach($item->servicio_sel_vari-
>precios as $index => $ytem) {
            if ($index == 0) {
                $totalnacional += floatval($ytem-
>adulto == 0 ? 0 : $ytem->adulto);
```



```
                $totalextranjero += floatval($ytem-  
>adulto == 0 ? 0 : $ytem->adulto);  
                break;  
            }  
        }  
    }  
    if($item->tipo == '3') {  
        $nombre = $item->servicio_sel_guia->nombre .' | ' .($item->  
>servicio_sel_guia->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO');  
        foreach($item->servicio_sel_guia->  
>precios as $index => $ytem) {  
            if ($index == 0) {  
                $totalnacional += floatval($ytem->  
>autorizado == 0 ? 0 : $ytem->autorizado);  
                $totalextranjero += floatval($ytem->  
>autorizado == 0 ? 0 : $ytem->autorizado);  
                break;  
            }  
        }  
    }  
    $incluye->push([  
        'nombre' => $nombre,  
        'dia' => explode('_', $item->dia)[1],  
    ]);  
}  
# Excursiones  
if($modelo->excursiones_count > 0) {  
    $excursion = PAExcursion::with(['servicio.servicio_exc' => funcio  
n($query) {  
        $query->with('servicio_sel_tran.tipotransporte')->  
>with('servicio_sel_vari.varios')->with('servicio_sel_guia');  
    }->with(['excursion' => function($query) {
```



```
        $query->with('servicio')->with('endosar')-  
>withCount('endosar')->withCount('servicio');  
    ]])->where('paquete_id', $id)->where('cuenta_id', $this->  
>getCuenta())->get();  
    foreach($excursion as $item) {  
        if($item->excursion->endosar_count == 0) {  
            foreach($item->servicio as $data) {  
                if($data->servicio_exc) {  
                    if($data->servicio_exc->tipo == '0') {  
                        if($data->servicio_exc->servicio_sel_tran-  
>tipotransporte->variable) {  
                            foreach($item->servicio_exc->  
>servicio_sel_tran->precios as $index => $ytem) {  
                                if ($ytem->nombre == 'NACIONAL') {  
                                    $totalnacional += floatval($ytem->  
>adulto);  
                                    break;  
                                }  
                            }  
                            foreach($item->servicio_exc->  
>servicio_sel_tran->precios as $index => $ytem) {  
                                if ($ytem->nombre == 'EXTRANJERO') {  
                                    $totalextranjero += floatval($ytem->  
>adulto);  
                                    break;  
                                }  
                            }  
                        } else {  
                            foreach($data->servicio_exc->  
>servicio_sel_tran->precios as $index => $ytem) {  
                                if (strrpos($ytem->  
>nombre, chr('NACIONAL')) >= 0 && strrpos($ytem->nombre, chr($data->  
>vehiculo_id)) >= 0) {  
                                    $totalnacional += floatval($ytem->  
>autorizado);  
                                }  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
}
```



```
                break;
            }
        }
        foreach($data->servicio_exc-
>servicio_sel_tran->precios as $index => $ytem) {
            if ( strstrpos($ytem-
>nombre, chr('EXTRANJERO')) >= 0 && strstrpos($ytem->nombre, chr($data-
>vehiculo_id)) >= 0) {
                $totalextranjero += floatval($ytem
->autorizado);
                break;
            }
        }
    }
    if($data->servicio_exc->tipo == '1') {
        foreach($data->servicio_exc-
>servicio_sel_vari->precios as $index => $ytem) {
            if ($index == 0) {
                $totalnacional += ($ytem-
>adulto == 0 ? 0 : $ytem->adulto);
                $totalextranjero += $ytem-
>adulto == 0 ? 0 : $ytem->adulto;
                break;
            }
        }
    }
    if($data->servicio_exc->tipo == '3') {
        foreach($data->servicio_exc-
>servicio_sel_guia->precios as $index => $ytem) {
            if ($index == 0) {
                $totalnacional += $ytem-
>autorizado == 0 ? 0 : $ytem->autorizado;
```



```

                                $totalextranjero += $ytem-
>autorizado == 0 ? 0 : $ytem->autorizado;
                                break;
                                }
                                }
                                }
                                $incluye->push([
                                    'nombre' => $data->servicio_exc-
>nombre . ($data->vehiculo_id == NULL || $data-
>vehiculo_id == '0' ? '' : (' => ' . $data->vehiculo_id)),
                                    'dia' => explode('_', $item->dia)[1],
                                ]);
                                }
                                }
                                } else {
                                    foreach ($item->excursion->costos as $index => $data) {
                                        if ($index == 0) {
                                            $totalnacional += $data->nacional;
                                            $totalextranjero += $data->extranjero;
                                            break;
                                        }
                                    }
                                    $incluye->push([
                                        'nombre' => $item->excursion->nombre . ' | ENDOSADA',
                                        'dia' => explode('_', $item->dia)[1],
                                    ]);
                                }
                                }
                                }
                                }
                                # Adicionales
                                if($modelo->adicional_count > 0) {
                                    $adicional = PAAadicional::with('servicio_sel_vari')-
>where('paquete_id', $id)->get();

```




```
        foreach($adicional as $item) {
            $dias = count(explode(',',$item->dias));
            $nombre = $item->servicio_sel_vari->nombre . ' | ' . $item->servicio_sel_vari->varios->nombre . ' | ' . ($item->servicio_sel_vari->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO');
            foreach($item->servicio_sel_vari->precios as $index => $ytem) {
                if ($index == 0) {
                    $totalnacional += floatval($ytem->adulto == 0 ? 0 : $ytem->adulto * $dias);
                    $totalextranjero += floatval($ytem->adulto == 0 ? 0 : $ytem->adulto * $dias);
                    break;
                }
            }
            $adicionales->push([
                'nombre' => $nombre,
                'dias' => explode(',',$item->dias),
            ]);
        }
    }

    # Noches
    if($modelo->noches_count > 0) {
        $noches = PAnoches::where('paquete_id', $id)->where('cuenta_id', $this->getCuenta())->get();
        $selectnoche = collect();
        foreach($noches as $item) {
            $selectnoche->push([
                'id' => $item->identificador,
                'cont' => $item->variable,
                'cant' => $item->cantidad,
            ]);
        }
    }
}
```



```
}  
  
# Alojamiento  
  
if($modelo->alojamiento_count > 0) {  
    $combinacion = PACombinacion::where('paquete_id', $id)-  
>where('cuenta_id', $this->getCuenta()->get());  
  
    $hospedaje = PAHospedaje::with('hospedaje')-  
>where('paquete_id', $id)->where('cuenta_id', $this->getCuenta()->get());  
    if($hospedaje) {  
        $alojamientos = collect();  
        $contador = 0;  
        foreach($hospedaje as $item) {  
            $contador++;  
            $alojamientos->push([  
                'id' => $contador,  
                'hospedaje_id' => $item->proveedor_id,  
                'nombre' => $item->hospedaje->nombre . ' ('.$item->hospedaje->categoria.'*)',  
                'costos' => $item->hospedaje->precios,  
                'distrito_name' => $item->hospedaje->distrito->nombre,  
                'noches' => PANoches::select('identificador')-  
>where('identificador',$item->proveedor_id)->where('paquete_id', $id)-  
>where('cuenta_id', $this->getCuenta()->first()) ?? '0',  
            ]);  
        }  
        foreach($combinacion as $index => $item) {  
            $_nombres = array(); $_precios = array();  
            foreach(json_decode($item->combinacion, true) as $ytem) {  
                foreach($alojamientos as $ztem) {  
                    if($ztem["hospedaje_id"] == $ytem) {  
                        array_push($_nombres, $ztem["nombre"]);  
                        foreach($ztem["costos"] as $w => $wtem) {
```



```
                foreach($wtem as $x => $xtem) {
                    $montopasajero = ($w == 'nacional' ? $
totalnacional : $totalextranjero);
                    $nuevomonto = (isset($_precios[$w][$xtem["capacidad"]]) ? $_precios[$w][$xtem["capacidad"]] + floatval($xtem["monto"]) : floatval($xtem["monto"]) * (intval($ztem["noches"]) / intval($xtem["capacidad"])));
                    $_precios[$w][$xtem["capacidad"]] = $nuevomonto;
                }
            }
        }
    }
    array_push($allalojamiento, array('nombre' => implode(' | ', $_nombres), 'precios' => $_precios));
}
}
}
# Configuraciones de lo que incluye el servicio
if(count($incluye) > 0) {
    $sortservicio = $incluye->sortBy('dia');
    $allservicio = $sortservicio->groupBy('dia');
}
//return view('administrar.paquete.armado_imprimir', compact('usuario', 'ahora', 'modelo', 'itinerario', 'allservicio', 'capacidad', 'allalojamiento', 'totalnacional', 'totalextranjero', 'margen', 'adicionales', 'incetivo'));
$pdf = PDF::loadView('administrar.paquete.armado_imprimir', compact('usuario', 'ahora', 'modelo', 'itinerario', 'allservicio', 'capacidad', 'allalojamiento', 'totalnacional', 'totalextranjero', 'margen', 'adicionales', 'incetivo'));
return $pdf->download('Paquete_'.time().'.pdf');
}
}
```



➤ **ANEXO 08:** Código fuente – Controlador Cotización inicial

```
<?php

namespace App\Http\Controllers\Cotizacion\Cotizador;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\PArmado;
use App\SHospedajeServicio;
use App\Servicio;
use App\PAHospedaje;
use App\PATicket;
use App\PAExcursion;
use App\PAServicio;
use App\Operador;
use App\DFile;
use App\URegion;
use App\PANoches;
use App\PADescripcion;
use App\Nacionalidad;
use App\CArmado;
use Illuminate\Support\Facades\Mail;
use App\Mail\DocumentoEmail;
```



```
use App\PAAdicional;
use App\PACombinacion;
use App\TipoTransporte;

class AvanzadoController extends Controller
{
    public function __construct()
    {
        $this->middleware('login');
        $this->middleware('privilege:cotcotavanzada');
    }

    public function index(Request $request)
    {
        $servicio = Servicio::where('cuenta_id', $this->getCuenta())->get();

        if(trim(auth()->user()->tipo) == '0')
            $operador = Operador::where('cuenta_id', $this->getCuenta())->get();
        else
            $operador = Operador::where('cuenta_id', $this->getCuenta())->where('id', trim(auth()->user()->tipo))->get();

        $nacionalidad = Nacionalidad::all();
//        $region = URegion::all();
        $tipotransporte = TipoTransporte::select('id', 'nombre')->get();

        $noincluye_d = '<ul><li>Vuelos</li><li>Hotel</li><li>Propinas</li><li>Alimentación adicional no mencionada.</li></ul>';

        $recomendacion_d = '<ul><li>Documentos personales.</li><li>Mochila para Machu Picchu – Políticas de Traslado de Equipaje a Bordo del tren (Ruta Machu Picchu): Solo se permite 1 bolso o mochila de 08kg/12lb.</li><li>Zapatillas, pantalón ligero, polo o camisa, una gorra o un sombrero, bloqueador para el sol, repelente, poncho para la lluvia en temporada de lluvias.</li><li>Botella
```



```
de agua.</li><li>Si usted toma medicamentos llevar sus medicamentos.</li><li>Cámara de fotos y filmadora (cargador o pilas)</li></ul>';
```

```
    $politica_d = '<ul><li>Los servicios son confirmados al tener una respuesta afirmativa por parte de nuestras ejecutivas de venta.</li><li>Nuestras tarifas son expresadas en Dólares Americanos.</li><li>Las tarifas son netas y no comisionables.</li><li>Las tarifas no incluyen el 18% del IGV, (tener en cuenta para clientes peruanos).</li><li>Las tarifas para niños aplican de 04 a 09 años de edad cumplidos.</li><li>Las tarifas de estudiante aplican a personas de 10 a 17 años cumplidos.</li><li>Las tarifas para estudiantes universitarios aplican a personas con la edad máxima de 25 años cumplidos además de presentar el carnet universitario original y vigente.</li></ul>';
```

```
    $entrada = $request->all();

    $capacidad = SHospedajeServicio::select('capacidad')->where('cuenta_id', $this->getCuenta())->orderBy('capacidad')->groupBy('capacidad')->get();

    if(isset($entrada["seleccion"])) {
        $paquete = collect();
        $res_hos = collect(); $res_tic = collect(); $res_exc = collect();
        $res_ser = collect();

        $res_noc = collect(); $res_des = collect(); $res_upd = collect();

        $res_destino = collect(); $res_combinacion = collect();

        $contador = 0;
        //variacion de precios
        if($entrada["operador_id"] != '0') {
            $request->session()->put('cuenta_cambio', $entrada["operador_id"]);
        } else {
            $request->session()->forget('cuenta_cambio');
        }

        if($entrada["seleccion"] == 'paquete') {
            $paquete = PArmado::withCount('alojamiento')->withCount('ticket')->withCount('noches')->withCount('descripcion')
```



```
        ->withCount('excursiones')->withCount('servicios')-  
>withCount('adicional')  
        ->with('servicio')->where('cuenta_id', $this->getCuenta())-  
>find($entrada["identificador"]);  
  
    // VERIFICAR PROPIO  
    if($entrada["operador_id"] != '0') {  
        $clientep = Operador::find($entrada["operador_id"]);  
        if($clientep->propio != '1') {  
            $paquete->cantidad = 0; $paquete->incentivo = 0;  
        }  
    }  
  
    //ticket  
    if($paquete->ticket_count > 0) {  
        $ticket = PATicket::with('ticket')-  
>where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())-  
>get();  
  
        foreach($ticket as $item) {  
            $costos_nac = 0; $costos_ext = 0;  
            $contador++;  
            $res_tic->push([  
                'id' => $contador,  
                'boleto_id' => $item->ticket_id,  
                'nombre' => $item->ticket->descripcion,  
                'dolares' => $item->ticket->dolares ?? '0',  
                'detalle' => $item->ticket->zonas_name,  
                'costos_nac' => number_format($costos_nac, 2, '.',  
''),  
                'costos_ext' => number_format($costos_ext, 2, '.',  
''),  
                'precios' => $item->ticket->precios,  
            ]);  
        }  
    }
```



```
    }
    //excursion
    if($paquete->excursiones_count > 0) {
        $excursion = PAExcursion::with(['servicio.servicio_exc' =>
function($query) {
            $query->with('servicio_sel_tran.tipotransporte')->
>with('servicio_sel_vari.varios')->with('servicio_sel_guia');
            }])->with(['excursion' => function($query) {
                $query->with('servicio')->with('endosar')->
>withCount('endosar')->withCount('servicio');
            }])->where('paquete_id', $paquete->id)->
>where('cuenta_id', $this->getCuenta())->get();
        foreach($excursion as $item) {
            $costos_nac = 0; $costos_ext = 0;
            foreach ($item->excursion-
>costos as $index => $data) {
                if ($index == 0) {
                    $costos_nac = $data->nacional;
                    $costos_ext = $data->extranjero;
                    break;
                }
            }
        }
    }
    //servicios
    $res_exc_ser = collect();
    foreach($item->servicio as $data) {
        if($data->servicio_exc) {
            $data->servicio_exc->vehiculo_id = $data->
>vehiculo_id ?? '0';
            if($data->servicio_exc->tipo == '0') {
                if(!$data->servicio_exc->
>servicio_sel_tran->tipotransporte->variable) {
                    foreach($data->servicio_exc->
>servicio_sel_tran->precios as $index => $ytem) {
```




```

                                if (strrpos($ytem-
>nombre, 'NACIONAL') >= 0 && strrpos($ytem->nombre, $item-
>vehiculo_id ?? '0') >= 0) {
                                $data->servicio_exc-
>costo_nac = floatval($ytem->autorizado);
                                break;
                                }
                                }
                                foreach($data->servicio_exc-
>servicio_sel_tran->precios as $index => $ytem) {
                                if ( strrpos($ytem-
>nombre, 'EXTRANJERO') >= 0 && strrpos($ytem->nombre, $item-
>vehiculo_id ?? '0') >= 0) {
                                $data->servicio_exc-
>costo_ext = floatval($ytem->autorizado);
                                break;
                                }
                                }
                                }
                                }
                                }

                                $res_exc_ser->push($data->servicio_exc);
                                }
                                }
                                $contador++;
                                $res_exc->push([
                                    'id' => $contador,
                                    'excursion_id' => $item->excursion_id,
                                    'nombre' => $item->excursion->nombre,
                                    'costos_nac' => $item->excursion-
>endosar_count != '0' ? number_format($costos_nac, 2, '.', '') : $item-
>excursion->precios,
                                    'costos_ext' => $item->excursion-
>endosar_count != '0' ? number_format($costos_ext, 2, '.', '') : $item-
>excursion->precios,
```



```
        'endosado' => $item->excursion->endosar_count,  
        'costos' => $item->excursion->costos,  
        'servicios' => $res_exc_ser,  
        'dia' => $item->dia,  
    ]);  
    }  
}  
  
//servicio  
if($paquete->servicios_count > 0) {  
    $servicio = PAservicio::with('servicio_sel_tran')->  
>with('servicio_sel_vari')->with('servicio_sel_guia')  
->where('paquete_id', $paquete->id)-  
>where('cuenta_id', $this->getCuenta())->get();  
    foreach($servicio as $item) {  
        // transporte  
        if($item->tipo == '0') {  
            $costos_nac_s = 0;  
            $costos_ext_s = 0;  
            $estatus = true;  
            // Privado && Variable  
            if($item->servicio_sel_tran-  
>compartido == '0' && $item->servicio_sel_tran->tipotransporte-  
>variable != '1') {  
                if($item->vehiculo_id == null) {  
                    $costos_nac_s = 0; $costos_ext_s = 0;  
                } else {  
                    foreach($item->servicio_sel_tran-  
>precios as $index => $ytem) {  
                        $split = explode(' | ', $ytem-  
>nombre);  
                        if($split[1] == $item-  
>vehiculo_id && $split[0] == 'NACIONAL' && $estatus) {
```



```

                                $costos_nac_s = $item-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;

                                }

                                if($split[1] == $item-
>vehiculo_id && $split[0] == 'EXTRANJERO' && $estatus) {

                                $costos_ext_s = $item-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;

                                }

                                }

                                }

                                } else {

                                foreach($item->servicio_sel_tran-
>precios as $index => $ytem) {

                                $split = explode(' | ', $ytem->nombre);
                                if($split[0] == 'NACIONAL' && $estatus) {

                                $costos_nac_s = $item-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;

                                }

                                if($split[0] == 'EXTRANJERO' && $estatus)
{

                                $costos_ext_s = $item-
>servicio_sel_tran->tipotransporte->variable == '1' ? $ytem->adulto : $ytem-
>autorizado;

                                $estatus = FALSE;

                                }

                                }

                                }

                                $contador++;

                                $res_ser->push([

                                'id' => $contador,
```



```
        'servicio_id' => $item->servicio_id,  
        'compartido' => $item->servicio_sel_tran-  
>compartido,  
        'precios' => $item->servicio_sel_tran-  
>precios,  
        'nombre' => $item->servicio_sel_tran-  
>nombre .' | '.($item->servicio_sel_tran-  
>compartido == '0' ? 'PRIVADO' : 'COMPARTIDO'),  
        'dia' => $item->dia,  
        'variable' => $item->servicio_sel_tran-  
>tipotransporte->variable,  
        'costos_nac' => $costos_nac_s,  
        'costos_ext' => $costos_ext_s,  
        'vehiculo_id' => $item->vehiculo_id ?? '0',  
        'tipo' => $item->tipo,  
    ]);  
    }  
    // varios  
    if($item->tipo == '1') {  
        $costo_name = 0;  
        foreach($item->servicio_sel_vari-  
>precios as $index => $ytem) {  
            if ($index == 0) {  
                $costo_name = $ytem->adulto;  
                break;  
            }  
        }  
        $contador++;  
        $res_ser->push([  
            'id' => $contador,  
            'servicio_id' => $item->servicio_id,  
            'compartido' => '0',
```



```
        'precios' => $item->servicio_sel_vari-
>precios,
        'nombre' => $item->servicio_sel_vari-
>nombre .' | ' .($item->servicio_sel_vari->varios->nombre .' | ' .($item-
>servicio_sel_vari->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO'),
        'dia' => $item->dia,
        'variable' => '0',
        'costos_nac' => $costo_name,
        'costos_ext' => $costo_name,
        'tipo' => $item->tipo,
    ]);
}
// guiado
if($item->tipo == '3') {
    $costo_name = 0;
    foreach($item->servicio_sel_guia-
>precios as $index => $ytem) {
        if ($index == 0) {
            $costo_name = $ytem->autorizado;
            break;
        }
    }

    $contador++;
    $res_ser->push([
        'id' => $contador,
        'servicio_id' => $item->servicio_id,
        'compartido' => $item->servicio_sel_guia-
>compartido,
        'precios' => $item->servicio_sel_guia-
>precios,
        'nombre' => $item->servicio_sel_guia-
>nombre .' | ' .($item->servicio_sel_guia-
>compartido == '0' ? 'PRIVADO' : 'COMPARTIDO'),
```



```
        'dia' => $item->dia,  
        'variable' => '0',  
        'costos_nac' => $costo_name,  
        'costos_ext' => $costo_name,  
        'tipo' => $item->tipo,  
    ]);  
    }  
}  
  
// noches hospedajes  
if($entrada["tipo"] == 'true') {  
    if($paquete->alojamiento_count > 0) {  
        $hospedaje = PAHospedaje::with('hospedaje')-  
>where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())-  
>get();  
  
        foreach($hospedaje as $item) {  
            $contador++;  
            $res_hos->push([  
                'id' => $contador,  
                'hospedaje_id' => $item->proveedor_id,  
                'nombre' => $item->hospedaje->nombre,  
                'costos' => $item->hospedaje->precios,  
                'distrito_id' => $item->hospedaje-  
>distrito_id,  
                'distrito_name' => $item->hospedaje-  
>ubicacion->generado,  
                'categoria' => $item->hospedaje->categoria,  
                'checkin' => $item->hospedaje-  
>checkin ?? 'No Especifica',  
                'checkout' => $item->hospedaje-  
>checkout ?? 'No Especifica',  
            ]);  
        }  
    }  
}
```



```
        $combinacion = PACombinacion::where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())->get();

        foreach($combinacion as $index => $item) {
            if($index == 0) $res_destino = $this->GenerarDestinos(json_decode($item->combinacion, true), $res_hos);
            $res_combinacion->push(json_decode($item->combinacion, true));
        }
    }

    if($paquete->noches_count > 0) {
        $noches = PANOches::where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())->get();

        foreach($noches as $item) {
            $res_noc->push([
                'id' => $item->identificador,
                'cont' => $item->variable,
                'cant' => $item->cantidad,
            ]);
        }
    }

    //descripcion
    if($paquete->descripcion_count > 0) {
        $descripcion = PADescripcion::where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())->get();

        foreach($descripcion as $item) {
            $res_des->push([
                'dia' => $item->dia,
                'descripcion' => $item->descripcion,
            ]);
        }
    }

    //adicional
```



```
        if($paquete->adicional_count > 0) {
            $servicio = PAA adicional::with('servicio_sel_vari')-
>where('paquete_id', $paquete->id)->where('cuenta_id', $this->getCuenta())-
>get();

            foreach($servicio as $item) {
                // varios
                $costo_name = 0;
                foreach($item->servicio_sel_vari-
>precios as $index => $ytem) {
                    if ($index == 0) {
                        $costo_name = $ytem->adulto;
                        break;
                    }
                }

                $contador++;
                $res_upd->push([
                    'id' => $contador,
                    'servicio_id' => $item->servicio_id,
                    'vehiculo_id' => '0',
                    'compartido' => $item->servicio_sel_vari-
>compartido,
                    'precios' => $item->servicio_sel_vari->precios,
                    'nombre' => $item->servicio_sel_vari-
>nombre . ' | ' . $item->servicio_sel_vari->varios->nombre . ' | ' . ($item-
>servicio_sel_vari->compartido == '0' ? 'PRIVADO' : 'COMPARTIDO'),
                    'dias' => explode(', ', $item->dias),
                    'variable' => '0',
                    'costos_nac' => $costo_name,
                    'costos_ext' => $costo_name,
                    'tipo' => '1',
                ]);
            }
        }
```




```
    }  
    //dd($res_des);  
  }  
  
  return redirect()-  
>action('Cotizacion\Cotizador\AvanzadoController@index')  
  ->with('paquete', $paquete)->with('res_hos', $res_hos)-  
>with('res_tic', $res_tic)  
  ->with('res_exc', $res_exc)->with('res_ser', $res_ser)-  
>with('contador', $contador)  
  ->with('res_noc', $res_noc)->with('res_des', $res_des)-  
>with('res_upd', $res_upd)  
  ->with('res_combinacion', $res_combinacion)-  
>with('res_destino', $res_destino)  
  ->withInput($request->all());  
  }  
  
  return view('cotizacion.cotizador.avanzado')-  
>with('noincluye_d', $noincluye_d)->with('recomendacion_d', $recomendacion_d)  
  ->with('politica_d', $politica_d)-  
>with('tipotransporte', $tipotransporte)->with('capacidad', $capacidad)  
  ->with('servicio', $servicio)->with('operador', $operador)-  
>with('nacionalidad', $nacionalidad);  
  }  
  
  protected function GenerarDestinos($primera_combinacion, $grupal)  
  {  
    $destino = collect();  
    foreach($primera_combinacion as $item) {  
      foreach($grupal as $data) {  
        if($item == $data["hospedaje_id"]) {  
          $destino->push([  
            'distrito_id' => $data["distrito_id"],  
            'distrito_name' => $data["distrito_name"],  
          ]);  
        }  
      }  
    }  
  }  
}
```



```
        }
    }
}
return $destino;
}

public function email(Request $request, $id, $tipo, $email)
{
    try {
        $modelo = $tipo == '0' ? CArmado::where('cuenta_id', $this->getCuenta())->find($id) : DFile::where('cuenta_id', $this->getCuenta())->find($id);

        // Mail::to($email)->send(new DocumentoEmail($modelo));

        $message = array('tipo' => 'info', 'mensaje' => 'Enviando Documento...');
    } catch (\Exception $e) {
        $message = array('tipo' => 'danger', 'mensaje'=> $e->getMessage());
    }

    return $message;
}
}
```

➤ ANEXO 09: Código fuente – Controlador Cotización confirmada

```
<?php

namespace App\Http\Controllers\Cotizacion\Cotizador;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\DFile;
use App\SHospedajeServicio;
use App\Nacionalidad;
```



```
use App\URegion;
use App\DDescripcion;
use App\DTicket;
use App\DServicio;
use App\DExcursion;
use App\DAadicional;
use App\DOTros;
use App\DItinero;
use App\DLiberado;
use App\DNoches;
use App\DHospedaje;
use App\PHospedaje;
use Illuminate\Support\Facades\DB;
use App\TipoTransporte;
use Dompdf\Dompdf as PDF;
use App\DNotas;
use App\MetodoPago;
use App\Operador;
use App\VFiles;

class FilesController extends Controller
{
    private $PAPER_SIZES = array(
        "4a0" => array(0, 0, 4767.87, 6740.79),
        "2a0" => array(0, 0, 3370.39, 4767.87),
        "a0" => array(0, 0, 2383.94, 3370.39),
        "a1" => array(0, 0, 1683.78, 2383.94),
        "a2" => array(0, 0, 1190.55, 1683.78),
        "a3" => array(0, 0, 841.89, 1190.55),
        "a4" => array(0, 0, 595.28, 841.89),
        "a5" => array(0, 0, 419.53, 595.28),
        "a6" => array(0, 0, 297.64, 419.53),
```



```
"a7" => array(0, 0, 209.76, 297.64),  
"a8" => array(0, 0, 147.40, 209.76),  
"a9" => array(0, 0, 104.88, 147.40),  
"a10" => array(0, 0, 73.70, 104.88),  
"b0" => array(0, 0, 2834.65, 4008.19),  
"b1" => array(0, 0, 2004.09, 2834.65),  
"b2" => array(0, 0, 1417.32, 2004.09),  
"b3" => array(0, 0, 1000.63, 1417.32),  
"b4" => array(0, 0, 708.66, 1000.63),  
"b5" => array(0, 0, 498.90, 708.66),  
"b6" => array(0, 0, 354.33, 498.90),  
"b7" => array(0, 0, 249.45, 354.33),  
"b8" => array(0, 0, 175.75, 249.45),  
"b9" => array(0, 0, 124.72, 175.75),  
"b10" => array(0, 0, 87.87, 124.72),  
"c0" => array(0, 0, 2599.37, 3676.54),  
"c1" => array(0, 0, 1836.85, 2599.37),  
"c2" => array(0, 0, 1298.27, 1836.85),  
"c3" => array(0, 0, 918.43, 1298.27),  
"c4" => array(0, 0, 649.13, 918.43),  
"c5" => array(0, 0, 459.21, 649.13),  
"c6" => array(0, 0, 323.15, 459.21),  
"c7" => array(0, 0, 229.61, 323.15),  
"c8" => array(0, 0, 161.57, 229.61),  
"c9" => array(0, 0, 113.39, 161.57),  
"c10" => array(0, 0, 79.37, 113.39),  
"ra0" => array(0, 0, 2437.80, 3458.27),  
"ra1" => array(0, 0, 1729.13, 2437.80),  
"ra2" => array(0, 0, 1218.90, 1729.13),  
"ra3" => array(0, 0, 864.57, 1218.90),  
"ra4" => array(0, 0, 609.45, 864.57),  
"sra0" => array(0, 0, 2551.18, 3628.35),
```



```
"sra1" => array(0, 0, 1814.17, 2551.18),
"sra2" => array(0, 0, 1275.59, 1814.17),
"sra3" => array(0, 0, 907.09, 1275.59),
"sra4" => array(0, 0, 637.80, 907.09),
"letter" => array(0, 0, 612.00, 792.00),
"legal" => array(0, 0, 612.00, 1008.00),
"ledger" => array(0, 0, 1224.00, 792.00),
"tabloid" => array(0, 0, 792.00, 1224.00),
"executive" => array(0, 0, 521.86, 756.00),
"folio" => array(0, 0, 612.00, 936.00),
"commercial #10 envelope" => array(0, 0, 684, 297),
"catalog #10 1/2 envelope" => array(0, 0, 648, 864),
"8.5x11" => array(0, 0, 612.00, 792.00),
"8.5x14" => array(0, 0, 612.00, 1008.0),
"11x17" => array(0, 0, 792.00, 1224.00),
"ticket" => array(0, 0, 280.63, 595.27),
);

public function __construct()
{
    $this->middleware('login');
    //$this->middleware('privilege:cotcotfiles');
}

public function index(Request $request)
{
    $metodos = MetodoPago::all();
    return view('cotizacion.cotizador.misfiles')-
>with('metodos', $metodos);
}

public function resumen(Request $request, $id)
```



```
{
    $nacionalidad = Nacionalidad::all();
    $capacidad = SHospedajeServicio::select('capacidad')-
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-
>groupBy('capacidad')->get();
    $tipotransporte = TipoTransporte::all();

    $serviciosver = collect(); $excursionever = collect();

    $file = DFile::withCount('integrantes')->where('cuenta_id', $this-
>getCuenta())->findOrFail($id);
    $notas = DNotas::where('file_id', $file->id)-
>where('cuenta_id', $this->getCuenta())->get();

    if($file->operador_id != '0') {
        $request->session()->put('cuenta_cambio', $file->operador_id);
    } else {
        $request->session()->forget('cuenta_cambio');
    }
    # Noches # Alojamiento
    $alojamiento = collect(); $auxiliar_a = collect();
    $noches = DNoches::where('file_id', $id)->where('cuenta_id', $this-
>getCuenta())->get();
    $hospedajes = DHospedaje::with('detalle')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();
    foreach($hospedajes as $hosp) {
        foreach($hosp->detalle as $detalle) {
            if($detalle->cantidad != '0') {
                $auxiliar_a->push(
                    (object) [
                        'hospedaje_id' => $detalle->hospedaje_id,
                        'programado' => $detalle->programado,
                        'codigo_reserva' => $detalle->cod_reserva,
```



```
        'nacionalidad_tipo' => $detalle->nacionalidad_tipo,
    >nacionalidad_tipo,
        'cantidad' => $detalle->cantidad ?? 0,
        'tipo_servicio' => $detalle->tipo_servicio ?? 0,
        'precio' => $detalle->precio,
    ]
    );
}

}

}
$grouped = $auxiliar_a->groupBy('hospedaje_id');
$contador = 0;
$ingreso_a = \Carbon\Carbon::parse($file->fecha_inicio);
foreach($grouped as $hospedaje_id => $group) {
    $noche_alo = count($noches) > 0 ? $noches[$contador]->cantidad : 0;
    $nacionalidad_group = $group->groupBy('nacionalidad_tipo');
    $nacionalidad_tipo = collect();
    $programado = ''; $codigo_reserva = '';
    foreach($nacionalidad_group as $nac_tipo => $tipo) {
        $suma_collect = collect();
        foreach($tipo as $data) {
            $programado = $data->programado; $codigo_reserva = $data->
            >codigo_reserva;
            $suma_collect->push($data->cantidad*$data->
            >tipo_servicio*$data->precio);
        }
        $nacionalidad_tipo->push( (object)[
            'nacionalidad_tipo' => $nac_tipo,
            'precio' => $suma_collect->sum()
        ] );
    }
}
```



```
$alojamiento_s = PHospedaje::find($hospedaje_id);  
if($alojamiento_s) {  
    $alojamiento->push(  
        (object) [  
            'hospedaje_id' => $hospedaje_id,  
            'programado' => $programado,  
            'codigo_reserva' => $codigo_reserva,  
            'noches' => $noche_alo,  
            "ingreso" => $ingreso_a->format('d/m/Y'),  
            "in" => $alojamiento_s->checkin ?? '-',  
            "out" => $alojamiento_s->checkout ?? '-',  
            'hospedaje_name' => '<h4 class="no-margin semi-  
bold text-uppercase" id="alojamiento_name_'.$hospedaje_id.'"><i class="fa fa-  
bed m-r-5"></i>'.$alojamiento_s->nombre.'<u class="ml-1 fs-  
14">'.$alojamiento_s->distrito->nombre .'</u><small class="ml-2 fs-  
13">'.str_repeat('<i class="fa fa-star text-warning"></i>', $alojamiento_s->  
>categoria).'</small>'.</h4>',  
            'detalle' => $nacionalidad_tipo  
        ]  
    );  
}  
$contador++;  
}  
  
# Boletos  
$boletosver = collect();  
$ticket = DTicket::where('file_id', $id)->where('cuenta_id', $this->  
>getCuenta())->get();  
foreach($ticket as $item) {  
    if($item->padre_id == NULL) {  
        $boletosver->push([  
            'boleto_id' => $item->ticket_id,  
        ]);  
    }  
}
```




```
    }  
  }  
  # Servicios, Excursiones, Upgrades y Otros  
  $collect_por_dia = collect(); $collect_variable = collect(); $collect_alojamiento = collect();  
  $servicio = DServicio::with('detalle')->whereNull('doc_excursion_id')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->where('dia', 'like', 'dia_%')->get();  
  foreach($servicio as $item) {  
    $variable = TRUE;  
    if($item->tipo == '0') {  
      if($item->servicio_sel_tran->tipotransporte->variable == '1') $variable = FALSE;  
    }  
    if($variable) {  
      $collect_por_dia->push([  
        'id' => $item->id,  
        'color' => 'text-primary',  
        'tipo' => 'SERV',  
        'nombre' => $item->generado,  
        'precio' => $item->precios,  
        'free' => $item->free,  
        'vehiculo' => $item->vehiculo_id,  
        'grupo' => collect(),  
        'dia' => explode("_", $item->dia)[1]  
      ]);  
    } else {  
      # verificar itinerario  
      $itinerario = DItinerario::where('servicio_id', $item->servicio_id)->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->first();  
      $data_iti = collect();  
    }  
  }  
}
```



```
        if($itinerario) {
            $data_iti->push([
                'proveedor' => $itinerario->proveedor,
                'ruta' => $itinerario->origen.'-'. $itinerario-
>destino,
                'tipo' => $itinerario->tipo == '11' ? 'IDA-
VUELTA' : ($itinerario->tipo == '10' ? 'IDA' : 'VUELTA'),
                'detalle' => $itinerario->generado,
            ]);
        }
        $collect_variable->push([
            'id' => $item->id,
            'servicio_id' => $item->servicio_id,
            'tipo' => 'SERV',
            'nombre' => $item->generado,
            'precio' => $item->precios,
            'telefono' => '0000',
            'por' => 'PROPIO',
            'direccion' => NULL,
            'icono' => $item->servicio_sel_tran->tipotransporte-
>icono,
            'itinerario' => $data_iti,
            'programado' => $item->programado,
            'codigo_reserva' => $item->codigo_reserva,
            'dia' => explode("_", $item->dia)[1]
        ]);
    }
    $serviciosver->push([
        'servicio_id' => $item->servicio_id,
        'tipo' => $item->tipo,
    ]);
}
```



```
$excursion = DExcursion::with('excursion')->with('servicio')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($excursion as $item) {
    if($item->endosado == '0') {
        if(count($item->servicio) > 0) {
            $collect_servicios_exc = collect();
            foreach($item->servicio as $ytem) {
                $collect_servicios_exc->push([
                    'id' => $ytem->id,
                    'tipo' => 'SERV',
                    'nombre' => $ytem->generado,
                    'precio' => $ytem->precios,
                    'free' => $ytem->free,
                    'vehiculo' => $ytem->vehiculo_id,
                ]);
                $serviciosver->push([
                    'servicio_id' => $ytem->servicio_id,
                    'tipo' => $ytem->tipo,
                ]);
            }
            $collect_por_dia->push([
                'id' => $item->id,
                'color' => 'text-complete',
                'tipo' => 'EXCU',
                'nombre' => $item->generado,
                'precio' => $item->precios,
                'tipo' => '10',
                'grupo' => $collect_servicios_exc,
                'dia' => explode("_", $item->dia)[1]
            ]);
        }
    } else {
```



```
$collect_por_dia->push([
    'id' => $item->id,
    'color' => 'text-complete',
    'tipo' => 'EXCU',
    'nombre' => $item->generado,
    'precio' => $item->precios,
    'free' => '0',
    'vehiculo' => NULL,
    'grupo' => collect(),
    'dia' => explode("_", $item->dia)[1]
]);
}
$excursionever->push([
    'excursion_id' => $item->excursion_id,
]);
}
$upgrade = DAdicional::where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();
foreach($upgrade as $item) {
    $dias = explode(',',$item->dia);
    for ($i=0; $i < count($dias); $i++) {
        $collect_por_dia->push([
            'id' => $item->id.'-'. $dias[$i],
            'color' => 'text-warning',
            'tipo' => 'UPDA',
            'nombre' => $item->generado,
            'free' => '0',
            'vehiculo' => NULL,
            'precio' => number_format(floatval($item->
precios)/count($dias), 2, '.', ''),
            'grupo' => collect(),
            'dia' => $dias[$i]
```



```
    ]);
  }
}
$otros = DOtros::with('detalle')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();
foreach($otros as $item) {
    //dump($item->dia != '' ? $item->dia : '0');
    $dias = explode(',',$item->dia != '' ? $item->dia : '0');
    for ($i=0; $i < count($dias); $i++) {
        $collect_por_dia->push([
            'id' => $item->id.'-'. $dias[$i],
            'color' => 'text-info',
            'tipo' => 'OTRO',
            'nombre' => $item->nombre,
            'free' => '0',
            'vehiculo' => NULL,
            'precio' => number_format(floatval($item-
>precios)/count($dias), 2, '.', ''),
            'grupo' => collect(),
            'dia' => $dias[$i]
        ]);
    }
}
//dd('sad');
# Itinerarios
$itinerarios = DItinerario::whereNull('servicio_id')-
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($itinerarios as $item) {
    $servicio_name = explode('*', $item->servicio_name);
    $data_iti = collect();
    if($servicio_name[0] == 'ALOJ') {
        $collect_alojamiento->push(
```



```
(object)[
  'id' => $item->id,
  'tipo' => 'ITI_ALO',
  'nombre' => $servicio_name[2],
  'direccion' => $item->direccion,
  'telefono' => $item->telefono ?? '0000',
  'por' => $item->por ?? 'PROPIO',
  'icono' => $servicio_name[1],
  'proveedor' => $item->proveedor,
  'ubicacion' => $item->origen,
  'detalle' => $item->generado,
];
} elseif($servicio_name[0] == 'SERV') {
  $data_iti->push([
    'proveedor' => $item->proveedor,
    'ruta' => $item->tipo == '11' ? ($item->origen.'-'. $item->destino.'-'. $item->origen) : ($item->origen.'-'. $item->destino),
    'tipo' => $item->tipo == '11' ? 'IDA-VUELTA' : ($item->tipo == '10' ? 'IDA' : 'VUELTA'),
    'detalle' => $item->generado,
  ]);
  $collect_variable->push([
    'id' => $item->id,
    'servicio_id' => 0,
    'tipo' => 'ITI_SERV',
    'nombre' => $servicio_name[2],
    'telefono' => $item->telefono ?? '0000',
    'por' => $item->por ?? 'PROPIO',
    'precio' => 0,
    'direccion' => $item->direccion,
    'icono' => $servicio_name[1],
    'itinerario' => $data_iti,
```



```
        'programado' => NULL,  
        'dia' => 0  
    ]]);  
    }  
}  
  
# Liberados  
$liberado = DLiberado::where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
return view('cotizacion.cotizador.files_resumen', compact('file', 'nacionalidad', 'alojamiento', 'ticket',  
    'collect_por_dia', 'collect_variable', 'tipotransporte', 'collect_alojamiento', 'notas', 'boletosver', 'serviciosver', 'excursionever', 'liberado'))  
;  
}  
  
public function imprimir(Request $request, $id, $tipo)  
{  
    $entrada = $request->all();  
  
    $usuario = $request->user();  
    $ahora = now();  
    $capacidad = SHospedajeServicio::select('capacidad')-  
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-  
>groupBy('capacidad')->get();  
    $file = DFile::with('integrantes')->with('boletos_contratados')-  
>withCount('boletos_contratados')->with('adicional_contratados')-  
>withCount('adicional_contratados')->with('otro_contratados')-  
>withCount('otro_contratados')->where('cuenta_id', $this->getCuenta())-  
>findOrFail($id);  
  
    # Notas  
    $notas = DNotas::where('tipo', 'file')->where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
    # incluye
```



```
$liquidar = collect();

$incluye = collect(); $trenes = collect(); $vuelos_buses = collect();
$alojamiento = collect();

$excursion = DExcursion::with('excursion')-
>with(['servicio' => function($query) { $query->with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle'); }])-
>with('detalle')->where('file_id', $id)->where('cuenta_id', $this-
>getCuenta());

    if(isset($entrada["endose"])) {
        $excursion = $excursion-
>whereIn('id', json_decode($entrada["endose"]));
    }

    foreach ($excursion->get() as $item) {
        if($item->endosado == '0') {
            foreach ($item->servicio as $data) {
                $incluye->push($data);
                $facturado = '0';
                if($data->tipo == '0') {
                    $facturado = $data->servicio_sel_tran->tipotransporte-
>variable;
                }
                $liquidar->push((object) [
                    "nombre" => $data->generado,
                    "group" => "SERVICIOS",
                    "facturado" => $facturado,
                    "cantidad" => $this->CantidadPersonas($data->detalle),
                    "dolares" => $data->free == '0' ? floatval($data-
>precios) : 0,
                    "soles" => 0,
                ]);
            }
        } else {
            $incluye->push($item);
            $liquidar->push((object) [
```




```
        "nombre" => $item->generado,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
    }
}

$servicio = DServicio::with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle')-
>whereNull('doc_excursion_id')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->where('dia', 'like', 'dia%');

    if(isset($entrada["servicio"])) {
        $servicio = $servicio-
>whereIn('id', json_decode($entrada["servicio"]));
    }

    foreach($servicio->get() as $item) {
        $variable = TRUE;
        $facturado = '0';
        if($item->tipo == '0') {
            $facturado = $item->servicio_sel_tran->tipotransporte-
>variable;

            if($item->servicio_sel_tran->tipotransporte-
>variable == '1') $variable = FALSE;
        }
        if(!$variable) {
            if($item->itinerario) {
                if($item->servicio_sel_tran->tipotransporte-
>nombre == 'FERROVIARIO') {
                    # trenes
                    foreach($item->itinerario->detalle as $ytem) {
                        $trenes->push([
                            "empresa" => $item->itinerario->proveedor,
```



```
        "servicio" => $item->servicio_sel_tran-
>nombre . ' # ' . $ytem->numero,
        "fecha" => $ytem->fecha->format('d/m/Y'),
        "salida" => $ytem->salida->format('H:i'),
        "llegada" => $ytem->llegada->format('H:i'),
        "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
        "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
    });
    }

    } else {
        # vuelos / buses
        foreach($item->itinerario->detalle as $ytem) {
            $vuelos_buses->push([
                "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
                "compania" => $item->itinerario->proveedor,
                "numero" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
            ]);
        }
    }
}
$incluye->push($item);
$liquidar->push((object) [
    "nombre" => $item->generado,
```



```
        "group" => "SERVICIOS",
        "facturado" => $facturado,
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => $item->free == '0' ? floatval($item->precios) : 0,
        "soles" => 0,
    ]);
}
foreach($file->adicional_contratados as $item) {
    $liquidar->push((object) [
        "nombre" => $item->generado,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
foreach($file->otro_contratados as $item) {
    $liquidar->push((object) [
        "nombre" => $item->nombre,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
foreach($file->boletos_contratados as $item) {
    # dolares = 1 -> SI | dolares = 0 -> NO
    if($item->padre_id == NULL) {
        $liquidar->push((object) [
```



```
        "nombre" => $item->ticket->descripcion,
        "group" => "BOLETOS",
        "facturado" => '1',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => $item->ticket-
>dolares == '1' ? floatval($item->precios) : 0,
        "soles" => $item->ticket->dolares == '0' ? floatval($item-
>precios) * floatval($file->tipo_cambio) : 0 ,
    ]);
    }
}

$sortservicio = $incluye->sortBy('dia');
$allservicio = $sortservicio->groupBy('dia');

$facturado = $liquidar->groupBy('group');
# itinerarios
$itinerarios = DItinerario::with('detalle')->whereNull('servicio_id')-
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($itinerarios as $item) {
    $servicio_name = explode('*', $item->servicio_name);
    if($servicio_name[0] == 'ALOJ') {
        $alojamiento->push([
            "hotel" => $item->proveedor,
            "direccion" => $item->direccion,
            "telefono" => $item->telefono ?? '0000',
            "acomodacion" => $servicio_name[2],
            "ingreso" => $item->detalle[0]->fecha->format('d/m/Y'),
            "salida" => $item->detalle[0]->fecha-
>addDays(intval($item->detalle[0]->numero))->format('d/m/Y'),
            "in" => $item->detalle[0]->llegada->format('H:i'),
            "out" => $item->detalle[0]->salida->format('H:i'),
        ]);
        $notas->push(
```



```
        (object) [  
            "nombre" => 'ALOJAMIENTO: '. $item->proveedor,  
            "descripcion" => 'RESERVADO POR: '.$item->  
>por ?? 'PROPIO',  
        ]  
    );  
} elseif($servicio_name[0] == 'SERV') {  
    if($servicio_name[1] == 'fa fa-subway') {  
        # trenes  
        foreach($item->detalle as $ytem) {  
            $trenes->push([  
                "empresa" => $item->proveedor,  
                "servicio" => $ytem->numero,  
                "fecha" => $ytem->fecha->format('d/m/Y'),  
                "salida" => $ytem->salida->format('H:i'),  
                "llegada" => $ytem->llegada->format('H:i'),  
                "origen" => $item->origen,  
                "destino" => $item->destino,  
            ]);  
        }  
        $notas->push(  
            (object) [  
                "nombre" => 'TREN: '. $item->proveedor,  
                "descripcion" => 'RESERVADO POR: '.$item->  
>por ?? 'PROPIO',  
            ]  
        );  
    } else {  
        # vuelos / buses  
        foreach($item->detalle as $ytem) {  
            $vuelos_buses->push([  
                "origen" => $item->origen,
```




```
        'cantidad' => $detalle->cantidad,
        'tipo_servicio' => $detalle->tipo_servicio
    ]
    );
}
}
$grouped = $auxiliar_a->groupBy('hospedaje_id');
$contador = 0;
$ingreso_a = \Carbon\Carbon::parse($file->fecha_inicio);
foreach($grouped as $hospedaje_id => $group) {
    if($hospedaje_id) {
        $hospedaje = PHospedaje::findOrFail($hospedaje_id);
        $noche_alo = count($noches) > 0 ? $noches[$contador]-
>cantidad : 0;

        $acomodacion = array(); $servicio_aco = array();
        foreach($group as $detalle) if($detalle->cantidad && $detalle-
>cantidad > 0) array_push($acomodacion, $detalle->tipo_servicio);
        $unique = array_unique($acomodacion);
        $tuple = array('Cero', 'Simple', 'Doble', 'Triple', 'Cuadruple',
, 'Quintuple', 'Sextuple', 'Septuple', 'Octuple', 'Nonuple', 'Decacuple');
        for ($y=0; $y < count($unique); $y++) array_push($servicio_aco
, 'Hab. '.$tuple[$unique[$y]]);
        $alojamiento->push([
            "hotel" => $hospedaje->nombre,
            "direccion" => $hospedaje->direccion,
            "telefono" => $hospedaje->telefonos,
            "acomodacion" => count($servicio_aco) > 0 ? implode(', ',
$servicio_aco) : 'NO ASIGNADO',
            "ingreso" => $ingreso_a->format('d/m/Y'),
            "salida" => $ingreso_a->addDays($noche_alo)-
>format('d/m/Y'),
            "in" => $hospedaje->checkin,
```



```
        "out" => $hospedaje->checkout,  
    ]);  
    $contador++;  
}  
}  
# liberados  
$liberado = DLiberado::where('file_id', $id)->first();  
  
$margen = (100 - floatval($file->margen)) / 100;  
$cot_basica = '0';  
  
$dmpdf = new PDF;  
# tipo | 1=liquidacion(normal+resumen) - 3=normal - 4=facturado(servic  
ios, excursion, boletos, trenes) - 5=selectivo  
if($tipo == '1' || $tipo == '3' || $tipo == '5') {  
    $dmpdf->  
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario','ahora'  
, 'file', 'allservicio', 'trenes', 'vuelos_buses', 'alojamiento', 'notas', 'tip  
o', 'liquidar'))  
        ->with('adicionale', $file->adicional_contratados)-  
>with('otros', $file->otro_contratados));  
    $dmpdf->setPaper($this->PAPER_SIZES['a5']);  
}  
# 2=pasajero(cotizacion basica)  
if($tipo == '2') {  
    $dmpdf->  
>loadHtml(view('cotizacion.cotizador.cotizaciones_completo_imprimir')-  
>with('usuario', $usuario)  
        ->with('ahora', $ahora)->with('adicionale', $file->  
>adicional_contratados)->with('allservicio', $allservicio)-  
>with('noches', $noches)  
        ->with('otros', $file->otro_contratados)->with('modelo', $file)-  
>with('itinerario', $itinerario)->with('ticket', $file->boletos_contratados)-  
>with('hospedaje', $hospedajes)
```




```
        ->with('capacidad', $capacidad)->with('margen', $margen)-  
>with('liberado', $liberado)->with('type', $cot_basica));  
        $dompdf->setPaper($this->PAPER_SIZES['a4']);  
    }  
    # 4 = facturado  
    if($tipo == '4') {  
        $dompdf->loadHtml(view('cotizacion.cotizador.file_facturado')-  
>with('usuario', $usuario)  
        ->with('ahora', $ahora)->with('facturado', $facturado)-  
>with('modelo', $file)->with('margen', $margen));  
        $dompdf->setPaper($this->PAPER_SIZES['a4'], 'landscape');  
    }  
    # 6=agencia  
    if($tipo == '6') {  
        $dompdf->  
>loadHtml(view('cotizacion.cotizador.cotizaciones_completo_imprimir_agencia')-  
>with('usuario', $usuario)  
        ->with('ahora', $ahora)->with('adicionale', $file-  
>adicional_contratados)->with('allservicio', $allservicio)-  
>with('noches', $noches)  
        ->with('otros', $file->otro_contratados)->with('modelo', $file)-  
>with('itinerario', $itinerario)->with('ticket', $file->boletos_contratados)  
        ->with('hospedaje', $hospedajes)  
        ->with('capacidad', $capacidad)->with('margen', $margen)-  
>with('type', $cot_basica));  
        $dompdf->setPaper($this->PAPER_SIZES['a4']);  
    }  
    $dompdf->render();  
    $response = \Response::make($dompdf->output(), 200);  
    $response->header('Content-Type', "application/pdf");  
    return $response;  
}  
  
public function nuevoimprimir(Request $request, $id, $tipo, $archivo)
```



```
{
    $entrada = $request->all();
    $dompdf = new PDF;

    $usuario = $request->user();
    $ahora = now();

    $capacidad = SHospedajeServicio::select('capacidad')-
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-
>groupBy('capacidad')->get();

    $modelo = DFile::with('integrantes')->with('boletos_contratados')-
>withCount('boletos_contratados')->with('otro_contratados')-
>withCount('otro_contratados')->where('cuenta_id', $this->getCuenta())-
>findOrFail($id);

    $notas = DNotas::where('tipo', 'file')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();

    $liquidar = collect(); $incluye = collect(); $vuelos_buses = collect()
; $trenes = collect(); $alojamiento = collect();

    $excursion = DExcursion::with('excursion')-
>with(['servicio' => function($query) { $query->with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle'); }])-
>with('detalle')->where('file_id', $id)->where('cuenta_id', $this-
>getCuenta());

    if(isset($entrada["endose"])) $excursion = $excursion-
>whereIn('id', json_decode($entrada["endose"]));

    foreach ($excursion->get() as $item) {
        if($item->endosado == '0') {
            foreach ($item->servicio as $data) {
                $incluye->push($data);
                $facturado = '0';
                if($data->tipo == '0') {
                    $facturado = $data->servicio_sel_tran->tipotransporte-
>variable;
                }
            }
        }
    }
}
```



```
    }
    $liquidar->push((object) [
        "nombre" => $data->generado,
        "group" => "Servicios",
        "facturado" => $facturado,
        "cantidad" => $this->CantidadPersonas($data->detalle),
        "dolares" => $data->free == '0' ? floatval($data->precios) : 0,
        "soles" => 0,
    ]);
}
} else {
    $incluye->push($item);
    $liquidar->push((object) [
        "nombre" => $item->generado,
        "group" => "Servicios",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
}

    $servicio = DServicio::with('servicio_sel_tran')->with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle')->whereNull('doc_excursion_id')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->where('dia', 'like', 'dia%');

    if(isset($entrada["servicio"])) {
        $servicio = $servicio->whereIn('id', json_decode($entrada["servicio"]));
    }

    foreach($servicio->get() as $item) {
        $variable = TRUE;
```



```
$facturado = '0';
if($item->tipo == '0') {
    $facturado = $item->servicio_sel_tran->tipotransporte-
>variable;
    if($item->servicio_sel_tran->tipotransporte-
>variable == '1') $variable = FALSE;
}
if(!$variable) {
    if($item->itinerario) {
        if($item->servicio_sel_tran->tipotransporte-
>nombre == 'FERROVIARIO') {
            # trenes
            foreach($item->itinerario->detalle as $ytem) {
                $trenes->push([
                    "empresa" => $item->itinerario->proveedor,
                    "servicio" => $item->servicio_sel_tran-
>nombre . ' # ' . $ytem->numero,
                    "fecha" => $ytem->fecha->format('d/m/Y'),
                    "salida" => $ytem->salida->format('H:i'),
                    "llegada" => $ytem->llegada->format('H:i'),
                    "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                    "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
                ]);
            }
        } else {
            # vuelos / buses
            foreach($item->itinerario->detalle as $ytem) {
                $vuelos_buses->push([
                    "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
```



```
                "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
                "compania" => $item->itinerario->proveedor,
                "numero" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
            ]);
        }
    }
}
$incluye->push($item);
$liquidar->push((object) [
    "nombre" => $item->generado,
    "group" => "Servicios",
    "facturado" => $facturado,
    "cantidad" => $this->CantidadPersonas($item->detalle),
    "dolares" => $item->free == '0' ? floatval($item-
>precios) : 0,
    "soles" => 0,
]);
}
foreach($modelo->otro_contratados as $item) {
    $liquidar->push((object) [
        "nombre" => $item->nombre,
        "group" => "Servicios",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
```



```
}  
  
foreach($modelo->boletos_contratados as $item) {  
    # dolares = 1 -> SI | dolares = 0 -> NO  
    if($item->padre_id == NULL) {  
        $liquidar->push((object) [  
            "nombre" => $item->ticket->descripcion,  
            "group" => "Boletos",  
            "facturado" => '1',  
            "cantidad" => $this->CantidadPersonas($item->detalle),  
            "dolares" => $item->ticket->  
>dolares == '1' ? floatval($item->precios) : 0,  
            "soles" => $item->ticket->dolares == '0' ? floatval($item->  
>precios) * floatval($modelo->tipo_cambio) : 0 ,  
        ]);  
    }  
}  
  
$allservicio = $incluye->sortBy('dia')->groupBy('dia');  
  
$descripcion = DDescripcion::where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
$itinerario = $descripcion->sortBy('dia');  
# hospedaje  
$auxiliar_a = collect();  
$noches = DNoches::where('file_id', $id)->where('cuenta_id', $this->  
>getCuenta())->get();  
$hospedajes = DHospedaje::with('detalle')->where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
foreach($hospedajes as $hosp) {  
    foreach($hosp->detalle as $detalle) {  
        $auxiliar_a->push(  
            (object) [  
                'hospedaje_id' => $detalle->hospedaje_id,  
                'cantidad' => $detalle->cantidad,
```



```
        'tipo_servicio' => $detalle->tipo_servicio
    ]
    );
}
}
$grouped = $auxiliar_a->groupBy('hospedaje_id');
$contador = 0;
$ingreso_a = \Carbon\Carbon::parse($modelo->fecha_inicio);
foreach($grouped as $hospedaje_id => $group) {
    if($hospedaje_id) {
        $hospedaje = PHospedaje::findOrFail($hospedaje_id);
        $noche_alo = count($noches) > 0 ? $noches[$contador]-
>cantidad : 0;

        $acomodacion = array(); $servicio_aco = array();
        foreach($group as $detalle) if($detalle->cantidad && $detalle-
>cantidad > 0) array_push($acomodacion, $detalle->tipo_servicio);
        $unique = array_unique($acomodacion);
        $tuple = array('Cero', 'Simple', 'Doble', 'Triple', 'Cuadruple',
', 'Quintuple', 'Sextuple', 'Septuple', 'Octuple', 'Nonuple', 'Decacuple');
        for ($y=0; $y < count($unique); $y++) array_push($servicio_aco
, 'Hab. '.$tuple[$unique[$y]]);
        $alojamiento->push([
            "hotel" => $hospedaje->nombre,
            "direccion" => $hospedaje->direccion,
            "telefono" => $hospedaje->telefonos,
            "acomodacion" => count($servicio_aco) > 0 ? implode(', ',
$servicio_aco) : 'NO ASIGNADO',
            "ingreso" => $ingreso_a->format('d/m/Y'),
            "salida" => $ingreso_a->addDays($noche_alo)-
>format('d/m/Y'),
            "in" => $hospedaje->checkin,
            "out" => $hospedaje->checkout,
```



```
    ]);
    $contador++;
  }
}

$facturado = $liquidar->groupBy('group');
$margen = (100 - floatval($modelo->margen)) / 100;

if($archivo == 'pdf') {
    switch ($tipo) {
        case '1':
            # Liquidar
            $dompokdf-
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario', 'ahora',
', 'modelo', 'vuelos_buses', 'alojamiento', 'trenes', 'notas', 'allservicio',
'tipo', 'liquidar')));

            $dompokdf->setPaper($this->PAPER_SIZES['a5']);

            break;
        case '2':
            # Agencia
            $dompokdf-
>loadHtml(view('cotizacion.cotizador.cotizaciones_completo_imprimir_agencia',
compact('usuario', 'ahora', 'modelo', 'allservicio', 'margen'))-
>with('ticket', $modelo->boletos_contratados)->with('otros', $modelo-
>otro_contratados));

            $dompokdf->setPaper($this->PAPER_SIZES['a4']);

            break;
        case '3':
            # Normal
            $dompokdf-
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario', 'ahora',
', 'modelo', 'vuelos_buses', 'alojamiento', 'trenes', 'notas', 'allservicio',
'tipo')));

            $dompokdf->setPaper($this->PAPER_SIZES['a5']);
```




```
        break;
    case '4':
        # Facturado
        $dmpdf-
>loadHtml(view('cotizacion.cotizador.file_facturado', compact('usuario', 'ahor
a', 'modelo', 'facturado', 'margen')));
        $dmpdf->setPaper($this->PAPER_SIZES['a4']);
        break;
    case '5':
        # Selectivo
        $dmpdf-
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario', 'ahora
', 'modelo', 'vuelos_buses', 'alojamiento', 'trenes', 'notas', 'allservicio',
'tipo')));
        $dmpdf->setPaper($this->PAPER_SIZES['a5']);
        break;
    default:
        $dmpdf->loadHtml('<h3>Error!</h3>');
        $dmpdf->setPaper($this->PAPER_SIZES['a4']);
        break;
    }
    $dmpdf->render();
    $response = \Response::make($dmpdf->output(), 200);
    $response->header('Content-Type', "application/pdf");
} elseif($archivo == 'word') {
    switch ($tipo) {
        case '3':
            $content = view('cotizacion.cotizador.file_imprimir_word',
compact('usuario', 'ahora', 'modelo', 'vuelos_buses', 'alojamiento', 'trenes'
, 'notas', 'allservicio', 'tipo'))->render();
            break;
        default:
            $content = '<h3>Error!</h3>';
            break;
    }
}
```



```
    }

    $response = \Response::make($content, 200);
    $response->header('Content-Type', "text/html");
    $response->header('Content-
Disposition', "attachment;Filename=".$modelo->nombre.".doc");
    }

    return $response;
}

public function selectivo(Request $request, $id)
{
    $usuario = $request->user();
    $ahora = now();
}

protected function CantidadPersonas($detalle) {
    $total = 0;
    foreach($detalle as $item) $total += $item->cantidad;
    return $total;
}

protected function ConversionDeEstado($estado)
{
    $retorno = '';
    if($estado == '0') {
        $retorno = '<span class="label label-default">CONFIRMADO</span>';
    } elseif($estado == '1') {
        $retorno = '<span class="label label-info">PROGRAMADO</span>';
    } elseif($estado == '2') {
```



```
        $retorno = '<span class="label label-inverse">PENDIENTE</span>';  
    } elseif($estado == '3') {  
        $retorno = '<span class="label label-success">COMPRADO</span>';  
    }  
    return $retorno;  
}  
}
```

➤ **ANEXO 10:** Código fuente – Controlador Caja chica

```
<?php  
  
namespace App\Http\Controllers\Cotizacion\Proceso;  
  
use App\CCArqueo;  
use App\CCEgresos;  
use App\CCIngresos;  
use App\Exports\CajaExport;  
use Illuminate\Http\Request;  
use App\Http\Controllers\Controller;  
use Dompdf\Dompdf as PDF;  
use Illuminate\Support\Facades\DB;  
  
class CajaChicaController extends Controller  
{  
    private $PAPER_SIZES = array(  
        "4a0" => array(0, 0, 4767.87, 6740.79),  
        "2a0" => array(0, 0, 3370.39, 4767.87),  
        "a0" => array(0, 0, 2383.94, 3370.39),  
        "a1" => array(0, 0, 1683.78, 2383.94),  
        "a2" => array(0, 0, 1190.55, 1683.78),  
        "a3" => array(0, 0, 841.89, 1190.55),  
        "a4" => array(0, 0, 595.28, 841.89),
```



```
"a5" => array(0, 0, 419.53, 595.28),
"a6" => array(0, 0, 297.64, 419.53),
"a7" => array(0, 0, 209.76, 297.64),
"a8" => array(0, 0, 147.40, 209.76),
"a9" => array(0, 0, 104.88, 147.40),
"a10" => array(0, 0, 73.70, 104.88),
"b0" => array(0, 0, 2834.65, 4008.19),
"b1" => array(0, 0, 2004.09, 2834.65),
"b2" => array(0, 0, 1417.32, 2004.09),
"b3" => array(0, 0, 1000.63, 1417.32),
"b4" => array(0, 0, 708.66, 1000.63),
"b5" => array(0, 0, 498.90, 708.66),
"b6" => array(0, 0, 354.33, 498.90),
"b7" => array(0, 0, 249.45, 354.33),
"b8" => array(0, 0, 175.75, 249.45),
"b9" => array(0, 0, 124.72, 175.75),
"b10" => array(0, 0, 87.87, 124.72),
"c0" => array(0, 0, 2599.37, 3676.54),
"c1" => array(0, 0, 1836.85, 2599.37),
"c2" => array(0, 0, 1298.27, 1836.85),
"c3" => array(0, 0, 918.43, 1298.27),
"c4" => array(0, 0, 649.13, 918.43),
"c5" => array(0, 0, 459.21, 649.13),
"c6" => array(0, 0, 323.15, 459.21),
"c7" => array(0, 0, 229.61, 323.15),
"c8" => array(0, 0, 161.57, 229.61),
"c9" => array(0, 0, 113.39, 161.57),
"c10" => array(0, 0, 79.37, 113.39),
"ra0" => array(0, 0, 2437.80, 3458.27),
"ra1" => array(0, 0, 1729.13, 2437.80),
"ra2" => array(0, 0, 1218.90, 1729.13),
"ra3" => array(0, 0, 864.57, 1218.90),
```



```
"ra4" => array(0, 0, 609.45, 864.57),
"sra0" => array(0, 0, 2551.18, 3628.35),
"sra1" => array(0, 0, 1814.17, 2551.18),
"sra2" => array(0, 0, 1275.59, 1814.17),
"sra3" => array(0, 0, 907.09, 1275.59),
"sra4" => array(0, 0, 637.80, 907.09),
"letter" => array(0, 0, 612.00, 792.00),
"legal" => array(0, 0, 612.00, 1008.00),
"ledger" => array(0, 0, 1224.00, 792.00),
"tabloid" => array(0, 0, 792.00, 1224.00),
"executive" => array(0, 0, 521.86, 756.00),
"folio" => array(0, 0, 612.00, 936.00),
"commercial #10 envelope" => array(0, 0, 684, 297),
"catalog #10 1/2 envelope" => array(0, 0, 648, 864),
"8.5x11" => array(0, 0, 612.00, 792.00),
"8.5x14" => array(0, 0, 612.00, 1008.0),
"11x17" => array(0, 0, 792.00, 1224.00),
"ticket" => array(0, 0, 280.63, 595.27),
);

public function __construct()
{
    $this->middleware('login');
    $this->middleware('privilege:cotprocajachica');
}

public function index(Request $request)
{
    $verificacion = CCArqueo::where('tipo', 'GRANDE')-
>where('estado', 'GUARDADO')->where('cuenta_id', $this->getCuenta())->count();
    $personal = DB::select("select aus.id as user_id, CONCAT(ape.amaterno,
' ',ape.apaterno,', ',ape.nombres, ' (' ,ISNULL(LTRIM(aus.username), 'N/A'), ')
```



```
' ) as personal from ADM_usuario aus join ADM_Personal ape on aus.personal_id =
ape.id");

        return view('cotizacion.proceso.caja')->with('tipo', '1')-
>with('personal', $personal)->with('verificacion', $verificacion);
    }

    public function export(Request $request, $id, $tipo)
    {
        $usuario = $request->user();
        $ahora = now();

        $arqueo = CCArqueo::findOrFail($id);

        $ingresos = CCIngresos::where('arqueo_id', $id)-
>orderBy('fecha', 'desc')->get();
        $egresos = CCEgresos::with('conceptos.clase')->with('documento')-
>where('arqueo_id', $id)->orderBy('fecha', 'desc')->get();

        if($tipo == 'pdf') {
            $dmpdf = new PDF;

            $dmpdf->loadHtml(view('cotizacion.proceso.caja_imprimir')-
>with('usuario', $usuario)->with('ahora', $ahora)->with('arqueo', $arqueo)
->with('ingresos', $ingresos)->with('egresos', $egresos));
            $dmpdf->setPaper($this->PAPER_SIZES['a4'], 'landscape');

            $dmpdf->render();
            $response = \Response::make($dmpdf->output(), 200);
            $response->header('Content-Type', "application/pdf");
            return $response;
        } else {
            $modelo = collect();
        }
    }
}
```



```
foreach ($ingresos as $index => $data) {
    $modelo->push((object) [
        'tipo' => 'INGRESOS',
        'nro' => $index + 1,
        'ingreso' => $data->fecha->format('Y-m-d'),
        'registro' => $data->created_at->format('Y-m-d'),
        'soles' => floatval($data->soles),
        'dolares' => floatval($data->dolares),
        'fisico' => $data->fisico ?? 'SI',
        'concepto' => $data->conceptos->descripcion ?? '-',
        'descripcion' => $data->concepto,
        'beneficiado' => '-',
        'documento' => '-',
        'observacion' => $data->observacion,
    ]);
}

foreach ($egresos as $data) {
    $modelo->push((object) [
        'tipo' => 'EGRESOS',
        'nro' => $data->nro_operacion,
        'ingreso' => $data->fecha->format('Y-m-d'),
        'registro' => $data->created_at->format('Y-m-d'),
        'soles' => floatval($data->soles),
        'dolares' => floatval($data->dolares),
        'fisico' => '-',
        'concepto' => $data->conceptos->descripcion,
        'descripcion' => $data->concepto,
        'beneficiado' => $data->beneficiado." / ".$data->doc_beneficiado,
        'documento' => $data->documento->descripcion." / ".$data->nro_documento,
        'observacion' => $data->observacion,
```



```
        ]);  
    }  
  
    return new CajaExport($modelo);  
}  
}
```

➤ **ANEXO 11:** Código fuente – Controlador Programar Servicios

```
<?php  
  
namespace App\Http\Controllers\Cotizacion\Proceso;  
  
use App\DDescripcion;  
use App\DExcursion;  
use App\DFile;  
use App\DHospedaje;  
use App\DIterinario;  
use App\DLiberado;  
use App\DNoches;  
use App\DNotas;  
use App\DServicio;  
use App\DServicioDetalle;  
use Illuminate\Http\Request;  
use App\Http\Controllers\Controller;  
use App\Idioma;  
use App\Nacionalidad;  
use App\PHospedaje;  
use App\SGservicioCosto;  
use App\SGuiadoServicio;  
use App\SHospedajeServicio;  
use App\STransporteServicio;
```




```
use App\STServicioCosto;
use App\SVServicioCosto;
use App\TipoTransporte;
use App\TipoVehiculo;
use App\VFiles;
use Illuminate\Support\Facades\DB;
use Dompdf\Dompdf as PDF;

class ProgramarController extends Controller
{
    private $PAPER_SIZES = array(
        "4a0" => array(0, 0, 4767.87, 6740.79),
        "2a0" => array(0, 0, 3370.39, 4767.87),
        "a0" => array(0, 0, 2383.94, 3370.39),
        "a1" => array(0, 0, 1683.78, 2383.94),
        "a2" => array(0, 0, 1190.55, 1683.78),
        "a3" => array(0, 0, 841.89, 1190.55),
        "a4" => array(0, 0, 595.28, 841.89),
        "a5" => array(0, 0, 419.53, 595.28),
        "a6" => array(0, 0, 297.64, 419.53),
        "a7" => array(0, 0, 209.76, 297.64),
        "a8" => array(0, 0, 147.40, 209.76),
        "a9" => array(0, 0, 104.88, 147.40),
        "a10" => array(0, 0, 73.70, 104.88),
        "b0" => array(0, 0, 2834.65, 4008.19),
        "b1" => array(0, 0, 2004.09, 2834.65),
        "b2" => array(0, 0, 1417.32, 2004.09),
        "b3" => array(0, 0, 1000.63, 1417.32),
        "b4" => array(0, 0, 708.66, 1000.63),
        "b5" => array(0, 0, 498.90, 708.66),
        "b6" => array(0, 0, 354.33, 498.90),
        "b7" => array(0, 0, 249.45, 354.33),
```



```
"b8" => array(0, 0, 175.75, 249.45),  
"b9" => array(0, 0, 124.72, 175.75),  
"b10" => array(0, 0, 87.87, 124.72),  
"c0" => array(0, 0, 2599.37, 3676.54),  
"c1" => array(0, 0, 1836.85, 2599.37),  
"c2" => array(0, 0, 1298.27, 1836.85),  
"c3" => array(0, 0, 918.43, 1298.27),  
"c4" => array(0, 0, 649.13, 918.43),  
"c5" => array(0, 0, 459.21, 649.13),  
"c6" => array(0, 0, 323.15, 459.21),  
"c7" => array(0, 0, 229.61, 323.15),  
"c8" => array(0, 0, 161.57, 229.61),  
"c9" => array(0, 0, 113.39, 161.57),  
"c10" => array(0, 0, 79.37, 113.39),  
"ra0" => array(0, 0, 2437.80, 3458.27),  
"ra1" => array(0, 0, 1729.13, 2437.80),  
"ra2" => array(0, 0, 1218.90, 1729.13),  
"ra3" => array(0, 0, 864.57, 1218.90),  
"ra4" => array(0, 0, 609.45, 864.57),  
"sra0" => array(0, 0, 2551.18, 3628.35),  
"sra1" => array(0, 0, 1814.17, 2551.18),  
"sra2" => array(0, 0, 1275.59, 1814.17),  
"sra3" => array(0, 0, 907.09, 1275.59),  
"sra4" => array(0, 0, 637.80, 907.09),  
"letter" => array(0, 0, 612.00, 792.00),  
"legal" => array(0, 0, 612.00, 1008.00),  
"ledger" => array(0, 0, 1224.00, 792.00),  
"tabloid" => array(0, 0, 792.00, 1224.00),  
"executive" => array(0, 0, 521.86, 756.00),  
"folio" => array(0, 0, 612.00, 936.00),  
"commercial #10 envelope" => array(0, 0, 684, 297),  
"catalog #10 1/2 envelope" => array(0, 0, 648, 864),
```



```
"8.5x11" => array(0, 0, 612.00, 792.00),
"8.5x14" => array(0, 0, 612.00, 1008.0),
"11x17" => array(0, 0, 792.00, 1224.00),
"ticket" => array(0, 0, 280.63, 595.27),
);

public function __construct()
{
    $this->middleware('login');
    //$this->middleware('privilege:cotproprogramacion');
}

public function index(Request $request)
{
    $entrada = $request->all();
    $tipotransporte = TipoTransporte::select('nombre')->where('variable', '0')->get();
    $nacionalidad = Idioma::all();

    if(isset($entrada["fecha"])) {
        //$consulta = DB::select('exec pj_VisualizarProgramacion ?, ?', [$entrada["fecha"], '0']);
        $consulta = DB::select('exec pj_VisualizarProgramacion_v3 ?', [$entrada["fecha"]]);
        foreach($consulta as $item) {
            $item->categoria_abr = $this->Abreviatura($item->categoria, " ", "", 1);
            $item->cliente_abr = $this->Abreviatura($item->cliente, " ", "", 1);
            $item->nacionalidad_abr = $this->Abreviatura($item->nacionalidad, "-", "-", 3);

            $item->transporte = json_decode($item->transporte, true);
        }
    }
}
```



```
$item->alimentos = json_decode($item->alimentos, true);
$item->guiado = json_decode($item->guiado, true);

$item->adicional = json_decode($item->adicional, true);
$item->otros = json_decode($item->otros, true);
$item->endosado = json_decode($item->endosado, true);

$item->notas = json_decode($item->notas, true);

$item->info_alojamiento = json_decode($item->info_alojamiento, true);
$item->info_transporte = json_decode($item->info_transporte, true);
}

return redirect()-
>action('Cotizacion\Proceso\ProgramarController@index')-
>with('files', $consulta)->withInput($request->all());
}

return view('cotizacion.proceso.programar', compact('tipotransporte',
'nacionalidad'));
}

public function actualizacion(Request $request)
{
    $entrada = $request->all();

    $modelo = collect(DB::select("select * from DOC_Servicio where dia not
like 'dia_%' and YEAR(created_at) = ".$entrada["year"]." and MONTH(created_at
) = ".$entrada["month"].""));
    $result = $modelo->groupBy('dia');

    foreach($result as $dia => $item) {
        if($dia != 'compartido') {
```



```
$chunks = $item->chunk(50);
foreach($chunks as $xtem) {
    foreach ($xtem as $ytem) {
        $file = DFile::findOrFail($ytem->file_id);
        $mayores = $file->nac_adulto+$file->
>nac_estudiante+$file->ext_adulto+$file->ext_estudiante+$file->
>can_adulto+$file->can_estudiante;
        $menores = $file->nac_menor+$file->ext_menor+$file->
>can_menor;

        DServicioDetalle::where('doc_servicio_id', $ytem->id)-
>delete();

        $auxiliar = new DServicioDetalle;
        $auxiliar->doc_servicio_id = $ytem->id;
        $auxiliar->nacionalidad_tipo = '0';
        $auxiliar->edad_tipo = '0';
        $auxiliar->cantidad = $mayores + $menores;

        if($ytem->tipo == '0') {
            $vehiculo = TipoVehiculo::select('id')-
>where('nombre', $ytem->vehiculo_id)->first();
            $servicio = STransporteServicio::findOrFail($ytem->
>servicio_id);
            if($vehiculo) {
                $costos = STServicioCosto::where('transporte_i
d', $ytem->proveedor_id)->where('transporte_servicio_id', $ytem->servicio_id)-
>where('tipo_vehiculo_id', $vehiculo->id)->where('edad_tipo', '0')-
>where('nacionalidad', '0')->where('year', '2019')->sum('costo');
                $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
            } else {
                $costos = STServicioCosto::where('transporte_i
d', $ytem->proveedor_id)->where('transporte_servicio_id', $ytem->servicio_id)-
```



```
>where('edad_tipo', '0')->where('nacionalidad', '0')->where('year', '2019')->sum('costo');

        $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
    }
}

    if($ytem->tipo == '1') {
        $array_edad = array();
        if($mayores > 0) { array_push($array_edad, 0); }
        if($menores > 0) { array_push($array_edad, 2); }

        $costos = SVServicioCosto::where('servicio_id', $ytem->servicio_id)->where('year', '2019')->whereIn('edad_tipo', $array_edad)->sum('costo');

        $auxiliar->precio = $costos;
    }

    if($ytem->tipo == '3') {
        $servicio = SGuiadoServicio::findOrFail($ytem->servicio_id);

        $costos = SGservicioCosto::where('servicio_id', $ytem->servicio_id)->where('guia_id', $ytem->proveedor_id)->where('year', '2019')->sum('costo');

        $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
    }

    $auxiliar->user_id = $this->getUser();
    $auxiliar->cuenta_id = $this->getCuenta();
    if($auxiliar->save()) print('Actualizado...');
    else print('Error');
```



```
    }  
  }  
  } else {  
    $agrupado = $item->groupBy('padre_id');  
    foreach ($agrupado as $padre_id => $ytem) {  
      $padre_servicio = DServicio::select('file_id')->  
>whereIn('id', explode(',', $padre_id))->groupBy('file_id')->get();  
      $array_files_selected = array();  
      foreach($padre_servicio as $files_selected) array_push($array_files_selected, $files_selected->file_id);  
      $file = DFile::whereIn('id', $array_files_selected)->  
>get();  
  
      $mayores = 0; $menores = 0;  
      foreach ($file as $file_item) {  
        $mayores += $file_item->nac_adulto+$file_item->nac_estudiante+$file_item->ext_adulto+$file_item->ext_estudiante+$file_item->can_adulto+$file_item->can_estudiante;  
        $menores += $file_item->nac_menor+$file_item->ext_menor+$file_item->can_menor;  
      }  
  
      $chunks = $ytem->chunk(50);  
      foreach($chunks as $xtem) {  
        foreach ($xtem as $ztem) {  
          DServicioDetalle::where('doc_servicio_id', $ztem->  
>id)->delete();  
  
          $auxiliar = new DServicioDetalle;  
          $auxiliar->doc_servicio_id = $ztem->id;  
          $auxiliar->nacionalidad_tipo = '0';  
          $auxiliar->edad_tipo = '0';  
          $auxiliar->cantidad = $mayores + $menores;  
  
          if($ztem->tipo == '0') {
```



```
        $vehiculo = TipoVehiculo::select('id')-
>where('nombre', $ztem->vehiculo_id)->first();

        $servicio = STransporteServicio::findOrFail($z
tem->servicio_id);

        if($vehiculo) {
            $costos = STServicioCosto::where('transpor
te_id', $ztem->proveedor_id)->where('transporte_servicio_id', $ztem-
>servicio_id)->where('tipo_vehiculo_id', $vehiculo->id)-
>where('edad_tipo', '0')->where('nacionalidad', '0')->where('year', '2019')-
>sum('costo');

            $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
        } else {
            $costos = STServicioCosto::where('transpor
te_id', $ztem->proveedor_id)->where('transporte_servicio_id', $ztem-
>servicio_id)->where('edad_tipo', '0')->where('nacionalidad', '0')-
>where('year', '2019')->sum('costo');

            $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
        }
    }

    if($ztem->tipo == '1') {
        $array_edad = array();
        if($mayores > 0) { array_push($array_edad, 0);
    }

        if($menores > 0) { array_push($array_edad, 2);
    }

        $costos = SVServicioCosto::where('servicio_id'
, $ztem->servicio_id)->where('year', '2019')-
>whereIn('edad_tipo', $array_edad)->sum('costo');

        $auxiliar->precio = $costos;
    }

    if($ztem->tipo == '3') {
```




```
        $servicio = SGuiadoServicio::findOrFail($ztem-
>servicio_id);

        $costos = SGservicioCosto::where('servicio_id'
, $ztem->servicio_id)->where('guia_id', $ztem->proveedor_id)-
>where('year', '2019')->sum('costo');

        $auxiliar->precio = $servicio-
>compartido == '0' ? $costos / intval($mayores + $menores) : $costos;
    }

    $auxiliar->user_id = $this->getUser();
    $auxiliar->cuenta_id = $this->getCuenta();
    if($auxiliar->save()) print('Actualizado...');
    else print('Error');
}
}
}
}
}
}

public function imprimir(Request $request, $id)
{
    $usuario = $request->user();
    $ahora = now();
    $tipo = '3';

    $capacidad = SHospedajeServicio::select('capacidad')-
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-
>groupBy('capacidad')->get();

    $file = DFile::with('integrantes')->with('boletos_contratados')-
>withCount('boletos_contratados')->with('adicional_contratados')-
>withCount('adicional_contratados')->with('otro_contratados')-
>withCount('otro_contratados')->where('cuenta_id', $this->getCuenta())-
>findOrFail($id);
```



```
# Notas

$notas = DNotas::where('tipo', 'file')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();

# incluye

$liquidar = collect();

$incluye = collect(); $trenes = collect(); $vuelos_buses = collect();
$alojamiento = collect();

$excursion = DExcursion::with('excursion')-
>with(['servicio' => function($query) { $query->with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle'); }])->with('detalle')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();

foreach ($excursion as $item) {
    if($item->endosado == '0') {
        foreach ($item->servicio as $data) {
            $incluye->push($data);

            $facturado = '0';
            if($data->tipo == '0') {
                $facturado = $data->servicio_sel_tran->tipotransporte-
>variable;
            }
        }
    } else {
        $incluye->push($item);
    }
}

$servicio = DServicio::with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle')-
>whereNull('doc_excursion_id')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->where('dia', 'like', 'dia%')->get();

foreach($servicio as $item) {
    $variable = TRUE;
    $facturado = '0';
```



```
        if($item->tipo == '0') {
            $facturado = $item->servicio_sel_tran->tipotransporte-
>variable;

            if($item->servicio_sel_tran->tipotransporte-
>variable == '1') $variable = FALSE;
        }
        if(!$variable) {
            if($item->itinerario) {
                if($item->servicio_sel_tran->tipotransporte-
>nombre == 'FERROVIARIO') {
                    # trenes
                    foreach($item->itinerario->detalle as $ytem) {
                        $trenes->push([
                            "empresa" => $item->itinerario->proveedor,
                            "servicio" => $item->servicio_sel_tran-
>nombre . ' # ' . $ytem->numero,
                            "fecha" => $ytem->fecha->format('d/m/Y'),
                            "salida" => $ytem->salida->format('H:i'),
                            "llegada" => $ytem->llegada->format('H:i'),
                            "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                            "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
                        ]);
                    }
                } else {
                    # vuelos / buses
                    foreach($item->itinerario->detalle as $ytem) {
                        $vuelos_buses->push([
                            "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                            "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
```



```
        "compania" => $item->itinerario->proveedor,  
        "numero" => $item->numero,  
        "fecha" => $item->fecha->format('d/m/Y'),  
        "salida" => $item->salida->format('H:i'),  
        "llegada" => $item->llegada->format('H:i'),  
    ]);  
    }  
    }  
    }  
    }  
    $incluye->push($item);  
}  
  
$sortservicio = $incluye->sortBy('dia');  
$allservicio = $sortservicio->groupBy('dia');  
  
//dd($incluye);  
# itinerarios  
$itinerarios = DIItinerario::with('detalle')->whereNull('servicio_id')->  
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();  
foreach($itinerarios as $item) {  
    $servicio_name = explode('*', $item->servicio_name);  
    if($servicio_name[0] == 'ALOJ') {  
        $alojamiento->push([  
            "hotel" => $item->proveedor,  
            "direccion" => $item->direccion,  
            "telefono" => $item->telefono ?? '0000',  
            "acomodacion" => $servicio_name[2],  
            "ingreso" => $item->detalle[0]->fecha->format('d/m/Y'),  
            "salida" => $item->detalle[0]->fecha->  
>addDays(intval($item->detalle[0]->numero))->format('d/m/Y'),  
            "in" => $item->detalle[0]->llegada->format('H:i'),
```



```
        "out" => $item->detalle[0]->salida->format('H:i'),
    ]);
    $notas->push(
        (object) [
            "nombre" => 'ALOJAMIENTO: '. $item->proveedor,
            "descripcion" => 'RESERVADO POR: '.$item-
>por ?? 'PROPIO',
        ]
    );
} elseif($servicio_name[0] == 'SERV') {
    if($servicio_name[1] == 'fa fa-subway') {
        # trenes
        foreach($item->detalle as $ytem) {
            $trenes->push([
                "empresa" => $item->proveedor,
                "servicio" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
                "origen" => $item->origen,
                "destino" => $item->destino,
            ]);
        }
        $notas->push(
            (object) [
                "nombre" => 'TREN: '. $item->proveedor,
                "descripcion" => 'RESERVADO POR: '.$item-
>por ?? 'PROPIO',
            ]
        );
    } else {
        # vuelos / buses
```



```
        foreach($item->detalle as $ytem) {
            $vuelos_buses->push([
                "origen" => $item->origen,
                "destino" => $item->destino,
                "compania" => $item->proveedor,
                "numero" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
            ]);
        }
        $notas->push(
            (object) [
                "nombre" => 'TRANSPORTE: '. $item->proveedor,
                "descripcion" => 'RESERVADO POR: '.$item->
                >por ?? 'PROPIO',
            ]
        );
    }
}

# descripcoin
$descripcion = DDescripcion::where('file_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
$itinerario = $descripcion->sortBy('dia');

# hospedaje
$auxiliar_a = collect();
$noches = DNoches::where('file_id', $id)->where('cuenta_id', $this->
>getCuenta()->get());
$hospedajes = DHospedaje::with('detalle')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
foreach($hospedajes as $hosp) {
    foreach($hosp->detalle as $detalle) {
```



```
$auxiliar_a->push(  
    (object) [  
        'hospedaje_id' => $detalle->hospedaje_id,  
        'cantidad' => $detalle->cantidad,  
        'tipo_servicio' => $detalle->tipo_servicio  
    ]  
);  
}  
}  
$grouped = $auxiliar_a->groupBy('hospedaje_id');  
$contador = 0;  
$ingreso_a = \Carbon\Carbon::parse($file->fecha_inicio);  
foreach($grouped as $hospedaje_id => $group) {  
    if($hospedaje_id) {  
        $hospedaje = PHospedaje::findOrFail($hospedaje_id);  
        $noche_alo = count($noches) > 0 ? $noches[$contador]-  
>cantidad : 0;  
  
        $acomodacion = array(); $servicio_aco = array();  
        foreach($group as $detalle) if($detalle->cantidad && $detalle->  
>cantidad > 0) array_push($acomodacion, $detalle->tipo_servicio);  
        $unique = array_unique($acomodacion);  
        $tuple = array('Cero', 'Simple', 'Doble', 'Triple', 'Cuadruple',  
, 'Quintuple', 'Sextuple', 'Septuple', 'Octuple', 'Nonuple', 'Decacuple');  
        for ($y=0; $y < count($unique); $y++) array_push($servicio_aco  
, 'Hab. '.$tuple[$unique[$y]]);  
        $alojamiento->push([  
            "hotel" => $hospedaje->nombre,  
            "direccion" => $hospedaje->direccion,  
            "telefono" => $hospedaje->telefonos,  
            "acomodacion" => count($servicio_aco) > 0 ? implode(', ',  
$servicio_aco) : 'NO ASIGNADO',  
            "ingreso" => $ingreso_a->format('d/m/Y'),
```



```
        "salida" => $ingreso_a->addDays($noche_alo)-
>format('d/m/Y'),
        "in" => $hospedaje->checkin,
        "out" => $hospedaje->checkout,
    ]);
    $contador++;
}
}
# liberados
$liberado = DLiberado::where('file_id', $id)->first();

$margen = 1 + (floatval($file->margen) / 100);
$cot_basica = '0';

$dmpdf = new PDF;
$dmpdf->
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario','ahora',
'file', 'allservicio', 'trenes', 'vuelos_buses', 'alojamiento', 'notas', 'tip
o', 'liquidar')))
    ->with('adicionale', $file->adicional_contratados)-
>with('otros', $file->otro_contratados));
$dmpdf->setPaper($this->PAPER_SIZES['a5']);

$dmpdf->render();
$response = \Response::make($dmpdf->output(), 200);
$response->header('Content-Type', "application/pdf");
return $response;
}

protected function Abreviatura($string, $limiter, $separator, $amount)
{
    $array_string = array();
    $slice = explode($limiter, $string);
```




```
        for ($i=0; $i < count($slice); $i++) array_push($array_string, substr(
        $slice[$i], 0, $amount));
        return implode($separator, $array_string);
    }

    protected function CantidadPersonas($detalle) {
        $total = 0;
        foreach($detalle as $item) $total += $item->cantidad;
        return $total;
    }

    protected function ConversionDeEstado($estado)
    {
        $retorno = '';
        if($estado == '0') {
            $retorno = '<span class="label label-default">CONFIRMADO</span>';
        } elseif($estado == '1') {
            $retorno = '<span class="label label-info">PROGRAMADO</span>';
        } elseif($estado == '2') {
            $retorno = '<span class="label label-inverse">PENDIENTE</span>';
        } elseif($estado == '3') {
            $retorno = '<span class="label label-success">COMPRADO</span>';
        }
        return $retorno;
    }
}
```

➤ **ANEXO 12:** Código fuente – Controlador Liquidacion de cotizaciones

```
<?php

namespace App\Http\Controllers\Cotizacion\Proceso;

use App\DAAdicional;
```



```
use App\DDescripcion;
use App\DExcursion;
use App\DFile;
use App\DHospedaje;
use App\DIterinario;
use App\DLiberado;
use App\DNoches;
use App\DNotas;
use App\DOTros;
use App\DServicio;
use App\DTicket;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Nacionalidad;
use App\PHospedaje;
use App\SHospedajeServicio;
use App\TipoTransporte;
use App\VFiles;
use Illuminate\Support\Facades\DB;
use Dompdf\Dompdf as PDF;

class LiquidarController extends Controller
{
    private $PAPER_SIZES = array(
        "4a0" => array(0, 0, 4767.87, 6740.79),
        "2a0" => array(0, 0, 3370.39, 4767.87),
        "a0" => array(0, 0, 2383.94, 3370.39),
        "a1" => array(0, 0, 1683.78, 2383.94),
        "a2" => array(0, 0, 1190.55, 1683.78),
        "a3" => array(0, 0, 841.89, 1190.55),
        "a4" => array(0, 0, 595.28, 841.89),
        "a5" => array(0, 0, 419.53, 595.28),
```



```
"a6" => array(0, 0, 297.64, 419.53),
"a7" => array(0, 0, 209.76, 297.64),
"a8" => array(0, 0, 147.40, 209.76),
"a9" => array(0, 0, 104.88, 147.40),
"a10" => array(0, 0, 73.70, 104.88),
"b0" => array(0, 0, 2834.65, 4008.19),
"b1" => array(0, 0, 2004.09, 2834.65),
"b2" => array(0, 0, 1417.32, 2004.09),
"b3" => array(0, 0, 1000.63, 1417.32),
"b4" => array(0, 0, 708.66, 1000.63),
"b5" => array(0, 0, 498.90, 708.66),
"b6" => array(0, 0, 354.33, 498.90),
"b7" => array(0, 0, 249.45, 354.33),
"b8" => array(0, 0, 175.75, 249.45),
"b9" => array(0, 0, 124.72, 175.75),
"b10" => array(0, 0, 87.87, 124.72),
"c0" => array(0, 0, 2599.37, 3676.54),
"c1" => array(0, 0, 1836.85, 2599.37),
"c2" => array(0, 0, 1298.27, 1836.85),
"c3" => array(0, 0, 918.43, 1298.27),
"c4" => array(0, 0, 649.13, 918.43),
"c5" => array(0, 0, 459.21, 649.13),
"c6" => array(0, 0, 323.15, 459.21),
"c7" => array(0, 0, 229.61, 323.15),
"c8" => array(0, 0, 161.57, 229.61),
"c9" => array(0, 0, 113.39, 161.57),
"c10" => array(0, 0, 79.37, 113.39),
"ra0" => array(0, 0, 2437.80, 3458.27),
"ra1" => array(0, 0, 1729.13, 2437.80),
"ra2" => array(0, 0, 1218.90, 1729.13),
"ra3" => array(0, 0, 864.57, 1218.90),
"ra4" => array(0, 0, 609.45, 864.57),
```



```
"sra0" => array(0, 0, 2551.18, 3628.35),
"sra1" => array(0, 0, 1814.17, 2551.18),
"sra2" => array(0, 0, 1275.59, 1814.17),
"sra3" => array(0, 0, 907.09, 1275.59),
"sra4" => array(0, 0, 637.80, 907.09),
"letter" => array(0, 0, 612.00, 792.00),
"legal" => array(0, 0, 612.00, 1008.00),
"ledger" => array(0, 0, 1224.00, 792.00),
"tabloid" => array(0, 0, 792.00, 1224.00),
"executive" => array(0, 0, 521.86, 756.00),
"folio" => array(0, 0, 612.00, 936.00),
"commercial #10 envelope" => array(0, 0, 684, 297),
"catalog #10 1/2 envelope" => array(0, 0, 648, 864),
"8.5x11" => array(0, 0, 612.00, 792.00),
"8.5x14" => array(0, 0, 612.00, 1008.0),
"11x17" => array(0, 0, 792.00, 1224.00),
"ticket" => array(0, 0, 280.63, 595.27),
);

public function __construct()
{
    $this->middleware('login');
    $this->middleware('privilege:cotproliquidar');
}

public function index(Request $request)
{
    return view('cotizacion.proceso.liquidar');
}

public function resumen(Request $request, $id)
{
```



```
$nacionalidad = Nacionalidad::all();

$capacidad = SHospedajeServicio::select('capacidad')-
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-
>groupBy('capacidad')->get();

$tipotransporte = TipoTransporte::all();

$serviciosver = collect(); $excursionever = collect();

$file = DFile::withCount('integrantes')->where('cuenta_id', $this-
>getCuenta())->findOrFail($id);

$notas = DNotas::where('file_id', $file->id)-
>where('cuenta_id', $this->getCuenta())->get();

if($file->operador_id != '0') {
    $request->session()->put('cuenta_cambio', $file->operador_id);
} else {
    $request->session()->forget('cuenta_cambio');
}

# Noches # Alojamiento
$alojamiento = collect(); $auxiliar_a = collect();
$noches = DNoches::where('file_id', $id)->where('cuenta_id', $this-
>getCuenta())->get();
$hospedajes = DHospedaje::with('detalle')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->get();
foreach($hospedajes as $hosp) {
    foreach($hosp->detalle as $detalle) {
        if($detalle->cantidad != '0') {
            $auxiliar_a->push(
                (object) [
                    'hospedaje_id' => $detalle->hospedaje_id,
                    'programado' => $detalle->programado,
                    'codigo_reserva' => $detalle->cod_reserva,
                    'nacionalidad_tipo' => $detalle-
>nacionalidad_tipo,
```



```
        'cantidad' => $detalle->cantidad ?? 0,  
        'tipo_servicio' => $detalle->tipo_servicio ?? 0,  
        'precio' => $detalle->precio,  
    ]  
    );  
    }  
    }  
    }  
    }  
$grouped = $auxiliar_a->groupBy('hospedaje_id');  
$contador = 0;  
$ingreso_a = \Carbon\Carbon::parse($file->fecha_inicio);  
foreach($grouped as $hospedaje_id => $group) {  
    $noche_alo = count($noches) > 0 ? $noches[$contador]->cantidad : 0;  
    $nacionalidad_group = $group->groupBy('nacionalidad_tipo');  
    $nacionalidad_tipo = collect();  
    $programado = ''; $codigo_reserva = '';  
    foreach($nacionalidad_group as $nac_tipo => $tipo) {  
        $suma_collect = collect();  
        foreach($tipo as $data) {  
            $programado = $data->programado; $codigo_reserva = $data->  
>codigo_reserva;  
            $suma_collect->push($data->cantidad*$data->  
>tipo_servicio*$data->precio);  
        }  
        $nacionalidad_tipo->push( (object)[  
            'nacionalidad_tipo' => $nac_tipo,  
            'precio' => $suma_collect->sum()  
        ] );  
    }  
    $alojamiento_s = PHospedaje::find($hospedaje_id);  
    if($alojamiento_s) {
```



```
$alojamiento->push(  
    (object) [  
        'hospedaje_id' => $hospedaje_id,  
        'programado' => $programado,  
        'codigo_reserva' => $codigo_reserva,  
        'noches' => $noche_alo,  
        "ingreso" => $ingreso_a->format('d/m/Y'),  
        "in" => $alojamiento_s->checkin ?? '-',  
        "out" => $alojamiento_s->checkout ?? '-',  
        'hospedaje_name' => '<h4 class="no-margin semi-  
bold text-uppercase" id="alojamiento_name_'.$hospedaje_id.'"><i class="fa fa-  
bed m-r-5"></i>'.$alojamiento_s->nombre.'<u class="ml-1 fs-  
14">'.$alojamiento_s->distrito->nombre .'</u><small class="ml-2 fs-  
13">'.str_repeat('<i class="fa fa-star text-warning"></i>', $alojamiento_s-  
>categoria).'</small>'.</h4>',  
        'detalle' => $nacionalidad_tipo  
    ]  
);  
}  
$contador++;  
}  
# Boletos  
$boletosver = collect();  
$ticket = DTicket::where('file_id', $id)->where('cuenta_id', $this-  
>getCuenta())->get();  
foreach($ticket as $item) {  
    if($item->padre_id == NULL) {  
        $boletosver->push([  
            'boleto_id' => $item->ticket_id,  
        ]);  
    }  
}  
}  
# Servicios, Excursiones, Upgrades y Otros
```



```
$collect_por_dia = collect(); $collect_variable = collect(); $collect_
alojamiento = collect();

$servicio = DServicio::with('detalle')->whereNull('doc_excursion_id')-
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())-
>where('dia', 'like', 'dia%')->get();

foreach($servicio as $item) {
    $variable = TRUE;
    if($item->tipo == '0') {
        if($item->servicio_sel_tran->tipotransporte-
>variable == '1') $variable = FALSE;
    }
    if($variable) {
        $collect_por_dia->push([
            'id' => $item->id,
            'color' => 'text-primary',
            'tipo' => 'SERV',
            'nombre' => $item->generado,
            'precio' => $item->precios,
            'free' => $item->free,
            'vehiculo' => $item->vehiculo_id,
            'grupo' => collect(),
            'dia' => explode("_", $item->dia)[1]
        ]);
    } else {
        # verificar itinerario
        $itinerario = DItinerario::where('servicio_id', $item-
>servicio_id)->where('file_id', $id)->where('cuenta_id', $this->getCuenta())-
>first();

        $data_iti = collect();

        if($itinerario) {
            $data_iti->push([
                'proveedor' => $itinerario->proveedor,
```




```
                'ruta' => $itinerario->origen.'-'. $itinerario-
>destino,
                'tipo' => $itinerario->tipo == '11' ? 'IDA-
VUELTA' : ($itinerario->tipo == '10' ? 'IDA' : 'VUELTA'),
                'detalle' => $itinerario->generado,
            ]);
        }
        $collect_variable->push([
            'id' => $item->id,
            'servicio_id' => $item->servicio_id,
            'tipo' => 'SERV',
            'nombre' => $item->generado,
            'precio' => $item->precios,
            'telefono' => '0000',
            'por' => 'PROPIO',
            'direccion' => NULL,
            'icono' => $item->servicio_sel_tran->tipotransporte-
>icono,
            'itinerario' => $data_iti,
            'programado' => $item->programado,
            'codigo_reserva' => $item->codigo_reserva,
            'dia' => explode("_", $item->dia)[1]
        ]);
    }
    $serviciosver->push([
        'servicio_id' => $item->servicio_id,
        'tipo' => $item->tipo,
    ]);
}
$excursion = DExcursion::with('excursion')->with('servicio')-
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($excursion as $item) {
    if($item->endosado == '0') {
```



```
if(count($item->servicio) > 0) {
    $collect_servicios_exc = collect();
    foreach($item->servicio as $ytem) {
        $collect_servicios_exc->push([
            'id' => $ytem->id,
            'tipo' => 'SERV',
            'nombre' => $ytem->generado,
            'precio' => $ytem->precios,
            'free' => $ytem->free,
            'vehiculo' => $ytem->vehiculo_id,
        ]);
        $serviciosver->push([
            'servicio_id' => $ytem->servicio_id,
            'tipo' => $ytem->tipo,
        ]);
    }
    $collect_por_dia->push([
        'id' => $item->id,
        'color' => 'text-complete',
        'tipo' => 'EXCU',
        'nombre' => $item->generado,
        'precio' => $item->precios,
        'tipo' => '10',
        'grupo' => $collect_servicios_exc,
        'dia' => explode("_", $item->dia)[1]
    ]);
}
} else {
    $collect_por_dia->push([
        'id' => $item->id,
        'color' => 'text-complete',
        'tipo' => 'EXCU',
```



```
        'nombre' => $item->generado,  
        'precio' => $item->precios,  
        'free' => '0',  
        'vehiculo' => NULL,  
        'grupo' => collect(),  
        'dia' => explode("_", $item->dia)[1]  
    ]);  
}  
$excursionever->push([  
    'excursion_id' => $item->excursion_id,  
]);  
}  
$upgrade = DAdicional::where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
foreach($upgrade as $item) {  
    $dias = explode(',',$item->dia);  
    for ($i=0; $i < count($dias); $i++) {  
        $collect_por_dia->push([  
            'id' => $item->id.'-'. $dias[$i],  
            'color' => 'text-warning',  
            'tipo' => 'UPDA',  
            'nombre' => $item->generado,  
            'free' => '0',  
            'vehiculo' => NULL,  
            'precio' => number_format(floatval($item->  
>precios)/count($dias), 2, '.', ''),  
            'grupo' => collect(),  
            'dia' => $dias[$i]  
        ]);  
    }  
}
```



```
$otros = DOtros::with('detalle')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();

foreach($otros as $item) {
    //dump($item->dia != '' ? $item->dia : '0');
    $dias = explode(',',$item->dia != '' ? $item->dia : '0');
    for ($i=0; $i < count($dias); $i++) {
        $collect_por_dia->push([
            'id' => $item->id.'-'. $dias[$i],
            'color' => 'text-info',
            'tipo' => 'OTRO',
            'nombre' => $item->nombre,
            'free' => '0',
            'vehiculo' => NULL,
            'precio' => number_format(floatval($item->precios)/count($dias), 2, '.', ''),
            'grupo' => collect(),
            'dia' => $dias[$i]
        ]);
    }
}

//dd('sad');

# Itinerarios

$itinerarios = DItinerario::whereNull('servicio_id')->where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();

foreach($itinerarios as $item) {
    $servicio_name = explode('*', $item->servicio_name);
    $data_iti = collect();
    if($servicio_name[0] == 'ALOJ') {
        $collect_alojamiento->push(
            (object)[
                'id' => $item->id,
                'tipo' => 'ITI_ALO',
            ]
        );
    }
}
```



```
'nombre' => $servicio_name[2],
'direccion' => $item->direccion,
'telefono' => $item->telefono ?? '0000',
'por' => $item->por ?? 'PROPIO',
'icono' => $servicio_name[1],
'proveedor' => $item->proveedor,
'ubicacion' => $item->origen,
'detalle' => $item->generado,
]);
} elseif($servicio_name[0] == 'SERV') {
    $data_iti->push([
        'proveedor' => $item->proveedor,
        'ruta' => $item->tipo == '11' ? ($item->origen.'-'. $item->destino.'-'. $item->origen) : ($item->origen.'-'. $item->destino),
        'tipo' => $item->tipo == '11' ? 'IDA-VUELTA' : ($item->tipo == '10' ? 'IDA' : 'VUELTA'),
        'detalle' => $item->generado,
    ]);
    $collect_variable->push([
        'id' => $item->id,
        'servicio_id' => 0,
        'tipo' => 'ITI_SERV',
        'nombre' => $servicio_name[2],
        'telefono' => $item->telefono ?? '0000',
        'por' => $item->por ?? 'PROPIO',
        'precio' => 0,
        'direccion' => $item->direccion,
        'icono' => $servicio_name[1],
        'itinerario' => $data_iti,
        'programado' => NULL,
        'dia' => 0
    ]);
});
```



```
    }  
  }  
  # Liberados  
  $liberado = DLiberado::where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
  return view('cotizacion.cotizador.files_resumen', compact('file', 'nac  
ionalidad', 'alojamiento', 'ticket',  
  'collect_por_dia', 'collect_variable', 'tipotransporte', 'collect_aloj  
amiento', 'notas', 'boletosver', 'serviciosver', 'excursionever', 'liberado'))  
;  
}  
  
public function liquidar(Request $request, $id)  
{  
  $modelo = collect(DB::select('exec pj_LiquidacionPorFile ?', [$id]));  
  $data = VFiles::where('idfile', $id)->first();  
  //  
  $vendidos = $modelo->where('lado', 'VENDIDO');  
  
  return view('cotizacion.cotizador.liquidacion')  
  ->with('data', $data)  
  ->with('vendidos', $vendidos->groupBy('tipo'))  
  ->with('operado', $modelo->groupBy('tipo'));  
}  
  
public function imprimir(Request $request, $id, $tipo)  
{  
  $usuario = $request->user();  
  $ahora = now();  
  $capacidad = SHospedajeServicio::select('capacidad')-  
>where('cuenta_id', $this->getCuenta())->orderBy('capacidad')-  
>groupBy('capacidad')->get();
```



```
    $file = DFile::with('integrantes')->with('boletos_contratados')-  
>withCount('boletos_contratados')->with('adicional_contratados')-  
>withCount('adicional_contratados')->with('otro_contratados')-  
>withCount('otro_contratados')->where('cuenta_id', $this->getCuenta())-  
>findOrFail($id);  
  
    # Notas  
    $notas = DNotas::where('tipo', 'file')->where('file_id', $id)-  
>where('cuenta_id', $this->getCuenta())->get();  
  
    # incluye  
    $liquidar = collect();  
    $incluye = collect(); $trenes = collect(); $vuelos_buses = collect();  
    $alojamiento = collect();  
  
    $excursion = DExcursion::with('excursion')-  
>with(['servicio' => function($query) { $query->with('servicio_sel_tran')-  
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle'); }])-  
>with('detalle')->where('file_id', $id)->where('cuenta_id', $this->  
    getCuenta())->get();  
  
    foreach ($excursion as $item) {  
        if($item->endosado == '0') {  
            foreach ($item->servicio as $data) {  
                $incluye->push($data);  
  
                $facturado = '0';  
                if($data->tipo == '0') {  
                    $facturado = $data->servicio_sel_tran->tipotransporte-  
>variable;  
                }  
                $liquidar->push((object) [  
                    "nombre" => $data->generado,  
                    "group" => "SERVICIOS",  
                    "facturado" => $facturado,  
                    "cantidad" => $this->CantidadPersonas($data->detalle),  
                    "dolares" => $data->free == '0' ? floatval($data->  
>precios) : 0,
```



```
        "soles" => 0,
    ]);
    }
} else {
    $incluye->push($item);
    $liquidar->push((object) [
        "nombre" => $item->generado,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
}

$servicio = DServicio::with('servicio_sel_tran')-
>with('servicio_sel_vari')->with('servicio_sel_guia')->with('detalle')-
>whereNull('doc_excursion_id')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta())->where('dia', 'like', 'dia%')->get();

foreach($servicio as $item) {
    $variable = TRUE;
    $facturado = '0';
    if($item->tipo == '0') {
        $facturado = $item->servicio_sel_tran->tipotransporte-
>variable;
        if($item->servicio_sel_tran->tipotransporte-
>variable == '1') $variable = FALSE;
    }
    if(!$variable) {
        if($item->itinerario) {
            if($item->servicio_sel_tran->tipotransporte-
>nombre == 'FERROVIARIO') {
                # trenes
```




```
        foreach($item->itinerario->detalle as $ytem) {
            $strenes->push([
                "empresa" => $item->itinerario->proveedor,
                "servicio" => $item->servicio_sel_tran-
>nombre . ' # ' . $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
                "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
            ]);
        }

    } else {
        # vuelos / buses
        foreach($item->itinerario->detalle as $ytem) {
            $vuelos_buses->push([
                "origen" => $ytem->tipo == '10' ? $item-
>itinerario->origen : $item->itinerario->destino,
                "destino" => $ytem->tipo == '10' ? $item-
>itinerario->destino : $item->itinerario->origen,
                "compania" => $item->itinerario->proveedor,
                "numero" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
            ]);
        }
    }
}
```



```
$incluye->push($item);
$liquidar->push((object) [
    "nombre" => $item->generado,
    "group" => "SERVICIOS",
    "facturado" => $facturado,
    "cantidad" => $this->CantidadPersonas($item->detalle),
    "dolares" => $item->free == '0' ? floatval($item->precios) : 0,
    "soles" => 0,
]);
}
foreach($file->adicional_contratados as $item) {
    $liquidar->push((object) [
        "nombre" => $item->generado,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
foreach($file->otro_contratados as $item) {
    $liquidar->push((object) [
        "nombre" => $item->nombre,
        "group" => "SERVICIOS",
        "facturado" => '0',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => floatval($item->precios),
        "soles" => 0,
    ]);
}
foreach($file->boletos_contratados as $item) {
```



```
# dolares = 1 -> SI | dolares = 0 -> NO
if($item->padre_id == NULL) {
    $liquidar->push((object) [
        "nombre" => $item->ticket->descripcion,
        "group" => "BOLETOS",
        "facturado" => '1',
        "cantidad" => $this->CantidadPersonas($item->detalle),
        "dolares" => $item->ticket-
>dolares == '1' ? floatval($item->precios) : 0,
        "soles" => $item->ticket->dolares == '0' ? floatval($item-
>precios) * floatval($file->tipo_cambio) : 0 ,
    ]);
}
}
}
$sortservicio = $incluye->sortBy('dia');
$allservicio = $sortservicio->groupBy('dia');

$facturado = $liquidar->groupBy('group');
//dd($incluye);
# itinerarios
$itinerarios = DIItinerario::with('detalle')->whereNull('servicio_id')-
>where('file_id', $id)->where('cuenta_id', $this->getCuenta())->get();
foreach($itinerarios as $item) {
    $servicio_name = explode('*', $item->servicio_name);
    if($servicio_name[0] == 'ALOJ') {
        $alojamiento->push([
            "hotel" => $item->proveedor,
            "direccion" => $item->direccion,
            "telefono" => $item->telefono ?? '0000',
            "acomodacion" => $servicio_name[2],
            "ingreso" => $item->detalle[0]->fecha->format('d/m/Y'),
            "salida" => $item->detalle[0]->fecha-
>addDays(intval($item->detalle[0]->numero))->format('d/m/Y'),
```



```
        "in" => $item->detalle[0]->llegada->format('H:i'),
        "out" => $item->detalle[0]->salida->format('H:i'),
    ]);
    $notas->push(
        (object) [
            "nombre" => 'ALOJAMIENTO: '. $item->proveedor,
            "descripcion" => 'RESERVADO POR: '.$item-
>por ?? 'PROPIO',
        ]
    );
} elseif($servicio_name[0] == 'SERV') {
    if($servicio_name[1] == 'fa fa-subway') {
        # trenes
        foreach($item->detalle as $ytem) {
            $trenes->push([
                "empresa" => $item->proveedor,
                "servicio" => $ytem->numero,
                "fecha" => $ytem->fecha->format('d/m/Y'),
                "salida" => $ytem->salida->format('H:i'),
                "llegada" => $ytem->llegada->format('H:i'),
                "origen" => $item->origen,
                "destino" => $item->destino,
            ]);
        }
        $notas->push(
            (object) [
                "nombre" => 'TREN: '. $item->proveedor,
                "descripcion" => 'RESERVADO POR: '.$item-
>por ?? 'PROPIO',
            ]
        );
    } else {
```



```
# vuelos / buses

foreach($item->detalle as $ytem) {
    $vuelos_buses->push([
        "origen" => $item->origen,
        "destino" => $item->destino,
        "compania" => $item->proveedor,
        "numero" => $ytem->numero,
        "fecha" => $ytem->fecha->format('d/m/Y'),
        "salida" => $ytem->salida->format('H:i'),
        "llegada" => $ytem->llegada->format('H:i'),
    ]);
}

$notas->push(
    (object) [
        "nombre" => 'TRANSPORTE: '. $item->proveedor,
        "descripcion" => 'RESERVADO POR: '.$item->
>por ?? 'PROPIO',
    ]
);
}

}

# descripcoin
$descripcion = DDescripcion::where('file_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
$itinerario = $descripcion->sortBy('dia');

# hospedaje
$auxiliar_a = collect();
$noches = DNoches::where('file_id', $id)->where('cuenta_id', $this-
>getCuenta()->get());
$hospedajes = DHospedaje::with('detalle')->where('file_id', $id)-
>where('cuenta_id', $this->getCuenta()->get());
foreach($hospedajes as $hosp) {
```



```
foreach($hosp->detalle as $detalle) {
    $auxiliar_a->push(
        (object) [
            'hospedaje_id' => $detalle->hospedaje_id,
            'cantidad' => $detalle->cantidad,
            'tipo_servicio' => $detalle->tipo_servicio
        ]
    );
}
}
$grouped = $auxiliar_a->groupBy('hospedaje_id');
$contador = 0;
$ingreso_a = \Carbon\Carbon::parse($file->fecha_inicio);
foreach($grouped as $hospedaje_id => $group) {
    if($hospedaje_id) {
        $hospedaje = PHospedaje::findOrFail($hospedaje_id);
        $noche_alo = count($noches) > 0 ? $noches[$contador]-
>cantidad : 0;

        $acomodacion = array(); $servicio_aco = array();
        foreach($group as $detalle) if($detalle->cantidad && $detalle-
>cantidad > 0) array_push($acomodacion, $detalle->tipo_servicio);
        $unique = array_unique($acomodacion);
        $tuple = array('Cero', 'Simple', 'Doble', 'Triple', 'Cuadruple',
', 'Quintuple', 'Sextuple', 'Septuple', 'Octuple', 'Nonuple', 'Decacuple');
        for ($y=0; $y < count($unique); $y++) array_push($servicio_aco
, 'Hab. '.$tuple[$unique[$y]]);
        $alojamiento->push([
            "hotel" => $hospedaje->nombre,
            "direccion" => $hospedaje->direccion,
            "telefono" => $hospedaje->telefonos,
            "acomodacion" => count($servicio_aco) > 0 ? implode(', ',
$servicio_aco) : 'NO ASIGNADO',
```



```
        "ingreso" => $ingreso_a->format('d/m/Y'),
        "salida" => $ingreso_a->addDays($noche_alo)-
>format('d/m/Y'),
        "in" => $hospedaje->checkin,
        "out" => $hospedaje->checkout,
    ]);
    $contador++;
}
}
# liberados
$liberado = DLiberado::where('file_id', $id)->first();

$margen = 1 + (floatval($file->margen) / 100);
$cot_basica = '0';

$dmpdf = new PDF;
# tipo | 1=liquidacion(normal+resumen) - 3=normal - 4=facturado(servic
ios, excursion, boletos, trenes) - 5=selectivo
if($tipo == '1' || $tipo == '3') {
    $dmpdf-
>loadHtml(view('cotizacion.cotizador.file_imprimir', compact('usuario','ahora'
,'file', 'allservicio', 'trenes', 'vuelos_buses', 'alojamiento', 'notas', 'tip
o', 'liquidar')))
        ->with('adicionale', $file->adicional_contratados)-
>with('otros', $file->otro_contratados));
    $dmpdf->setPaper($this->PAPER_SIZES['a5']);
}
# 2=pasajero(cotizacion basica)
if($tipo == '2') {
    $dmpdf-
>loadHtml(view('cotizacion.cotizador.cotizaciones_completo_imprimir')-
>with('usuario', $usuario)
```



```
        ->with('ahora', $ahora)->with('adicionale', $file-
>adicional_contratados)->with('allservicio', $allservicio)-
>with('noches', $noches)

        ->with('otros', $file->otro_contratados)->with('modelo', $file)-
>with('itinerario', $itinerario)->with('ticket', $file->boletos_contratados)-
>with('hospedaje', $hospedajes)

        ->with('capacidad', $capacidad)->with('margen', $margen)-
>with('liberado', $liberado)->with('type', $cot_basica));

        $dmpdf->setPaper($this->PAPER_SIZES['a4']);
    }

    # 4 = facturado
    if($tipo == '4') {
        $dmpdf->loadHtml(view('cotizacion.cotizador.file_facturado')-
>with('usuario', $usuario)

        ->with('ahora', $ahora)->with('facturado', $facturado)-
>with('modelo', $file));

        $dmpdf->setPaper($this->PAPER_SIZES['a4'], 'landscape');
    }

    # 6=agencia
    if($tipo == '6') {
        $dmpdf-
>loadHtml(view('cotizacion.cotizador.cotizaciones_completo_imprimir_agencia')-
>with('usuario', $usuario)

        ->with('ahora', $ahora)->with('adicionale', $file-
>adicional_contratados)->with('allservicio', $allservicio)-
>with('noches', $noches)

        ->with('otros', $file->otro_contratados)->with('modelo', $file)-
>with('itinerario', $itinerario)->with('ticket', $file->boletos_contratados)

        ->with('hospedaje', $hospedajes)

        ->with('capacidad', $capacidad)->with('margen', $margen)-
>with('type', $cot_basica));

        $dmpdf->setPaper($this->PAPER_SIZES['a4']);
    }

    $dmpdf->render();

    $response = \Response::make($dmpdf->output(), 200);
```




```
$response->header('Content-Type', "application/pdf");  
return $response;  
}  
  
protected function CantidadPersonas($detalle) {  
    $total = 0;  
    foreach($detalle as $item) $total += $item->cantidad;  
    return $total;  
}  
  
protected function ConversionDeEstado($estado)  
{  
    $retorno = '';  
    if($estado == '0') {  
        $retorno = '<span class="label label-default">CONFIRMADO</span>';  
    } elseif($estado == '1') {  
        $retorno = '<span class="label label-info">PROGRAMADO</span>';  
    } elseif($estado == '2') {  
        $retorno = '<span class="label label-inverse">PENDIENTE</span>';  
    } elseif($estado == '3') {  
        $retorno = '<span class="label label-success">COMPRADO</span>';  
    }  
    return $retorno;  
}  
}
```

➤ **ANEXO 13:** Código fuente – Controlador Reporte de caja

```
<?php  
  
namespace App\Http\Controllers\Cotizacion\Reporte;  
  
use App\CCEgresos;
```



```
use App\CCIngresos;
use App\Exports\ReporteCajaExport;
use Rap2hpoutre\FastExcel\FastExcel;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class CajaController extends Controller
{
    public function __construct()
    {
        $this->middleware('login');
        $this->middleware('privilege:cotrepcaja');
    }

    public function index(Request $request)
    {
        return view('cotizacion.reporte.caja');
    }

    public function imprimir(Request $request)
    {
        $entrada = $request->all();

        $ingresos = CCIngresos::with('conceptos.clase')-
>whereHas('arqueo', function($query) use ($entrada) {
            $query->where('tipo', $entrada["t"]);
        })->with('arqueo')-
>whereBetween('fecha', [$entrada["i"], $entrada["f"]])-
>orderBy('fecha', 'desc')->get();

        $egresos = CCEgresos::with('conceptos.clase')->with('documento')-
>whereHas('arqueo', function($query) use ($entrada) {
            $query->where('tipo', $entrada["t"]);
```



```
    })->with('arqueo')-  
>whereBetween('fecha', [$entrada["i"], $entrada["f"]])-  
>orderBy('fecha', 'desc')->get();  
  
$modelo = collect();  
foreach ($ingresos as $index => $data) {  
    $modelo->push((object) [  
        'id' => $data->arqueo->tipo,  
        'tipo' => 'INGRESOS',  
        'nro' => $index + 1,  
        'ingreso' => $data->fecha->format('Y-m-d'),  
        'registro' => $data->created_at->format('Y-m-d'),  
        'soles' => floatval($data->soles),  
        'dolares' => floatval($data->dolares),  
        'fisico' => $data->fisico ?? 'SI',  
        'clase' => $data->conceptos->clase->descripcion ?? '-',  
        'concepto' => $data->conceptos->descripcion ?? '-',  
        'descripcion' => $data->concepto,  
        'beneficiado' => '-',  
        'documento' => '-',  
        'observacion' => $data->observacion,  
    ]);  
}  
foreach ($egresos as $data) {  
    $modelo->push((object) [  
        'id' => $data->arqueo->tipo,  
        'tipo' => 'EGRESOS',  
        'nro' => $data->nro_operacion,  
        'ingreso' => $data->fecha->format('Y-m-d'),  
        'registro' => $data->created_at->format('Y-m-d'),  
        'soles' => floatval($data->soles),  
        'dolares' => floatval($data->dolares),
```



```
        'fisico' => '-',
        'clase' => $data->conceptos->clase->descripcion,
        'concepto' => $data->conceptos->descripcion,
        'descripcion' => $data->concepto,
        'beneficiado' => $data->beneficiado." / ".$data->doc_beneficiado,
        'documento' => $data->documento->descripcion." / ".$data->nro_documento,
        'observacion' => $data->observacion,
    ]);
}
return new ReporteCajaExport($modelo);
}
```

➤ **ANEXO 14:** Código fuente – Controlador Reporte de pagos

```
<?php

namespace App\Http\Controllers\Cotizacion\Reporte;

use App\Exports\PagosExport;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Operador;
use Illuminate\Support\Facades\DB;

class PagosController extends Controller
{
    public function __construct()
    {
        $this->middleware('login');
        $this->middleware('privilege:cotreppagos');
    }
}
```



```
public function index(Request $request)
{
    $clientes = Operador::where('cuenta_id', $this->getCuenta()->get());
    return view('cotizacion.reporte.pagos')->with('clientes', $clientes);
}

public function imprimir(Request $request)
{
    $entrada = $request->all();

    $operador = '';
    if($entrada["c"] != 'TODOS') {
        if($entrada["c"] != 'PROPIO') {
            $informacion = Operador::findOrFail($entrada["c"]);
            $operador = $informacion->razon_social;
        } else {
            $operador = 'PROPIO';
        }
    }

    $subconsulta = $entrada["c"] == 'TODOS' ? '' : " vfi.operador = '".$operador."' and ";

    $consulta = collect(DB::select(";with SeguimientoPagos(id, counter, cliente, file_id, fecha_abono, transaccion, nro_operacion, entidad, moneda, tipo_cambio, total, abonado, diferencia) as (
        select vfi.idfile, vfi.counter, vfi.operador, vfi.nombre, isnull(format(dpa.fecha, 'dd MMM yyyy'), '-')
        ), amp.descripcion, dpa.numero_operacion, isnull(dpa.entidad, '-')
        ), dpa.moneda,
        isnull(dpa.tipo_cambio, 0), vfi.precio, dpa.monto_registro, (vfi.precio-dpa.monto_registro)
        from DOC_Pago dpa
```



```
        join V_Files vfi on dpa.file_id = vfi.idfile
        join ADM_MetodoPago amp on amp.id = dpa.metodo_pago_id
        where ".$subconsulta." cast(vfi.inicio as date) between '".\Carbon
\Carbon::parse($entrada["i"])->format('Y-m-
d')." and '".\Carbon\Carbon::parse($entrada["f"])->format('Y-m-d')."
    )
    select * from SeguimientoPagos"));

$agruped = $consulta->groupBy('id');
$data = collect();
foreach($agruped as $file_id => $contenido) {
    $valor_inicial = floatval($contenido[0]->total);
    foreach($contenido as $item) {
        $data->push((object) [
            'counter' => $item->counter,
            'cliente' => $item->cliente,
            'file_id' => $item->file_id,
            'fecha_abono' => $item->fecha_abono,
            'transaccion' => $item->transaccion,
            'nro_operacion' => $item->nro_operacion,
            'entidad' => $item->entidad,
            'moneda' => $item->moneda,
            'tipo_cambio' => floatval($item->tipo_cambio),
            'total' => $valor_inicial,
            'abonado' => floatval($item->abonado),
            'diferencia' => $valor_inicial - floatval($item->abonado),
        ]);
        $valor_inicial = $valor_inicial - floatval($item->abonado);
    }
}

return new PagosExport($data);
```



```
}  
}
```

➤ ANEXO 15: Código fuente – Middleware Validar Login

```
<?php  
  
namespace App\Http\Middleware;  
  
use Closure;  
use Illuminate\Support\Facades\Auth;  
use Illuminate\Support\Facades\Session;  
  
class ValidateLogin  
{  
    public function handle($request, Closure $next)  
    {  
        if(Auth::check())  
        {  
            $cuenta = Auth::user()->cuenta;  
  
            if ($cuenta)  
            {  
                $sistema_contratados = Auth::user()->SistemaContratados();  
                $sistema_sel = $request->session()-  
>get('sistema_sel') == null ? 0 : $request->session()->get('sistema_sel');  
  
                if($request->session()->get('sistema_sel') == null)  
                {  
                    foreach ($sistema_contratados as $index => $contratado)  
                    {  
                        if($index==0)  
                        {
```



```
                $request->session()-  
>put('sistema_sel', $contratado->id);  
                $request->session()-  
>put('sistema_name', $contratado->titulo);  
            }  
            break;  
        }  
    }  
  
    $modulos = Auth::user()->Modulos();  
  
    foreach ($modulos as $adquirido)  
    {  
        foreach ($adquirido->sistema->modulos as $modulo)  
        {  
            $cont_subprivi = 0;  
            foreach ($modulo->submodulos as $submodulo)  
            {  
                if(count($submodulo->privilegios) > 0)  
                    $cont_subprivi ++;  
            }  
  
            $modulo->contsub = $cont_subprivi;  
        }  
    }  
  
    $request->attributes->add(['modulos'=>$modulos]);  
    $request->attributes->  
>add(['contratado'=>$sistema_contratados]);  
  
    if(count($modulos)>0)  
    {
```




```
        return $next($request);
    } else
    {
        Auth::logout();
        Session::flash('login_message', 'No Cuenta con Sistemas Ac
tivos!');
        return redirect()->guest('login');
    }
} else
{
    Auth::logout();
    Session::flash('login_message', 'Su Cuenta esta inhabilitada!'
);
    return redirect()->guest('login');
}

} else
{
    return redirect()->guest('login');
}

return $next($request);
}
}
```

➤ ANEXO 16: Código fuente – Middleware Validar privilegio

```
<?php

namespace App\Http\Middleware;

use Closure;

class ValidatePrivilege
```



```
{  
  
/**  
 * Handle an incoming request.  
 *  
 * @param \Illuminate\Http\Request $request  
 * @param \Closure $next  
 * @return mixed  
 */  
public function handle($request, Closure $next, $submodulo_s)  
{  
    $modulos = $request->attributes->get('modulos');  
  
    $acceso = false;  
    $sistema = null;  
    $modulo_t = '';  
    $submodulo_t = '';  
  
    foreach ($modulos as $contratado)  
    {  
        foreach ($contratado->sistema->modulos as $modulo)  
        {  
            foreach ($modulo->submodulos as $submodulo)  
            {  
                if($submodulo->nombre == $submodulo_s && count($submodulo->  
>privilegios) > 0)  
                {  
                    $acceso=true;  
                    $sistema=$contratado;  
                    $modulo_t=$modulo->titulo;  
                    $submodulo_t=$submodulo->titulo;  
                    break;  
                }  
            }  
        }  
    }  
}
```



```
    }  
  }  
}  
  
if($acceso)  
{  
  $ahora = now();  
  $vencimiento = $sistema->fecha_fin;  
  if($vencimiento->gte($ahora))  
  {  
    $routeName = $request->route()->getActionMethod();  
    $request->attributes->  
>add([ 'modulo_s' => $modulo_t, 'submodulo_s' =>$submodulo_t, 'accion_s' => $r  
outeName]);  
    return $next($request);  
  } else  
  {  
    return redirect('/home');  
  }  
} else {  
  abort(403);  
}  
}
```