



PROYECTO DE GRADO

Presentado ante la ilustre UNIVERSIDAD DE LOS ANDES como requisito final para
obtener el Título de INGENIERO DE SISTEMAS

DESARROLLO DE UN SISTEMA WEB DE CENTRALIZACIÓN DE RESERVAS PARA GRUPOS HOTELEROS

Por

Br. Diego José Perrone Pérez.

Tutor: Gerard Páez, PhD.

Noviembre 2019

©2019 Universidad de Los Andes Mérida, Venezuela

C.C. Reconocimiento

Desarrollo de un Sistema Web de Centralización de Reservas para Grupos Hoteleros

Br. Diego José Perrone Pérez

Proyecto de Grado — Departamento de Sistemas Computacionales, 87 páginas.

Escuela de Ingeniería en Sistemas, Universidad de los Andes, 2019.

Resumen: Los grupos hoteleros son empresas que agrupan una cantidad determinada de hoteles bajo una misma marca y gestión, esto hace que requieran de herramientas de software especializadas, capaces de apoyar todas las funciones que llevan a cabo para su funcionamiento y además, que permitan la conexión con las distintas sedes hoteleras para obtener información de cualquiera de ellas de manera inmediata. En este proyecto de investigación, se presentan las fases de análisis y desarrollo de un sistema de reservaciones construido sobre una plataforma web, conformado por módulos que permiten sean realizadas todas las actividades necesarias para la operación de un grupo hotelero. Las principales acciones que definen el funcionamiento de este sistema son la consulta de disponibilidad, la realización de reservas y la emisión de pagos, para cada una de las cuales fue realizado un módulo que permita su fácil utilización y posterior consulta. Es posible con la utilización de este sistema, aumentar el rendimiento de cada uno de los departamentos que componen este tipo de empresas, así como generar una base de datos que unifique la información producto de todas las transacciones ejecutadas. Estos datos permiten una gestión adecuada del grupo hotelero, optimizando cada uno de sus procesos y permitiendo que se preste un mejor servicio a los clientes.

Palabras clave: Sistema de gestión hotelera, Centralización de reservas, gestión de grupos hoteleros

Índice

Índice	iv
Índice de Tablas.....	vii
Índice de Figuras.....	viii
Agradecimientos.....	x
Capítulo 1 Introducción.....	11
1.1 Antecedentes	12
1.2 Planteamiento del Problema.....	13
1.3 Objetivos	15
1.3.1 Objetivo General	15
1.3.2 Objetivos Específicos	15
1.4 Metodología	16
1.5 Justificación	17
1.6 Alcance	18
Capítulo 2 Marco Teórico	19
2.1 Contexto	19
2.1.1 Grupo Hotelero	19
2.1.2 Software de Gestión Hotelera.....	20
2.1.3 Sistema de Centralización de Reservas	20
2.2 Tecnologías	20
2.2.1 Tecnologías de Información y Comunicación (TIC)	21
2.2.2 Fundamentos de la Web	21
2.2.2.1 El Protocolo HTTP	22
2.2.2.2 Lenguaje HTML	22
2.2.3 Framework	23
2.2.4 Lenguajes de Programación.....	23
2.2.4.1 Typescript	24
2.2.4.2 PHP.....	24
2.2.5 Patrón de Diseño	25

2.2.5.1	Patrón de Diseño MVC (Modelo, Vista, Controlador).....	25
2.2.6	Framework AngularJS	26
2.2.7	Framework Laravel	27
2.3	Bases de Datos.....	28
2.3.1	SQL	28
2.3.2	Manejador MySQL	29
Capítulo 3	Análisis de Requisitos	30
3.1	Levantamiento de la Información	31
3.2	Análisis de la Información	32
3.2.1	Requerimientos No Funcionales del Sistema	32
3.2.1.1	Requerimientos de Conexión	32
3.2.2	Requerimientos Funcionales del Sistema.....	34
3.2.2.1	Módulo de Consultar Disponibilidad	34
3.2.2.2	Módulo de Realizar Reservación	36
3.2.2.3	Módulo de Realizar Pago	37
3.2.2.4	Módulo de Cambio de Parámetros del Sistema.....	38
3.2.2.5	Módulo de Gestión de Usuarios	40
Capítulo 4	Diseño Arquitectónico	41
4.1	Arquitectura de la Aplicación.....	42
4.1.1	Vista.....	43
4.1.2	Modelo	43
4.1.3	Controlador	44
4.2	Diagrama de Clases.....	44
4.3	Diseño de la Interfaz	47
4.3.1	Interfaz de Usuario No Autenticado	47
4.3.2	Interfaz de Usuario Autenticado	48
4.3.2.1	Interfaz del Módulo de Consultar Disponibilidad	49
4.3.2.2	Interfaz del Módulo de Realizar Reservación	50
4.3.2.3	Interfaz del Módulo de Realizar Pagos	51
4.3.2.4	Interfaz del Módulo de Gestión de Usuarios	52
4.3.2.5	Interfaz del Módulo de Cambio de Parámetros del Sistema	53

Capítulo 5	Desarrollo de Versiones.....	55
5.1	Primera Iteración	56
5.2	Segunda Iteración.....	61
5.3	Tercera Iteración	65
Capítulo 6	Pruebas	68
6.1	Objetivo de las pruebas	69
6.2	Técnicas a implementar	69
6.3	Criterios de las pruebas.....	69
6.4	Pruebas Realizadas.....	69
6.4.1	Consultar Disponibilidad en un hotel	69
6.4.1.1	Flujo de la prueba	69
6.4.1.2	Postcondiciones.....	70
6.4.2	Realizar una reservación	71
6.4.2.1	Flujo de la prueba	71
6.4.2.2	Postcondiciones.....	73
6.4.3	Realizar un pago	74
6.4.3.1	Flujo de la prueba	74
6.4.3.2	Postcondiciones.....	76
6.4.4	Gestionar Usuarios	77
6.4.4.1	Flujo de la prueba	77
6.4.4.2	Postcondiciones.....	79
Capítulo 7	Conclusiones y Recomendaciones.....	81
7.1	Conclusiones	81
7.2	Recomendaciones.....	82
Bibliografía.....		84

Índice de Tablas

Tabla 1: Formato de la tabla de descripción de casos de uso.	34
Tabla 2: Caso de Uso: Consultar Disponibilidad.	35
Tabla 3: Caso de Uso: Realizar Reservación.	37
Tabla 4: Caso de Uso: Realizar un Pago.....	38
Tabla 5: Caso de Uso: Cambiar Parámetros del Sistema y Gestión de Usuarios.	39
Tabla 6: Caso de uso: Gestión de Usuarios del Sistema.	40

www.bdigital.ula.ve

C.C. Reconocimiento

Índice de Figuras

Figura 1: Interacción entre cliente y servidor siguiendo el Protocolo HTTP.....	22
Figura 2: Diagrama de Actividad proceso de reservación.	31
Figura 3: Diagrama de Topología de red para conexión entre sedes y servidores principales.	33
Figura 4: Diagrama de caso de uso: Consultar Disponibilidad.	35
Figura 5: Diagrama de caso de uso: Realizar Reservación.	36
Figura 6: Diagrama de caso de uso: Realizar Pago.....	38
Figura 7: Diagrama de caso de uso: Cambiar Parámetros del Sistema	39
Figura 8: Diagrama de caso de uso: Gestión de Usuarios del Sistema.	40
Figura 9: Arquitectura de software del sistema	42
Figura 10: Diagrama de clases.....	45
Figura 11: Diseño de Interfaz de Usuario No Autenticado.	48
Figura 12: Diseño de Interfaz de Usuario Autenticado	49
Figura 13: Diseño de Interfaz: Módulo de Consultar Disponibilidad.	50
Figura 14: Diseño de Interfaz: Módulo de Realizar Reservas.....	51
Figura 15: Diseño de Interfaz: Módulo de Realizar Pagos.	52
Figura 16: Diseño de Interfaz: Módulo de Gestión de Usuarios	53
Figura 17: Diseño de Interfaz: Módulo de Cambio de Parámetros	54
Figura 18: Página de Inicio de Sesión.	57
Figura 19: Menú Principal del Sistema.	57
Figura 20: Interfaz de Módulo: Consultar Disponibilidad.	58
Figura 21: Interfaz de Módulo: Realizar Reservas.....	59
Figura 22: Botón “Añadir al Carrito” para añadir habitaciones a reserva.	59
Figura 23: Interfaz de Módulo: Gestión de Usuarios.	60
Figura 24: Formulario para registrar nuevos usuarios.	60
Figura 25: Interfaz de Módulo: Consulta de Reservas.	61
Figura 26: Consulta de datos de reserva.....	62
Figura 27: Interfaz de Módulo: Realizar Pagos.....	63
Figura 28: Formulario de Pago.....	63

Figura 29: Interfaz de Módulo: Inclusión de formas de pago.	63
Figura 30: Interfaz de Módulo: Consulta de Pagos.	64
Figura 31: Tabla de Consulta de pagos y totalizaciones.	64
Figura 32: Parte de JSON de respuesta proveniente de sistema de gestión <i>Argus</i>	65
Figura 33: Comprobante de Reserva.	66
Figura 34: Comprobante de Pago.	67
Figura 35: Reporte de Reservaciones.	67
Figura 36: Prueba de Verificación de Disponibilidad.	70
Figura 37: Disponibilidad verificada en Sistema de Gestión <i>Argus</i>	71
Figura 38: Flujo de la prueba realizar reserva.	72
Figura 39: Prueba de emisión de comprobante de reserva.....	73
Figura 40: Prueba de consulta de reserva.	73
Figura 41: Prueba de verificación de inserción de reservas en sistema de gestión <i>Argus</i>	74
Figura 42: Porcentaje mínimo de cobro según módulo de configuración	75
Figura 43: Flujo de prueba de realizar pago.	75
Figura 44: Prueba de emisión de recibo de pago.	76
Figura 45: Prueba de consulta de pagos.	76
Figura 46: Prueba de actualización de datos de la reserva al realizar pago.	77
Figura 47: Prueba de Registro de Usuario.	78
Figura 48: Prueba de llenado de formulario para registro de Usuario.	78
Figura 49: Prueba de modificación de usuario.	78
Figura 50: Prueba de Inicio de sesión.	79
Figura 51: Prueba de consulta de usuarios del sistema.....	79

Capítulo 1

Introducción

En la sociedad actual, las nuevas tecnologías han tomado cada vez más importancia en todos los ámbitos de la sociedad, en las empresas, su correcta implementación determinará el futuro y posibilidades de éxito en cualquier modelo de negocios. El camino que ha tomado el desarrollo de aplicaciones bajo la influencia de este fenómeno es hacia la plataforma web, ya que ésta emplea un conjunto de técnicas y tecnologías que ofrecen un gran rendimiento y fiabilidad, así como un mecanismo de navegación que resulta familiar a los usuarios. Básicamente los sistemas web son sistemas informáticos codificados en lenguajes que son soportados por los navegadores web, éstos se alojan en un servidor que puede ser en el internet o en una red local y facilitan la accesibilidad a la información desde cualquier puesto de trabajo.

Históricamente, han sido utilizados sistemas computarizados para dar solución a las necesidades de información de los principales procesos que se realizan en un hotel, la importancia de su uso radica en la relación directa que existe entre la mejora del funcionamiento interno de la empresa y la mejora del servicio ofrecido. Por su parte, aparte del tamaño del hotel, otras características determinan también cuáles son sus necesidades en cuanto a tecnologías de información. Estudios reflejan que aquellos hoteles que pertenecen a cadenas o grupos hoteleros tienen una mayor necesidad de hacer uso de tecnologías para el manejo de información. Así, se destaca el carácter imperativo de la utilización de herramientas de software por parte de este tipo de empresas, es necesario que puedan manejar en un mismo sistema, el inventario conjunto de habitaciones disponibles en tiempo real y debe permitir que se realicen todas aquellas actividades que complementan su operación.

El uso adecuado de tecnologías informáticas es una fuente de ventajas competitivas en los sectores donde el manejo de información juega un papel fundamental en el desempeño de las empresas, permite que se aumente la productividad, mejora significativamente el funcionamiento de los departamentos que conforman la empresa e influye en la reducción de costos. Es por esto, que la

finalidad del presente proyecto de investigación es el desarrollo de un sistema de centralización de reservas para grupos hoteleros, que permita a la empresa realizar cada proceso que compone su operación diaria de manera simple y óptima.

1.1 Antecedentes

En la hotelería actual, existe la inevitable influencia creada por el nuevo entorno empresarial definido por las nuevas tecnologías de manejo de información. A lo largo de esta sección, se analizan y discuten estudios que se han realizado sobre las herramientas de software que han apoyado la innovación turística y han hecho su operación más simple y ordenada.

Figueroa y Talón (2006) en su estudio realizado para medir el efecto que tiene la inversión de las empresas hoteleras en productos de tecnología y comunicación, incluyendo software para reservaciones, destacan que el turismo es una industria compleja cuando se trata del manejo de información y es necesario para realizar la recolección, almacenamiento e interpretación de la misma, el uso de tecnologías específicas que permitan un incremento en la productividad y un mejoramiento de la calidad del control interno de operaciones. Concluyen, que el uso de las herramientas de información y comunicación se ha convertido en un requisito indispensable para el manejo de la información en la industria hotelera, pero el disponer y usar de estas tecnologías no es suficiente si no existe la conectividad entre ellas, convirtiéndose este en un factor de máxima importancia.

El objetivo de la conectividad entre las herramientas, según los mismos autores, debe ser la generación de una base de datos única y compartida, ya que esto genera eficiencia en el proceso de recolección de información, evitándose el coste de tener que introducir la información en varias herramientas y las equivocaciones que esto puede traer. Permite también tener la información en tiempo real agilizando la toma de decisiones y posibilita una mejora en la gestión de la empresa gracias a un mejor conocimiento del cliente y sus preferencias.

Caro (2008) cuyo trabajo de investigación tuvo el objetivo de crear un modelo para explicar el impacto que tienen las Tecnologías de Información y Comunicación (TIC) en el porcentaje de ocupación de empresas hoteleras en México, explica que estas tecnologías proporcionan multitud de ventajas que permiten mejorar el servicio que se ofrece a la clientela en general. Desde el punto de vista

del marketing, dichas tecnologías permiten conocer las preferencias y necesidades de los clientes, ofrecer una mejor entrega del servicio, llegar a un mayor número de clientes potenciales, aumentar la eficiencia y optimizar los recursos en las organizaciones. Concluye que el uso de las TIC incrementa la productividad fruto de los procesos de automatización que permiten que los trabajadores realicen su trabajo de una forma más eficiente, gracias a la reducción en los costes temporales de las distintas tareas que llevan a cabo.

Buhalis y Law (2008) comentan que los hoteles pertenecientes a cadenas o grupos hoteleros, tienden a tener una mayor cantidad de departamentos y servicios, por lo que suelen hacer un uso más intensivo de las TIC en relación a los hoteles pequeños e independientes, estos por lo general realizan gran parte de sus operaciones de forma manual e ineficiente. Una gran diferencia que se encuentra entre hoteles independientes y pertenecientes a cadenas, es la cantidad de información que se debe enviar y recibir para su supervisión, lo que resalta la necesidad que tienen de usar herramientas de software adaptadas a las operaciones que realizan constantemente.

Ruiz, et al. (2012) plantea conocer el nivel de uso de las Tecnologías de Información y Comunicación (TIC) en la Industria Hotelera en la revista Análisis Turístico. Establecen que las empresas turísticas y en concreto los hoteles tienen a su disposición una gran cantidad de herramientas tecnológicas que pueden utilizar tanto para mejorar la eficiencia en sus procesos internos, como para mejorar la calidad del servicio prestado a sus clientes. Indican que la satisfacción de las necesidades y deseos de los clientes no sólo requiere de la inversión en tecnologías de la información, si no también disponer de personal formado en el uso de estas tecnologías, concluyendo finalmente, que las empresas del sector hotelero deben realizar inversiones para implementar TIC que les permitan gestionar de una mejor manera la información generada con su operación y añadir valor a las relaciones con sus clientes, esto a través de un mejor conocimiento de sus preferencias.

1.2 Planteamiento del Problema

El entorno turístico y la gestión de las empresas de este sector se ven sujetos a continuos cambios motivados, en gran parte, al desarrollo que sufre la industria en materia tecnológica. El internet y las nuevas tecnologías han provocado una revolución en la manera de comercializar los servicios turísticos,

tras la aparición de nuevos canales de distribución y la accesibilidad que tienen los potenciales clientes a herramientas de comunicación, es necesario que las empresas se posicen y se adapten para cubrir las necesidades del público, prestar un servicio de calidad con herramientas a la altura de las nuevas exigencias tecnológicas es vital para mantenerse en el actual mercado.

Otro aspecto a tomar en cuenta es la aplicación y el uso de las nuevas tecnologías disponibles para la gestión de las empresas dedicadas al sector turístico, en este caso específicamente la hotelería, generan cambios significativos en los resultados operativos de cada área de trabajo, permiten la optimización de procesos y aumentan la productividad de las distintas unidades de gestión que hacen vida en un hotel. En el caso de los grupos hoteleros, sin embargo, al ser un conjunto de hoteles agrupados bajo una misma marca, que comparten una serie de características que los hace diferenciarse en el mercado y comparten también una gestión administrativa única, hace que la gestión general de este tipo de empresas sea un proceso más extenso y requiera la recolección y análisis de una mayor cantidad de datos, que provienen de herramientas de software utilizadas para la gestión individual de cada una de sus sedes.

Por esto, cuando los grupos hoteleros usan herramientas de software que trabajan de manera independiente en sus sedes, la administración única enfrenta la dificultad de tener la información que debe ser revisada ubicada en bases de datos aisladas. Los elementos de software que no tengan prevista la integración o exportación de los datos generados con su uso, dispersan la información que debe revisar constantemente la empresa para conocer los resultados de su operación y además hace engoroso el proceso de cálculo de ventas, habitaciones vendidas, cálculo de disponibilidad inmediata y otras actividades como el establecimiento de estrategias de mercadeo que pueden causar insatisfacción de clientes o baja calidad del servicio prestado.

Del mismo modo, un mecanismo de ventas y estrategias de mercado manejado de manera independiente, no permite tener una visión amplia y conjunta de todas las posibilidades de la empresa para revertir resultados negativos o insuficientes ventas en cada sede. Adicionalmente, la existencia de departamentos de mercadeo independientes genera un costo agregado al necesitar mayor cantidad de personal y tener que disponer de una cantidad mayor de recursos.

Por otra parte, todos los indicadores de operación tienen que ser constantemente remitidos manualmente para su revisión por parte del personal de auditoría y esto genera retraso en toma de decisiones que pudieran realizarse de manera más oportuna. Una manera más adecuada de llevar a cabo

el control general del grupo hotelero es realizando todas las operaciones como reservaciones, pagos, entre otros, en un mismo módulo de software para todos los hoteles bajo su operación, esto centraliza gran parte de la información y mejora en gran medida los procedimientos realizados por el grupo para su gestión diaria.

De acuerdo a lo anteriormente planteado y ante la innegable necesidad que tienen los grupos hoteleros de tener una herramienta de software personalizada, que les permita obtener la información comercial en tiempo real para realizar sus labores operativas, en la presente investigación se plantea el desarrollo de un sistema web de centralización de reservas, con el fin de permitir a los usuarios realizar las principales actividades de la empresa en una misma plataforma, recolectando así la información que posibilita la adecuada gestión del grupo hotelero.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un Sistema Web de Centralización de Reservas para Grupos Hoteleros.

1.3.2 Objetivos Específicos

- Identificar los procesos que realizan actualmente los grupos hoteleros.
- Determinar los requerimientos básicos para el Sistema Web de Centralización de Reservas para Grupos Hoteleros.
- Describir el modelo de la base de datos tomando en cuenta la integración con el Software de Gestión Hotelera *Argus*.
- Elaborar el Sistema Web de Centralización de Reservas para Grupos Hoteleros.
- Desarrollar un plan de pruebas para comprobar el funcionamiento de los módulos que conforman el Sistema.

1.4 Metodología

La primera fase de ejecución de este proyecto, comprende la recopilación de la información necesaria para realizar el diagnóstico que permitirá definir cuáles son los requerimientos mínimos que el sistema debe cumplir. Esto se realizará mediante la técnica de la observación participante en combinación con entrevistas. En la observación participante el observador es parte de los eventos que están siendo estudiados, está caracterizada porque el investigador se aproxima a los participantes en su propio ambiente y trata de aprender cómo es el desarrollo de las actividades desde la perspectiva de quién experimenta determinadas situaciones (Bernal, 2018).

Se realizarán entrevistas en profundidad para complementar las anotaciones y la información recolectada de la fase de observación, en esta técnica el entrevistador se considera el estudiante y la persona entrevistada el experto, en base a esta nueva información el investigador obtiene una amplia visión del trabajo y de las necesidades de los usuarios (Escalona, 2002). Estos procesos permiten llevar a cabo la actividad de captura de requisitos sobre los cuales se hará el desarrollo del sistema.

La etapa de definición-validación de los requisitos extraídos en el proceso anterior es de naturaleza iterativa, y es posible que sea necesario realizarse varias veces. Esta fase se conoce como planificación y comprende un diálogo permanente entre la parte empresarial, que representa el deseo del cliente y la parte técnica, que está en deber de indicar si es posible llevar a cabo las ideas planteadas.

Luego de la extracción de requisitos necesaria para iniciar el desarrollo del proyecto, se realiza el diseño de los componentes de la aplicación, usando un patrón de arquitectura. Este diseño debe permitir visualizar y validar la interacción entre las entidades o módulos a través de diagramas, pueden ser de clases, de componentes, de despliegue, de secuencia, entre otros.

Para la segunda y última fase, se hará uso de una metodología de desarrollo ágil denominada *SCRUM* (Schwab et al., 2002) la cual está especialmente indicada para proyectos con rápidos cambios de requisitos, el desarrollo del software se realiza mediante iteraciones, en cada una de las cuales se obtiene una versión funcional del producto final, que se va perfeccionando cada vez más gracias al enriquecimiento que se obtiene en el área y la retroalimentación obtenida del usuario. Una de las principales características de este método de desarrollo es las reuniones a lo largo del proyecto, vitales

para verificar el trabajo que se está realizando, el progreso hacia los objetivos y evaluar soluciones a posibles obstáculos que puedan surgir durante la ejecución de la iteración.

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto en los proyectos, pero estas representan un esfuerzo considerable para ser adoptadas sobre todo en proyectos pequeños y requisitos cambiantes, es por esto que las metodologías ágiles ofrecen una solución a medida para este tipo de proyectos. Su fácil implantación y su sencillez se traducen en una reducción del tiempo de desarrollo pero manteniendo siempre una alta calidad del producto generado. (Canós et al., 2012).

Finalmente, se diseña un conjunto de pruebas, para comprobar que el software realice correctamente las tareas programadas. Se pretende probar cada módulo por separado y posteriormente de manera integral, para verificar que los objetivos han sido logrados.

1.5 Justificación

La globalización y la acelerada evolución tecnológica hacen necesario el funcionamiento de los mercados en todo momento del día, y conlleva la necesidad de poder operar desde centros de información donde puedan ser satisfechos los requerimientos del mercado. Para esto, de acuerdo con Falquemberg (2012) los miembros del sector hotelero necesitan diseñar productos y servicios diferenciados y competitivos si desean perdurar en el mercado, para lo cual deben incorporar las nuevas tecnologías disponibles, ya que permiten un fácil acceso a los recursos y contribuyen a mejorar la gestión y aumentar la productividad en los negocios.

Las ventajas que tiene la automatización de procesos son numerosas, podemos encontrar reducción del tiempo utilizado para obtener información, reducción de costos operativos, reducción de la complejidad operativa de una determinada actividad, evitar perdida de información, entre otros. En el caso de este proyecto de investigación, se automatizará la realización de operaciones del grupo hotelero, los procesos relacionados con esta actividad consisten en consultar disponibilidad, inclusión de reservas y pagos, generación de reportes, todos vitales para la aumentar la productividad administrativa del grupo hotelero.

Así, los usuarios al usar el módulo de software realizado, podrán consultar las disponibilidades de forma automática y transparente en cualquiera de las sedes del grupo hotelero, esto permite disminuir de manera importante la dependencia que existe hacia el personal local de la sede. Una vez que sea realizada una reservación, ésta será insertada de manera automática en la sede correspondiente sin la interacción adicional de ningún operador, lo que libra este proceso de retardos innecesarios y facilita en gran medida la operación de cada hotel del grupo, ya que una parte importante de las reservas serán procesadas a través del módulo de software desarrollado.

El agrupamiento de esta información en un software único es de vital importancia para el grupo hotelero ya que permite tener el control de la información de cada sede de manera inmediata, facilitando así los procesos de auditoría y revisión de ventas. Un aspecto a destacar, es la competencia entre los actores del sector turismo, que hacen necesaria la actualización constante de sus herramientas de software, ya que estas brindan el soporte a la gerencia para poder realizar sus diferentes procesos y lo que es más importante, permiten que la empresa preste un mejor servicio a sus clientes.

Por su parte, los departamentos del grupo hotelero dedicados al mercadeo y ventas, al tener una mejor perspectiva del comportamiento del mercado, conociendo de mejor manera las preferencias de los clientes y teniendo el acceso adecuado y necesario a la información, les será posible aumentar la eficiencia, productividad y competitividad de la organización.

1.6 Alcance

Para realizar la implementación de este sistema en una fase posterior, el software gestión hotelera interno de cada sede debe ser *Argus*, sistema distribuido por *Argus Sistemas C.A* en Venezuela, ya que este último exporta e importa información con parámetros que se establecieron como estándar para la herramienta desarrollada. Esto delimita la utilización de esta herramienta a grupos hoteleros que como primer paso, deben utilizar en sus sedes el software de manejo integral *Argus*, posteriormente podrán usar el desarrollo realizado en esta investigación para abandonar el esquema de reservaciones individuales y aisladas, unificando toda la información generada en un mismo módulo de software que les permitirá manejar globalmente las estadísticas generadas.

Capítulo 2

Marco Teórico

Se explican a continuación de manera breve y concreta, términos y conceptos relacionados con la presente investigación para una mejor comprensión de los capítulos posteriores.

2.1 Contexto

2.1.1 Grupo Hotelero

Un grupo hotelero, definido como una alianza horizontal de hoteles bajo una marca común, es una empresa que tiene bajo una única dirección y gerencia un número de hoteles que pueden estar localizados en distintas áreas, se maneja de forma unificada su gestión, promoción, marketing, entre otros. En la búsqueda de un mejor posicionamiento en el mercado, éstos pueden combinar sus recursos, actividades y relaciones, a nivel nacional e internacional, dependiendo de la envergadura del grupo.

La constante transferencia e intercambio de recursos, es considerada como la clave para proteger y mantener la ventaja competitiva en el mercado para el grupo hotelero. Políticas aplicadas sobre los estándares de calidad producen que clientes se sientan identificados con la marca, asociándola con productos con ciertas características y haciendo que sea fácilmente reconocible. Siendo la industria hotelera una industria del servicio con un gran componente humano, que genera un impacto en la experiencia de los clientes, se definen pautas y regulaciones para hacer que la percepción final del producto recibido sea similar en cualquiera de las sedes del grupo.

Estas relaciones son fundamentales para entender la relación cercana que viven los miembros de un grupo hotelero, y la dependencia que existe hacia las herramientas de software, que deben ofrecer la

capacidad de hacer posibles los intercambios de información que la gerencia necesita para llevar a cabo sus funciones.

2.1.2 Software de Gestión Hotelera

Son programas informáticos que facilitan la gestión diaria de las empresas dedicadas al alojamiento turístico. El objetivo de su implementación es el de automatizar ciertos procesos relacionados con la prestación de servicios hoteleros, estos pueden dividirse en dos tipos:

- Gestión de *Back-Office*: Reservaciones de los clientes, gestión comercial, gestión de RRHH, Gestión Contable, entre otros.
- Gestión de *Front-Office*: Actividades de Recepción, Facturación, Auditoria, entre otros.

2.1.3 Sistema de Centralización de Reservas

Es una ubicación central que controla el manejo del inventario total de habitaciones y precios, cuando se trata de un grupo hotelero, tenemos que se debe manejar de forma individual y globalizada según sea el caso. Al actualizarse de forma automática la disponibilidad y precios, es posible desde este lugar y con las herramientas de software adecuadas, realizar ventas, crear ofertas y promociones, tener una comercialización diferenciada por canal, entre otros. La información que genera el uso de este sistema permite controlar la oferta y demanda, optimizar los precios de venta y tener una mejor perspectiva sobre el estado general de la operación de la empresa.

2.2 Tecnologías

El ritmo con el que avanzan las nuevas tecnologías es cada vez más acelerado, así como la aparición de cada vez más herramientas para los desarrolladores web y todo aquel que se dedique a estas áreas de hardware y software. Es necesario para mantenerse actualizado en este nuevo ecosistema digital una revisión de las herramientas ofrecidas actualmente, comparar sus características y en base a esto seleccionar las que se consideren ofrezcan una mayor cantidad de ventajas en cuanto a tiempo de desarrollo y robustez de la aplicación creada. A continuación se realiza un recorrido por los principales

conceptos y se detallan las tecnologías seleccionadas para el desarrollo de cada área de la presente investigación.

2.2.1 Tecnologías de Información y Comunicación (TIC)

Cabero (1998) indica la siguiente definición: “*En líneas generales podríamos decir que las nuevas tecnologías de la información y comunicación son las que giran en torno a tres medio básicos: la informática, la microelectrónica y las telecomunicaciones; pero giran, no solo de forma aislada, sino lo que es más significativo de manera interactiva e interconectada, lo que permite conseguir nuevas realidades comunicativas*”.

Extendiendo un poco esta idea, son un conjunto de recursos, procedimientos y técnicas usadas en el procesamiento, almacenamiento y transmisión de la información. Existen múltiples dispositivos electrónicos que pueden encajar dentro de la definición de las TIC como por ejemplo, la televisión, teléfono celular, entre otros, pero sin lugar a dudas los medios más representativos son los ordenadores y las redes comunicacionales, que nos permiten tener acceso a la información, realizar una amplia gama de actividades y utilizar distintos programas informáticos que hacen más fácil las tareas cotidianas. (Belloch, 2012).

Al mismo tiempo, las TIC son excelentes herramientas de gestión empresarial, ayudan positivamente para el desarrollo y viabilidad de las organizaciones, su implementación en el comercio electrónico han transformado el comercio global y reforzado la competitividad empresarial, permitiendo que se mejore la productividad, la calidad de los servicios prestados y se facilite la comunicación y el manejo de la información en general.

2.2.2 Fundamentos de la Web

Hoy día acceder a la web se ha convertido en una actividad rutinaria, necesaria. La web es un servicio que permite que millones de personas accedan a una multitud de funciones y prestaciones, así como a infinidad de servicios, entretenimiento, tiendas, programas, entre otros. Su funcionamiento está basado en dos puntos fundamentales: el lenguaje HTML y el protocolo HTTP. Uno implementa un sistema de comunicaciones que nos permite intercambiar peticiones de manera sencilla, y el otro presenta un mecanismo de composición de páginas web usando etiquetas altamente eficientes. Estos principios aunque se trata de páginas web de acceso público, se extienden a las aplicaciones web de uso privado, es decir en una red local, como el presentado en este caso de investigación.

2.2.2.1 El Protocolo HTTP

Por sus siglas en inglés *Hypertext Transfer Protocol*, es el protocolo base de la *World Wide Web (WWW)*, se trata de un protocolo simple que establece un canal de comunicación de extremo a extremo, en este caso entre un cliente y un servidor, por el cual transitará un flujo de *bytes* que constituyen los datos de la comunicación. En este protocolo, cada transferencia de datos es una nueva conexión independiente de cualquier otra que haya existido.

De manera esquemática, el funcionamiento del protocolo HTTP se resume de la siguiente manera: el cliente realiza una conexión con el servidor, envía un comando HTTP de petición de recurso y el servidor responde por la misma vía con los datos solicitados y alguna información adicional, como se observa en la figura 1.

Las principales directivas de petición del protocolo son las siguientes:

- GET: Petición de recurso
- POST: Petición de recurso con pase de parámetros
- HEAD: Petición de datos sobre recurso
- PUT: Creación de recurso
- DELETE: Eliminación de recurso

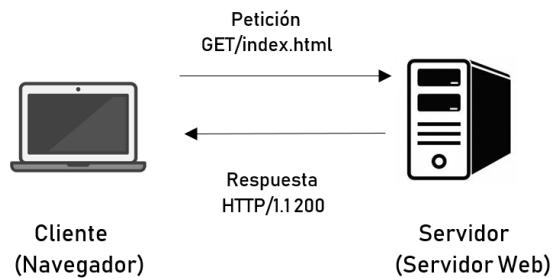


Figura 1: Interacción entre cliente y servidor siguiendo el Protocolo HTTP.

2.2.2.2 Lenguaje HTML

Consiste en un lenguaje de marcas o etiquetas que nos permite construir un elemento de hipertexto *HTML* o *página web* cuya visualización es posible en diferentes navegadores. Este término

aborda la creación de sitios web para el internet o para la intranet, como en el caso de la presente investigación. En esencia, este lenguaje permite estructurar documentos, en forma de títulos, párrafos, tablas, listas, entre otras herramientas que sirven para dar formato y presentar un contenido claro y bien estructurado, que resulte fácil de leer y entender para un usuario.

Existen dos aspectos importantes al realizar una página web, el lado del cliente que se denomina desarrollo *Front-End* y el lado del servidor denominado *Back-End*, para ello se usan tecnologías de software específicas dependiendo del área que sea. El desarrollo en este lenguaje provee un producto homogéneo y multiplataforma, según los resultados de estudios empíricos, se pudo conocer que los tiempos de desarrollo de proyectos de este tipo son cortos (McDonald et al, 2001), siendo el promedio encontrado menor a tres meses en la mayoría de casos para obtener versiones funcionales de los productos.

2.2.3 Framework

Este término se refiere a una estructura la cual contiene una serie de técnicas y elementos personalizables que funcionan como herramientas estratégicas para apoyar a los desarrolladores de software, estas proveen de ventajas como acelerar el proceso de desarrollo de aplicaciones web, permite reutilizar código de manera eficiente, además de añadir seguridad, robustez y facilidades de uso a la aplicación desarrollada. Asimismo, un *framework* sigue una arquitectura definida, MVC por lo general, que será explicada más en detalle posteriormente, la cual define un entorno de trabajo que establece una organización determinada para los archivos según la función a realizar. Funciona como una guía para el diseño de arquitecturas de aplicaciones, plantea muchas ventajas y es capaz de hacer posible que se realicen tareas de manera casi automática, que de otra manera serían difíciles de plantearse llevar a cabo, o llevaría una cantidad mayor de tiempo para ser culminadas de manera exitosa.

2.2.4 Lenguajes de Programación

Son lenguajes artificiales que se utilizan para expresar secuencias de órdenes y algoritmos, siguen una serie de reglas semánticas y sintácticas propias del lenguaje y permiten controlar el comportamiento lógico de un módulo de software, para que realice determinadas tareas según los requisitos planteados

por el programador. A continuación se presenta descripción de los lenguajes de programación que fueron utilizados como parte del desarrollo de esta investigación.

2.2.4.1 **TypeScript**

Este es un lenguaje de programación que comprende una extensión de JavaScript destinado a permitir un desarrollo más rápido de aplicaciones de gran envergadura, es un lenguaje tipado, lo que significa que para cada operación debe definirse para qué tipos de datos es aplicable. Pone a disposición de los desarrolladores una gran cantidad de librerías y frameworks que en conjunto con herramientas avanzadas permite el desarrollo de aplicaciones robustas, de esta manera se eliminan muchos de los problemas documentados de JavaScript y se proporciona una opción más fiable a los desarrolladores. Principalmente es utilizado para programar aplicaciones del lado del cliente, es decir el *front-end* del desarrollo, esto porque a ello se enfocan los frameworks que hacen uso de este lenguaje de programación.

2.2.4.2 **PHP**

Es un lenguaje de programación imperativo, dinámicamente tipado, con una cantidad estándar de tipos de datos predefinidos que está enfocado al desarrollo de aplicaciones del lado del servidor. Es de código abierto y de uso gratuito para los programadores, se encuentra entre los lenguajes de programación más populares entre los desarrolladores, ubicándose entre los 10 primeros puestos al 2010 (TIOBE index) y según estimaciones (W3techs) cerca del 80% de los servidores en internet, de los cuales se puede conocer su lenguaje implementado, utilizan *PHP* para sus aplicaciones de *Back-End*.

Una de las características más potentes y destacables de *PHP* es su soporte para un amplio abanico de tipos de bases de datos, su esquema de funcionamiento se basa en responder solicitudes que sean generadas en páginas web, éstas pueden ser de distintos tipos por ejemplo inserción, modificación o eliminación de la información y se relacionan directamente con las acciones que los usuarios llevan a cabo al usar la aplicación.

2.2.5 Patrón de Diseño

El diseño y análisis de software debe llevarse a cabo de manera sistemática y no de forma artesanal, el uso de métodos y técnicas existentes para llevar a cabo estas actividades permite disminuir el tiempo de desarrollo, mejorar la calidad del producto, tener una mayor reutilización del código y facilitar el mantenimiento de los sistemas desarrollados. Los modelos o patrones de diseño se muestran como una guía para quien realiza el diseño y desarrollo de un sistema, son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño e interacción de interfaces. En otras palabras, permite establecer explícitamente un enlace entre un problema y la existencia de una “mejor práctica” que produce la solución del mismo.

Existen tres tipos de patrones de diseño que podemos utilizar dependiendo del tipo de problema para el que ofrecen una solución:

- Patrones Creacionales: Inicialización y creación de objetos.
- Patrones Estructurales: Se ocupan de cómo las clases y los objetos se agrupan según su función.
- Patrones de Comportamiento: Describen la comunicación entre las instancias.

2.2.5.1 Patrón de Diseño MVC (Modelo, Vista, Controlador)

Sagredo, et al. (2012) indica que este patrón fue diseñado para reducir el esfuerzo y tiempo necesario para la implementación de sistemas. Este patrón considera en separar en tres capas los elementos que conforman el proyecto, detalladas a continuación:

- Capa Modelo: son los objetos que representan los datos e información del sistema.
- Capa Vista: maneja la presentación visual de los datos representados por el modelo, define cómo interactúa el usuario con la interfaz.
- Capa Controlador: está a cargo de la interacción desde y hacia la base de datos, así como del mapeo de las vistas, de acuerdo a las solicitudes del usuario. Representa el enlace entre las vistas y los modelos.

La separación que existe entre los componentes lo hace deseable para proyectos con gran cantidad de archivos, debido a que pueden construirse las piezas de un sistema por separado y luego unirse, o si se encuentra que alguno de los módulos tiene un mal funcionamiento, puede ser reemplazado sin que los demás componentes se vean afectados. Al usar este patrón de diseño se garantiza que se cumplan requisitos mínimos de calidad, éstos atienden diversos parámetros como la estructuración de los programas y la reutilización del código, que influirá positivamente en la facilidad del desarrollo y su posterior mantenimiento.

Por otro lado, el patrón MVC puede ser implementado sin utilizar ningún *framework* específico, pero no es una práctica recomendada ya que éstos lo implementan de forma implícita y deja de ser un trabajo que realiza el programador de manera manual, proporcionan una base sólida para la construcción de las aplicaciones ya que vienen preparados con toda la estructura y funciones necesarias para desarrollar casi cualquier tipo de aplicación. Es por esta razón y las anteriormente descritas que se usaron dos *frameworks* para el desarrollo de este proyecto que serán descritos posteriormente en este capítulo.

2.2.6 Framework AngularJS

AngularJS es, en síntesis, un *framework* que está basado en el lenguaje *TypeScript* cuyo objetivo es la creación de aplicaciones web dinámicas y eficientes. Gracias a su capacidad y velocidad, es uno de los más populares entre las empresas y emprendedores; ofrece una alternativa de desarrollo rápida, escalable y efectiva (Solis, 2015). Permite ahorrar tiempo y al mismo tiempo cumplir con la demanda de calidad que es necesaria para tener un software robusto y organizado.

Una de sus principales características es el *data binding* que consiste en la unión en tiempo real de los datos dos o más elementos, así, si uno de ellos cambia su valor, también lo harán aquellos que están asociados a él. Esta técnica es sumamente útil para la realización de cálculos y para representar de forma gráfica las interacciones del usuario. Otros principales rasgos distintivos que pueden nombrarse

de este *framework* son la inyección de dependencias, las directivas que pueden aplicarse, sus eventos programables, entre otros.

De esta manera, debido a su popularidad y a la cantidad de material documental que puede encontrarse, gracias a su capacidad de ahorrar tiempo al escribir código y a su innegable capacidad para mantener sus cualidades aun cuando los desarrollos crecen en tamaño y complejidad, además de las características destacadas anteriormente, fue escogido como marco de trabajo para ocupar el *front-end* del desarrollo llevado a cabo en esta investigación.

2.2.7 Framework Laravel

Es un poderoso y flexible *framework* de *PHP* basado en el patrón MVC cuya filosofía es el desarrollo del código de una manera elegante y ordenada. Es de código abierto y define un amplio conjunto de objetos y conceptos estandarizados que dan solución a problemas o circunstancias al momento de programar, permite usar y ensamblar componentes sobre una capa bien estructurada sobre la cual pueden construirse las aplicaciones web de una manera práctica y eficiente.

Laravel ofrece una herramienta integrada a la terminal de comandos conocida como *Artisan*, permite realizar la mayoría de las tareas repetitivas y tediosas que los desarrolladores prefieren evitar realizar de forma manual. Da acceso a muchas de las principales funciones del *framework* de manera sencilla y rápida, incluye generación de archivos, creación de estructuras de bases de datos, entre otros. Su sistema de migraciones ayuda a los desarrolladores a generar las bases de datos de manera instantánea cada vez que sea necesario, sin necesidad de hacerlo en el manejador que se utilice en el proyecto.

Otra de sus virtudes es la seguridad, Laravel ofrece seguridad ante las más comunes vulnerabilidades como inyecciones *SQL*, ataques *CSRF* que consisten en falsificación de peticiones, por nombrar algunas. Finalmente, tomando en cuenta otra de sus principales distinciones que es la velocidad de su curva de aprendizaje y debido al gran soporte que existe tanto en documentación como en la comunidad de desarrolladores, se convierte en una de las mejores opciones en el mercado para el desarrollo de aplicaciones web a gran escala, seleccionado así como responsable del desarrollo de todo lo relacionado con el *Back-End* de este proyecto.

2.3 Bases de Datos

Un sistema de base de datos, es básicamente un sistema computarizado para llevar registros, cuya finalidad general es almacenar información y permitir que usuarios interactúen con ella mediante peticiones (Date, 2001). Para llevar a cabo estas funciones se hace uso de un sistema gestor de base de datos (SGBD) los cuales administran el acceso de los usuarios a la información, así como proporcionan métodos para mantener la integridad de los datos y permiten la recuperación de los mismos en caso de fallas. Funcionan como interfaz entre las aplicaciones y los datos que se almacenan, conformando un entorno eficiente al momento de almacenar y recopilar la información. (Cobo, 2007).

A pesar de la gran cantidad de modelos de datos que existen para representar la información, ninguno como el modelo relacional de las bases de datos ha sido adoptado de una manera tan categórica y unánime, ocupando los primeros puestos de estadísticas de uso de los principales motores de bases de datos a nivel mundial (DB-Engines Ranking). Este modelo organiza los datos en tablas bidimensionales, compuestas de filas y columnas, que luego de pasar por un proceso de abstracción y modelado, nos permite representar la realidad de manera correcta y encontrar una estructura adecuada para almacenar la información.

2.3.1 SQL

Es un sub-lenguaje orientado a datos para acceder bases de datos relacionadas que son manejadas por Sistemas de manejo de Bases de datos relacionales por sus siglas en inglés (RDBMS). Sus siglas significan de forma no oficial *Structured Query Language*, cuya traducción corresponde con “Lenguaje Estructurado para Consultas”, básicamente. (Melton et al. 1993). De su carácter de sub-lenguaje, necesita la asociación con otro lenguaje especializado para acceder efectivamente la información, un ejemplo de implementación en términos básicos sería cuando SQL realiza una consulta y entrega esta información al otro lenguaje para llevársela al usuario.

SQL es el lenguaje estándar internacional para crear y mantener bases de datos relacionales, es la base sobre la cual se ejecutan la mayoría de bases de datos de gran escala hoy día, exceptuando las bases de datos no relacionales que manejan los sitios con características que les impiden usar este

esquema. Permite a los usuarios regresar, analizar, crear, administrar y eliminar datos dentro de una base de datos, todo a través de sentencias que realizan estos procedimientos de forma rápida y eficiente.

2.3.2 Manejador MySQL

Pertenece al grupo de sistemas de gestión de bases de datos relacionales (RDBMS), es de código abierto y está basado en lenguaje *SQL*. Permite a aplicaciones desarrolladas en distintos lenguajes de programación acceder a las tablas de su base de datos, a través de una interfaz específica. Dichas tablas, alojan los datos de manera separada y ordenada y están conectadas mediante relaciones que hacen posible una combinación de datos según la naturaleza de las consultas recibidas. Para el año 2013 era el segundo manejador de bases de datos con más uso del mundo, lo que deja claro su reputación y su confiabilidad. En la presente investigación será el encargado de contener la base de datos del desarrollo planteado, en conjunto con el *framework* Laravel, quién realiza las veces de intermediario entre las acciones del usuario y la base de datos.

Capítulo 3

Análisis de Requisitos

Una aplicación de software se desarrolla a través de un proceso, no es algo que se fabrica a partir de materia prima, ni se ensambla a partir de piezas prefabricadas. Independientemente del modelo de proceso adoptado, el proyecto de construcción de un software involucra diversas áreas de conocimiento, que se utilizan en mayor o menor medida en todas las fases de su desarrollo. Así, luego de este proceso de desarrollo, se espera obtener una solución para el cliente que permita que los objetivos de negocio se cumplan en función de calidad y funcionalidad.

De esta manera Pressman, (2002) establece que un aspecto fundamental para este fin es la ingeniería de requisitos de software, este es un proceso de descubrimiento, refinamiento, modelado y especificación en cuanto se refiere a los requisitos que definen lo que el sistema debe realizar y bajo qué circunstancias debe hacerlo, para satisfacer las necesidades identificadas. Por otro lado, Cardozzo (2016) indica que una solución para mapear cada funcionalidad implementada es a través de los casos de uso, estos son diagramas que sirven para especificar la comunicación y el comportamiento de un sistema a medida que se produce interacción con los usuarios. Describe además las condiciones previas necesarias para la correcta ejecución de las distintas funciones y las post-condiciones generadas, que son los resultados de las medidas que se han adoptado.

En este capítulo se presentan de manera resumida los aspectos más relevantes del proceso de ingeniería de requisitos, explicando cada una de las características del sistema, detallando de cada una de ellas, su alcance, funcionalidad, entre otros. Además, se detallan los requisitos no funcionales del sistema, los cuales no hacen referencia a funciones explícitas que deba incluir el sistema, sino características generales de funcionamiento, son condiciones que deben cumplirse para que el sistema funcione correctamente.

3.1 Levantamiento de la Información

Para lograr entender la dinámica de negocio de las labores llevadas a cabo en un grupo hotelero, y el papel que debe desempeñar el sistema desarrollado, se realizó un levantamiento de información a fin de diagnosticar los distintos procesos que llevan a cabo.

Así, el proceso de reserva comienza con la solicitud por parte del cliente de la misma, este deberá indicar para qué sede del grupo hotelero desea el alojamiento, por cuantos días y para cuantas personas. El agente de reservas con esta información realiza la consulta e indica si es posible continuar el proceso, dependiendo de la disponibilidad y las fechas seleccionadas. En caso positivo, se toman los datos personales y de contacto del cliente para llenar el formulario de reserva y al final de este proceso se proporcionan los datos al cliente para que realice el pago. La reserva tiene una vigencia determinada en espera del pago correspondiente, al recibir este último se culmina el proceso para el operador. En la figura 2 encontramos un diagrama de actividad que permite mostrar visualmente el modelado del proceso explicado anteriormente.

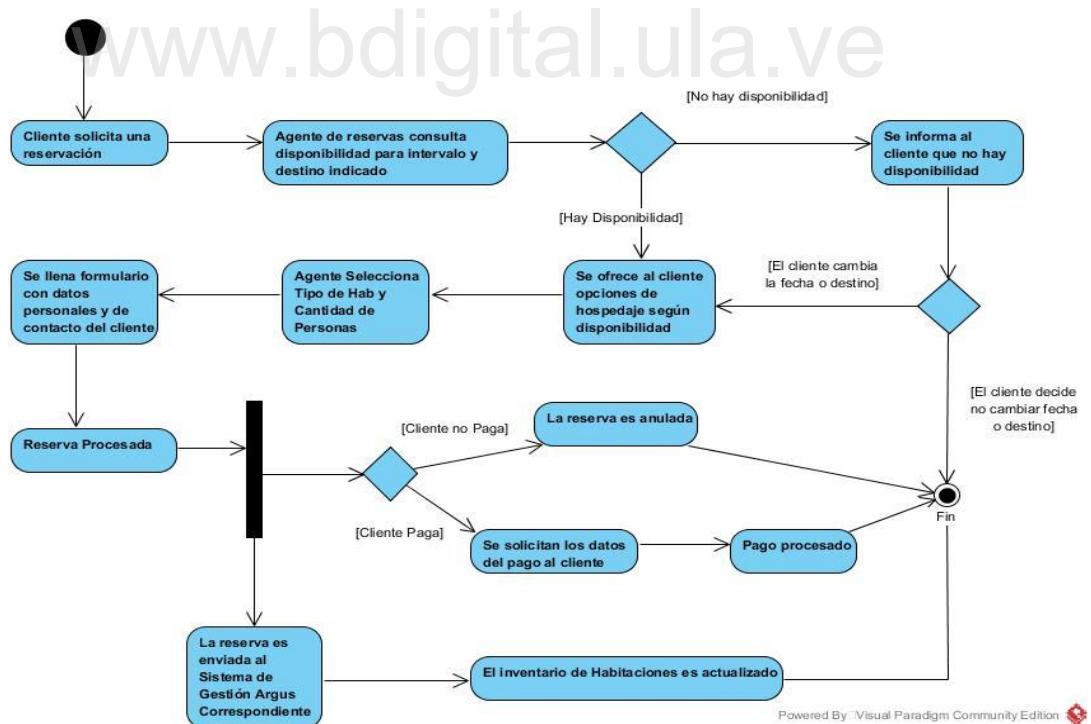


Figura 2: Diagrama de Actividad proceso de reservación.

De esta manera, de cada operación básica realizada se lleva un registro para su posterior seguimiento y control, esto puede visualizarse a manera de reporte, las reservas que aún no han sido pagadas por el cliente deben ser diferenciables de las que sí lo han hecho. Así como los pagos recibidos deben poder ser consultados, para que el departamento encargado realice la verificación de los mismos. Todas estas operaciones deben ser evaluadas por la gerencia del grupo cada cierto tiempo, para determinar a través de la información recolectada el estado de las operaciones, así como para tomar decisiones en cuanto a la búsqueda del bienestar de la empresa.

Por último, existe una figura que ejerce el rol de administrador de la red que debe tener la capacidad de realizar acciones en el sistema que los usuarios convencionales no tienen, como lo son la creación de usuarios, los cambios de los parámetros de configuración, entre otros. Esta figura deberá tener además acceso a todos los módulos y funciones del sistema para poder detectar fallas y verificar que se llevan a cabo todos los procesos correctamente.

3.2 Análisis de la Información

Luego del levantamiento de información realizado, se inició el proceso de análisis de la misma, en el cual se definieron las funciones del sistema, estas pueden incluir cálculos, detalles técnicos, manipulación de datos, entre otros. Dando forma de esta manera a los requisitos funcionales y no funcionales del sistema presentados a continuación.

3.2.1 Requerimientos No Funcionales del Sistema

Se trata de requerimientos que no expresan directamente funciones suministradas por el sistema, si no propiedades que debe poseer el sistema. De manera alternativa, también describen restricciones que deben cumplirse para el satisfacer las necesidades de los usuarios, estas pueden ser de tipo organizativo, de intercomunicación con otros sistemas, de seguridad, de disponibilidad, entre otros.

3.2.1.1 Requerimientos de Conexión

Incluso las aplicaciones realizadas con las más innovadoras tecnologías de desarrollo pueden sufrir interrupciones o fallar debido a un problema de conexión, en el caso de este sistema este es el principal requerimiento no funcional. Para poder garantizar el rendimiento, el tiempo de respuesta y la confiabilidad, es de vital importancia que la infraestructura de comunicación entre las sedes y el centro

de reservas donde se implemente este sistema, permita una conexión segura, estable y suficiente para el tráfico que se genera.

De esta manera, utilizando una tecnología *VPN* o red privada virtual, es posible extender de manera segura una red local (*LAN*) sobre una red pública o no controlada, como lo es el internet. Esta tecnología permite que se envíen y reciban paquetes sobre redes públicas con todas las características de seguridad y funcionalidad que ofrece una red privada. Para este fin, es necesario que cada en cada sede hotelera se instale un enrutador que permita realizar túneles *VPN*, así como que posea acceso a internet, esto garantiza que todas las solicitudes de entre el sistema y las distintas sedes puedan ser satisfechas, sean de consulta de disponibilidad o de escritura, en el caso de las nuevas reservas que deben ser insertadas en cada sistema de gestión hotelero correspondiente. En la figura 3 se puede observar un diagrama que permite visualizar la configuración de red mencionada anteriormente.

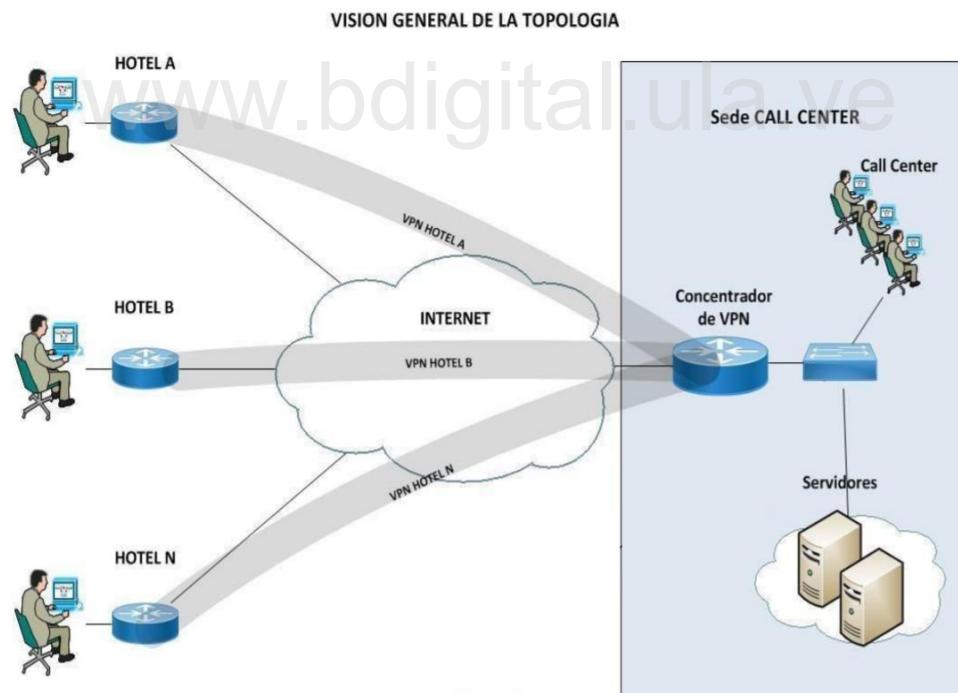


Figura 3: Diagrama de Topología de red para conexión entre sedes y servidores principales.

3.2.2 Requerimientos Funcionales del Sistema

Los requisitos funcionales del Sistema Web de Centralización de reservas se han dividido según el tipo de operación que se realizan, estos son: consultar disponibilidad, realizar y consultar reservas, realizar y consultar pagos y por último la especificación del módulo que permite al administrador de la red realizar cambios en los parámetros de control del sistema. Se muestran a continuación los diagramas de caso de uso correspondientes a cada requisito funcional, acompañados de una tabla de especificación de requisitos (ver Tabla 1) que permiten detallar las interacciones entre el sistema y los distintos actores.

Tabla 1: Formato de la tabla de descripción de casos de uso.

Nombre:	Nombre del caso de uso.
Descripción:	Descripción del caso de uso.
Actores Participantes:	Lista de actores que participan.
Condiciones de Entrada:	Precondiciones para que se ejecute correctamente el caso de uso.
Flujo de Eventos:	Flujo normal de eventos que se necesitan para ejecutar el caso de uso.
Flujo Alternativo:	Flujo alternativo de eventos para ejecutar el caso de uso.
Condiciones de Salida:	Post-condiciones luego de ejecutarse el caso de uso.

3.2.2.1 Módulo de Consultar Disponibilidad

Un usuario autenticado debe poder consultar el inventario de habitaciones disponibles en cualquiera de las sedes del grupo hotelero de manera sencilla y en tiempo real, para esto, es necesario indicar los parámetros requeridos para realizar la consulta, como lo son fecha de entrada y salida, así como el destino deseado por el cliente. Esta operación es el primer paso del proceso de reservas, así como la operación primaria básica sobre la cual se ejecuta el proceso de reservación. En la figura 4 se puede observar el diagrama de caso de uso donde el usuario solicita la información referente al inventario de habitaciones de una sede hotelera en específico, en el intervalo de fechas indicado y, por último, en la tabla 2 se define este requerimiento.



Figura 4: Diagrama de caso de uso: Consultar Disponibilidad.

Tabla 2: Caso de Uso: Consultar Disponibilidad.

Nombre:	Consultar Disponibilidad
Descripción:	El usuario puede consultar el inventario de habitaciones disponibles, luego de ingresar todos los datos requeridos para la operación
Actores:	Usuario Autenticado, Sistema de Gestión Argus
Precondiciones:	El actor debe estar ubicado en el módulo Consultar Disponibilidad
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el hotel que desea consultar 2. El usuario ingresa la fecha de entrada y salida 3. El usuario presiona el botón consultar
Flujo Alternativo:	<p>1.A El usuario selecciona un hotel que no se encuentra en línea, no es posible realizar la consulta</p> <p>2.A El usuario ingresa una combinación de fechas que no es admitida por la validación del sistema.</p>
Post-condiciones	La tabla que contiene la información es generada para su visualización

3.2.2.2 Módulo de Realizar Reservación

Un usuario autenticado en el Sistema debe poder realizar una nueva reservación, para esto debe estar ubicado en el módulo indicado y seleccionar el hotel solicitado por el cliente. Es necesario repetir el proceso de consultar disponibilidad, porque en este caso se muestra en la tabla conteniente de los datos, un botón para permitir que el agente de reservas añada las habitaciones solicitadas por el cliente a la reserva. Posteriormente, se solicitan datos personales y de contacto del cliente para llenar el formulario de reserva, una vez realizada y almacenada por el sistema, debe poder ser consultada por el agente además. En la figura 5 podemos observar el diagrama de caso de uso correspondiente y en la tabla 3 la especificación de este requerimiento.

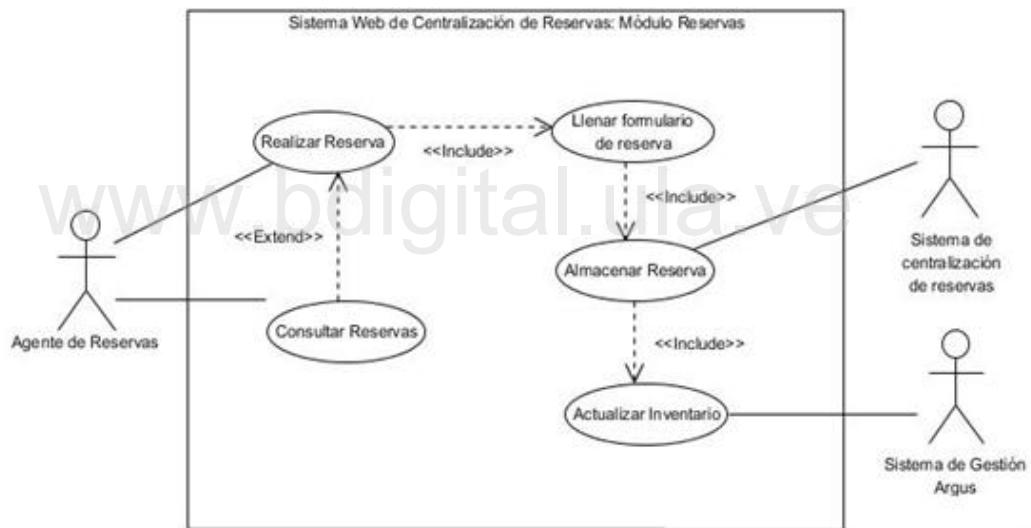


Figura 5: Diagrama de caso de uso: Realizar Reservación.

Tabla 3: Caso de Uso: Realizar Reservación.

Nombre:	Realizar una Reserva
Descripción:	El usuario puede realizar una nueva reservación.
Actores:	Usuario Autenticado, Sistema de Gestión Argus
Precondiciones:	El usuario debe haber consultado que existe disponibilidad para reservar para ese intervalo de fechas en el hotel indicado.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario debe consultar la disponibilidad del hotel deseado (Ver figura 4) 2. El usuario debe añadir las habitaciones deseadas a la reserva 3. El usuario debe ingresar los datos personales y de contacto del cliente (Identificación, Nombre y Apellido, Teléfonos, email, entre otros) 4. El usuario presiona el botón Reservar.
Flujo Alternativo:	3.A El usuario ingresa datos que no cumplen con el formato del sistema (email, teléfono) se indica que debe ingresar los datos correctamente
Post-condiciones	La reservación es almacenada y enviada al software de gestión Argus, así como se genera un comprobante en PDF para su impresión.

3.2.2.3 Módulo de Realizar Pago

Un usuario del sistema debe poder realizar un pago para asociarlo a una reserva, para ello será requerido en el módulo correspondiente, seleccionar la reserva que desea que contenga el pago y posteriormente llenar el formulario de pago. Los datos solicitados tienen que ver con el método de pago, moneda, número de transacción en caso de ser transferencia electrónica, y monto. Existe una validación que debe realizar el sistema sobre el monto de la operación, que es definido por la gerencia de la empresa, esta es el monto mínimo a cancelar, este parámetro es modificable por el administrador de la red en el módulo de configuraciones. En la figura 6 podemos observar el diagrama de caso de uso para este requisito y en la tabla 4 la definición del mismo.

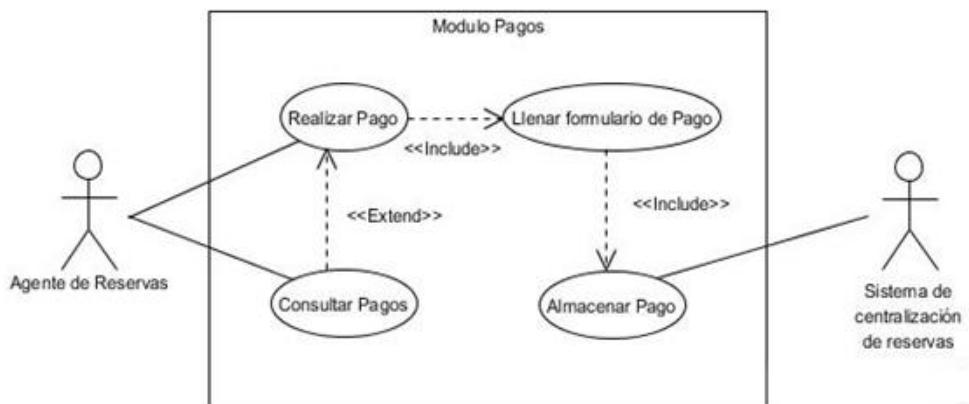


Figura 6: Diagrama de caso de uso: Realizar Pago.

Tabla 4: Caso de Uso: Realizar un Pago.

Nombre:	Realizar un Pago
Descripción:	El usuario puede realizar un nuevo pago.
Actores:	Usuario Autenticado, Sistema de Gestión Argus
Precondiciones:	El pago se realiza sobre una reserva almacenada
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona a qué reserva desea añadirle un pago 2. El usuario llena el formulario de pago (Moneda, Método de pago, Número de transacción, monto, comentarios) 3. El usuario presiona el botón Guardar
Flujo Alternativo:	2.A El usuario ingresa un monto que no cumple las restricciones del sistema, se solicita sea ingresado nuevamente
Post-condiciones:	El pago es almacenado y un comprobante en PDF es generado para su impresión

3.2.2.4 Módulo de Cambio de Parámetros del Sistema

Un usuario, con privilegios de administrador, deberá poder realizar cambios en los parámetros del sistema, como lo son crear y modificar métodos de pago, añadir nuevos hoteles al sistema, así como modificar los existentes, entre otros. Un parámetro que deberá poder ser cambiado en este módulo es

el porcentaje mínimo de pago para una reserva, esto funciona como un requisito para que el cliente cancele al menos una parte del valor de la reserva para confirmar su estadía. En la figura 7 se puede observar el diagrama de caso de uso correspondiente y en la tabla 5 la especificación de este requisito.

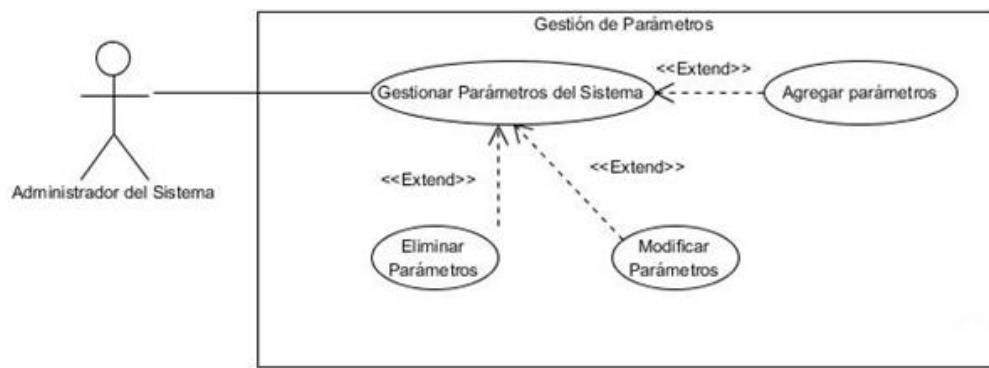


Figura 7: Diagrama de caso de uso: Cambiar Parámetros del Sistema

Tabla 5: Caso de Uso: Cambiar Parámetros del Sistema y Gestión de Usuarios.

Nombre:	Cambiar parámetros del sistema
Descripción:	El usuario puede realizar cambios en los parámetros del sistema, así como añadir nuevos.
Actores:	Usuario Autenticado,
Precondiciones:	El usuario debe poseer privilegios de administrador
Flujo Normal:	<ol style="list-style-type: none"> 4. El usuario selecciona que parámetro desea modificar 5. El usuario llena el formulario del parámetro deseado, en el caso de modificación sólo debe modificar el campo correspondiente 6. El usuario presiona el botón Guardar
Flujo Alternativo:	
Post-condiciones	Los parámetros del sistema se actualizan en la base de datos.

3.2.2.5 Módulo de Gestión de Usuarios

Un usuario con privilegios de administrador debe poder gestionar los usuarios que tienen acceso al sistema, deben poder crearse nuevos usuarios, así como modificar los existentes y eliminar aquellos que se no desea tengan acceso al sistema. En la figura 8 y la tabla 6 se define este requerimiento.

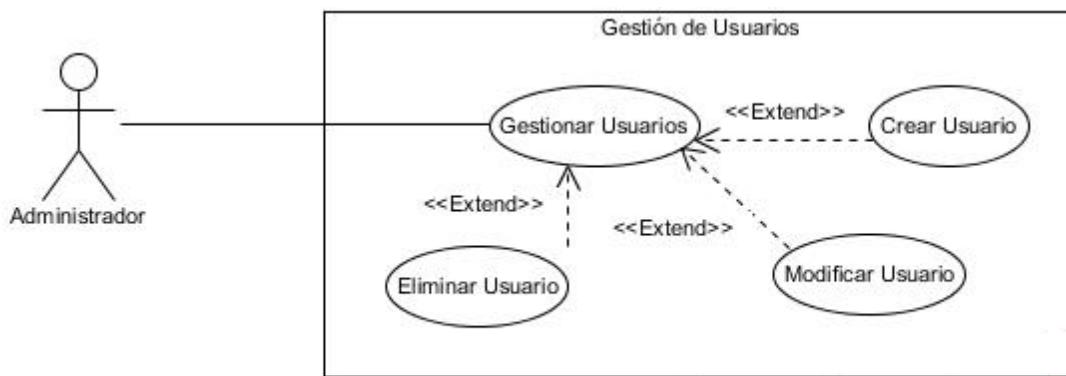


Figura 8: Diagrama de caso de uso: Gestión de Usuarios del Sistema.

Tabla 6: Caso de uso: Gestión de Usuarios del Sistema.

Nombre:	Gestión de usuarios del Sistema
Descripción: El usuario puede gestionar los usuarios del sistema.	
Actores:	Usuario Autenticado
Precondiciones:	El usuario debe poseer privilegios de administrador
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario selecciona un usuario que deseé modificar 2. El usuario cambia el valor deseado 3. El usuario presiona guardar 4. En caso de crear un nuevo usuario, selecciona la opción Ingresar nuevo Usuario 5. Luego de llenar el formulario, presiona el botón crear.
Post-condiciones	El nuevo usuario puede acceder al sistema. En caso de modificación el sistema almacena los nuevos datos ingresados.

Capítulo 4

Diseño Arquitectónico

El diseño arquitectónico del software se define como el conjunto de procesos técnicos que describen la arquitectura que deberá tener la aplicación, de acuerdo con lo expuesto por Bass (2006), es la estructura de un sistema computacional, la cual comprende elementos de software, las propiedades observables de estos componentes y las relaciones entre ellos. En este proceso de diseño se establecen prioridades de calidad tomando en cuenta todos los requisitos, restricciones y la infraestructura disponible, para desarrollar un software de calidad dentro de un plazo de tiempo determinado.

Con la intención de realizar el modelado de todos estos aspectos correspondientes al diseño de la aplicación, el lenguaje *UML (Unified Modeling Language)* está consolidado como el lenguaje estándar para realizar esta tarea, trabaja mediante el uso de diversos diagramas encargados de representar en una etapa previa a la codificación del sistema, sus características, funciones, comportamiento, composición, entre otros. De esta manera, se facilita el proceso de desarrollo de software para el programador, ya que previamente tiene una visión gráfica del trabajo a realizar y la totalidad de las condiciones que debe tomar en cuenta.

Por su parte, un aspecto importante del proceso de diseño de la aplicación es la construcción de la interfaz, en este caso una interfaz web. Para que su diseño sea efectivo debe lograr que los usuarios accedan con facilidad a las distintas funciones desarrolladas, la ubicación de todos los elementos con los que el usuario puede interactuar deben disponerse de forma coherente para facilitar la comprensión y navegación sin complejos razonamientos. Sin importar la estabilidad funcional de la aplicación y la efectividad de las herramientas presentadas al usuario, se deben seguir estos principios al diseñar la interfaz para evitar que el usuario tenga una experiencia no satisfactoria al usar la aplicación.

Por último, se destaca la importancia del proceso de diseño y modelado de la aplicación, usando las herramientas adecuadas para realizar esta tarea, se puede garantizar una correcta

implementación posterior de la aplicación. En este capítulo se describen los procesos del diseño de la arquitectura de la aplicación, de la interfaz de usuario y del diseño del modelo de bases de datos, acompañados de gráficos que permiten visualizar estos elementos de manera sencilla y práctica.

4.1 Arquitectura de la Aplicación

El diseño de la arquitectura propuesta para este sistema está estrechamente relacionado con el patrón que implementan los *frameworks* con los que se ha decidido construir la aplicación. Esta será una arquitectura separada en tres capas como lo indica el patrón de arquitectura Modelo, Vista, Controlador (MVC), que además de estructurar lógicamente la aplicación, detalla las responsabilidades exactas de cada capa y la forma de relacionarse entre sí. En la figura 9 se puede observar la arquitectura propuesta para esta aplicación web utilizando el patrón MVC, las cuales serán explicadas con más detalle a continuación.

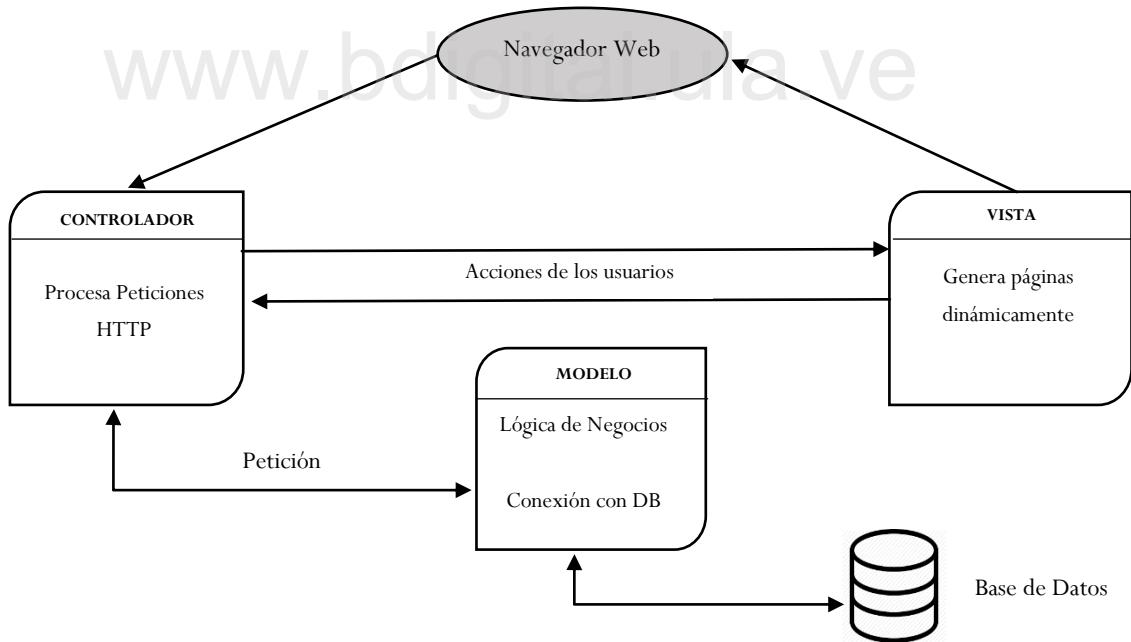


Figura 9: Arquitectura de software del sistema

4.1.1 Vista

Esta capa comprende la presentación de la interfaz de usuario en el navegador web, donde interactúan archivos *HTML*, hojas de estilo *CSS*, y scripts de *TypeScript* quienes controlan la lógica de todo lo que sucede a través de las directivas de *AngularJS*. Desde esta capa se generan las peticiones *HTTP* que viajan al controlador, donde estarán definidas las acciones que deben ejecutarse según la naturaleza de la petición, en consecuencia también interviene la capa modelo, que será explicada más adelante en este capítulo. En la interfaz de usuario se ubicarán todos aquellos elementos con los que el usuario puede interactuar, así como elementos que servirán para mostrar la información que este solicite y los formularios para enviar nueva información a la base de datos.

Para ubicar en la interfaz los elementos web de una manera estandarizada y correcta, se cuenta con el *framework Bootstrap*, que es una biblioteca multiplataforma de código abierto que ofrece un conjunto de herramientas y plantillas para construir páginas web con rapidez y gran compatibilidad, ya que es multiplataforma y es soportado por múltiples navegadores web. Su método de implementación es realmente simple y le ahorra al programador valioso tiempo de programación, básicamente su funcionamiento gira en torno a la división del área de diseño en 12 columnas, donde se deben encajar todos los elementos deseados y automáticamente se encargará de ajustar esta configuración a distintas resoluciones y múltiples dispositivos, convirtiéndose así en una valiosa herramienta para el desarrollo de esta aplicación.

4.1.2 Modelo

Es un conjunto de clases que representan la información del mundo real que el sistema debe procesar, deben plasmarse todas las características que poseen los elementos que se desean modelar, para que de esta manera la información que se almacena en la base de datos sea útil para el usuario, también deben ser definidas las reglas del negocio, que corresponden con la definición de las funcionalidades del sistema y su lógica. Esta capa se encarga de recibir las peticiones *HTTP* que surgen a través de los usuarios que interactúan con la interfaz, y realiza las operaciones correspondientes en la base de datos para cumplir estas solicitudes.

En cuanto a la base de datos se refiere, cada uno de estos elementos que surgen del proceso de análisis se representan con una tabla alojada en el *SMBD* (Sistema manejador de base de datos), las relaciones entre estos elementos, pertenecientes a la lógica de negocios, también deben ser plasmadas

en el diseño de la base de datos, para que la información pueda ser guardada, recolectada y eliminada de manera correcta. De esta forma, se observa que la capa modelo y la estructuración de la base de datos son entidades que están estrechamente relacionadas.

4.1.3 Controlador

Se encarga de dirigir el flujo del control de la aplicación, así como de gestionar la lógica de manejo de las peticiones que se generan desde la capa vista, a través de ellas se almacena y se provee la información que solicita el usuario, pasando por la capa modelo para finalmente interactuar con la base de datos. Su funcionamiento consiste en recibir una solicitud que coincide con una *URL* especificada, se ejecuta el método correspondiente y se realiza la operación correspondiente con la información alojada en la base de datos.

Al mismo tiempo, la capa controlador está encargada de abrir nuevas vistas según sea la navegación del usuario en la interfaz, para este fin existe una tabla que contiene un mapeo de las *URL* que sirven como enlace para obtener una nueva vista, junto con el archivo debe ser ejecutado para mostrar al usuario esta nueva página. Las actividades que se han descrito se llevan a cabo usando funcionalidades del *framework Laravel*, ofreciendo así integridad a la información durante las interacciones con la base de datos y llevando al usuario una mejor experiencia al usar la aplicación.

4.2 Diagrama de Clases

En el diagrama de clases mostrado en la figura 10, se muestran las clases que componen la base de datos diseñada y las relaciones entre ellas. Cada clase funciona como un molde para describir un objeto relevante para el desarrollo de las funciones del sistema, cada una de ellas posee atributos, para describir las características de cada una, y métodos, que define su comportamiento y las acciones que puede llevar a cabo.

Para comenzar, se puede observar que existe un conjunto de clases que sirven la configuración del sistema, estas son: *users*, *hotels* y *payment-type*. La clase *users* está encargada de albergar los usuarios del sistema, incluye datos personales de identificación, el nivel de acceso del usuario representado por el atributo *admin*, y finalmente la contraseña con la que el usuario hará ingreso al sistema. Automáticamente esta contraseña es cifrada al momento de registrar el usuario, función que ofrece *laravel* para aumentar la seguridad en la base de datos cuando se trata de la gestión de los usuarios. El

atributo *remember-token* es un código que asigna el *framework* cuando autentica a un usuario para darle ingreso al sistema, este *token* es validado al momento que el usuario realiza peticiones *HTTP* para evitar la falsificación de peticiones por parte de sujetos externos.

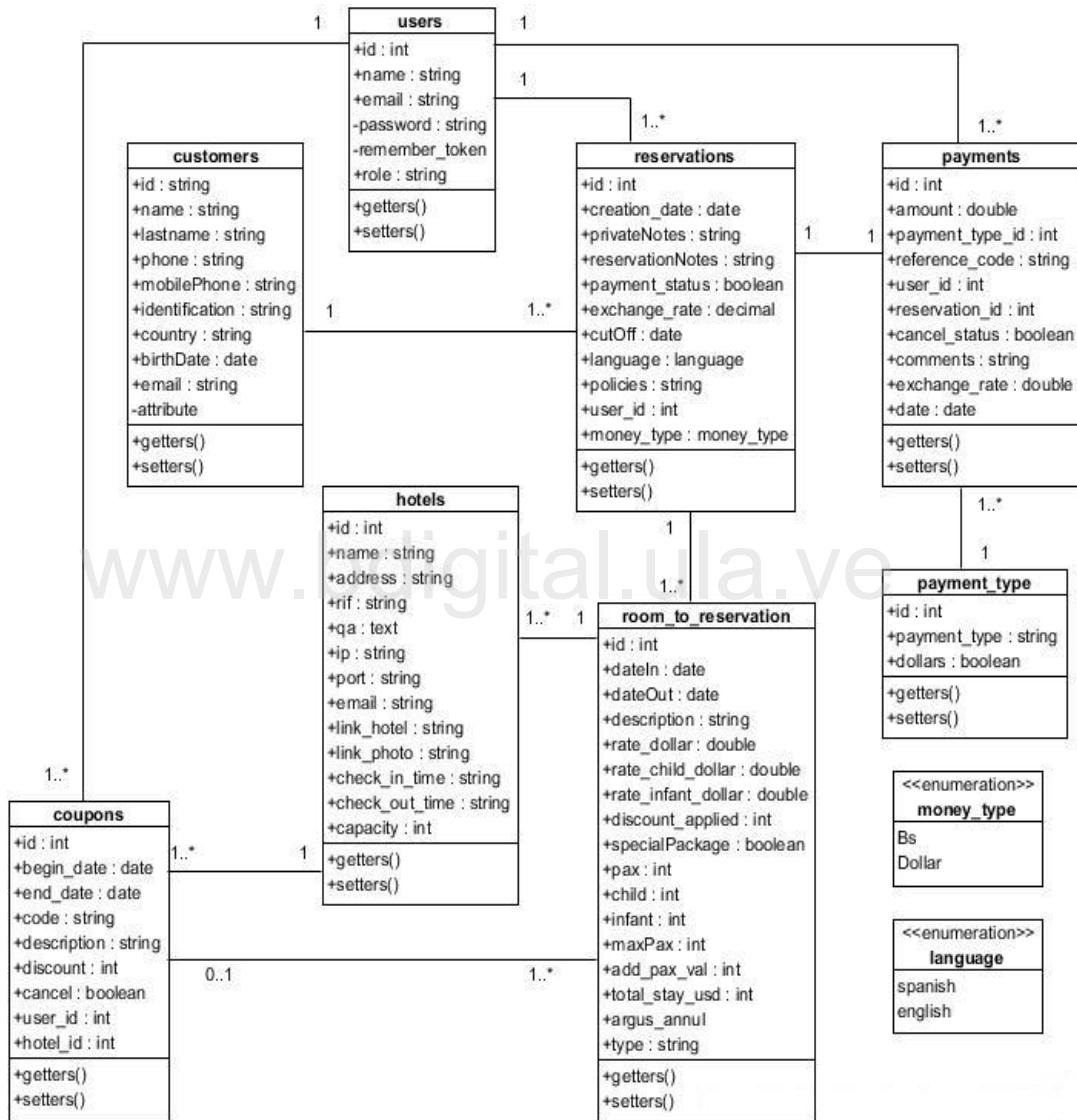


Figura 10: Diagrama de clases.

La clase *hotels* está diseñada para alojar la lista de hoteles con los cuales se podrá interactuar a través del sistema, principalmente está compuesta de datos informativos del hotel como el nombre,

RIF, capacidad, email y hora de entrada y salida. Existe un atributo llamado *QA* por las siglas de *Questions & Answers* que significa preguntas y respuestas, acá el personal del grupo podrá incluir una serie de preguntas frecuentes que pueda realizar un cliente sobre las distintas sedes hoteleras, para apoyar al agente de reservas a dar respuesta oportuna a las inquietudes del cliente al momento de solicitar una reserva. Podemos destacar por último en esta clase, el atributo *IP*, esta corresponde con la dirección *IP* que posea el servidor de datos de cada sede y permitirá la comunicación entre el sistema de gestión hotelera de cada sede con el sistema desarrollado.

La clase *payment_type* es la última clase de configuración del sistema, en ella se especifican los tipos de pagos que admite el sistema para cada moneda. Se plantea la posibilidad de que el cliente cancele en moneda local, o en divisas internacionales, por esto la presencia del atributo *dollar* en la clase. Básicamente estas clases permiten que el sistema comience a funcionar, definiendo el entorno inicial de trabajo sobre el cual se pueden comenzar a realizar todas las operaciones plantadas en la ingeniería de requisitos.

Luego, tenemos las clases que están destinadas a almacenar los datos que se producen a partir del uso de la aplicación por parte de los usuarios, la clase *reservations* es la encargada de recibir la información proveniente de las reservaciones realizadas, sus atributos permiten precisar su fecha de realización, la moneda en la cual se desea emitir la reservación, fecha de vencimiento, operador que la realizó y los comentarios adicionales sobre la reserva. Como se puede observar, existe una relación entre esta clase y la clase *room_to_reservation*, esta será la encargada de contener los tipos de habitación incluidos en la reserva, estas pueden ser en hoteles distintos y en fechas diferentes, de manera que una reserva con estas características no es necesario que se divida en varias reservas individuales. Además en esta clase se especifica el costo de la estadía y la cantidad de personas, niños e infantes para cada habitación reservada.

Otra de las clases relacionadas con la clase *reservations*, es la clase *coupons*, esta permite al personal de marketing y ventas del grupo hotelero diseñar cupones de descuento para el uso de clientes, estos cupones funcionan como una promoción e incluye un descuento sobre el precio regular de la habitación, podemos observar cómo sus atributos permiten definir una fecha inicial de validez y una final, una descripción, un hotel para el cual es válido el cupón, un código mediante el cual se puede tener acceso al descuento y finalmente el porcentaje asignado.

Por último, la clase *payments* tiene la funcionalidad de almacenar los pagos que se realizan a las reservaciones creadas, sus atributos permiten almacenar todos los datos referentes al pago como lo son

el monto, el código de referencia, la fecha de realización, comentarios y un atributo para conocer si el pago es válido o ha sido anulado. A través de su relación con la tabla *payments_type* se permite que el usuario seleccione el método con el que ha sido realizado el pago, esto con la finalidad que no sean recibidos pagos por vías que no están autorizadas por la gerencia y para evitar errores que puedan ocurrir al permitir que el usuario escriba manualmente en este campo.

4.3 Diseño de la Interfaz

La interfaz de usuario presenta en dos diseños, de manera general. En la primera, es solicitado al usuario los datos para que realice su ingreso al sistema, y en la segunda se lleva al usuario un menú a través del cual puede acceder a todas las funciones del sistema. Cada módulo contendrá los elementos web necesarios para mostrar la información al usuario y permitir que interactúe, enmarcado en el diseño que se presenta a continuación.

4.3.1 Interfaz de Usuario No Autenticado

Esta interfaz está destinada a usuarios que acaban de iniciar el sistema y aún no han introducido los datos necesarios para autenticarse. Por seguridad, no es mostrada ninguna de las opciones del sistema, es un requisito indispensable para esto, que el usuario llene el formulario de ingreso colocando su email y su contraseña y de ser exitoso su intento de iniciar sesión, será llevado al menú principal del sistema donde según sus privilegios podrá navegar a través de las distintas funciones desarrolladas. En la figura 11 se observa la interfaz descrita anteriormente.

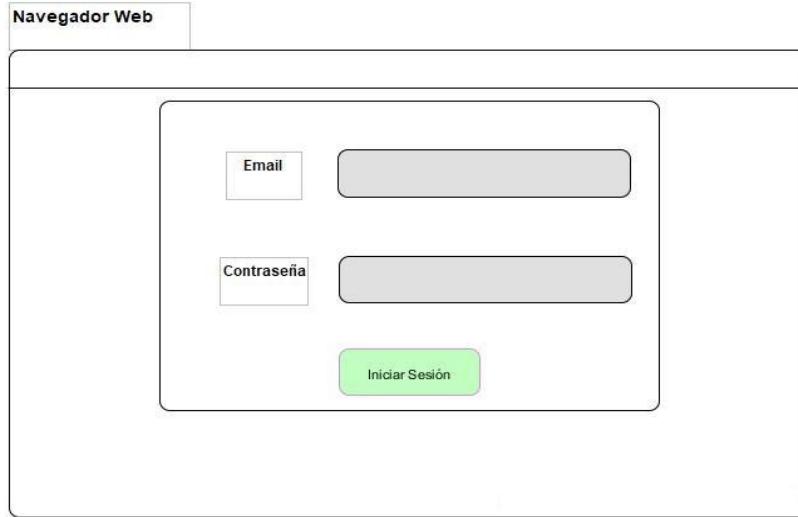


Figura 11: Diseño de Interfaz de Usuario No Autenticado.

4.3.2 Interfaz de Usuario Autenticado

A partir del momento que el usuario ingresa correctamente al sistema, se muestra el diseño que se observa en la figura 12. Este diseño será el empleado para cada uno de los módulos, así como para el menú principal, y se divide básicamente en dos partes, una barra de navegación y un espacio para contener todos los elementos web deseados. Esto se realiza para maximizar el uso de la pantalla en el navegador web, de esta manera pueden contenerse una mayor cantidad de elementos y se lleva al usuario una interfaz más cómoda.

La barra de navegación estará compuesta de elementos básicos informativos como el logo de la empresa y el nombre del usuario, además de elementos de navegación como lo son menús desplegables, a fin de llevar al usuario todas las opciones disponibles de manera resumida y fácil de utilizar. En el área destinada para mostrar el contenido de cada módulo, se ubicarán los elementos necesarios según la información con la que el usuario debe interactuar, se hará la disposición de los mismos para garantizar que su uso sea sencillo, entendible e intuitivo. Cabe destacar que durante la navegación del usuario, el área de la barra de navegación y todos sus componentes permanecerá fija, los cambios entre las páginas se notarán en el área del contenido de la página.

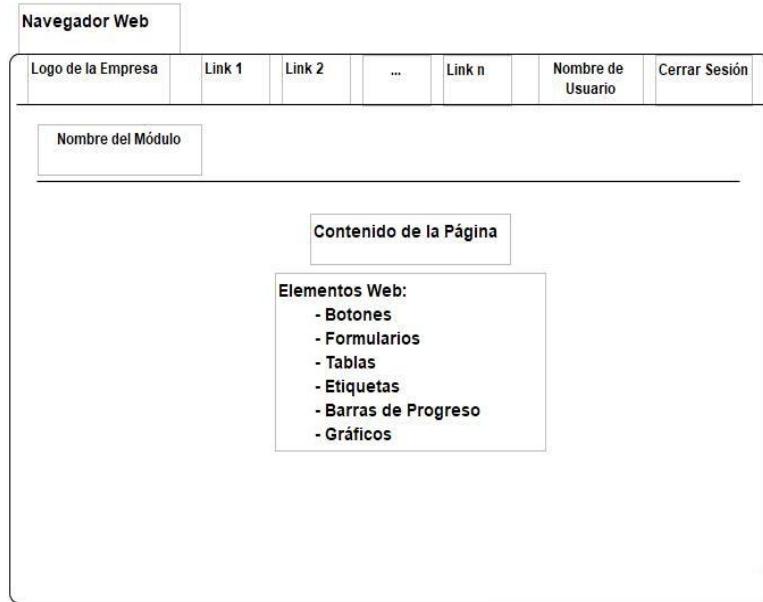


Figura 12: Diseño de Interfaz de Usuario Autenticado

4.3.2.1 Interfaz del Módulo de Consultar Disponibilidad

En la figura 13 se muestra el módulo de consultar disponibilidad, donde se observa que el campo *Hotel* incluye una búsqueda, dicha búsqueda presentará la lista de hoteles registrados al usuario para que este seleccione y posteriormente indique la fecha de entrada y salida en los campos *Fecha desde* y *Fecha Hasta*. Al presionar el botón *Consultar* se hará la petición correspondiente al software de manejo hotelero *Argus* quién entregará la lista de habitaciones disponibles para cada día del intervalo con sus cantidades, permitiendo al agente verificar la posibilidad de realizar la reserva que desea el cliente para la fecha indicada. En caso de no existir comunicación con el hotel que se desea consultar será llevado al usuario un mensaje de *Time Out* o tiempo de espera agotado, para que se realice por parte del personal encargado una verificación en los protocolos de comunicación y sea devuelto a la normalidad el funcionamiento del sistema.

Navegador Web							
Logo de la Empresa	Link 1	Link 2	...	Link n	Nombre de Usuario	Cerrar Sesión	
Módulo de Consulta de Disponibilidad							
Hotel (Busqueda)	Fecha Desde	Fecha Hasta	Consultar				
Habitaciones Disponibles: (tabla)	Tipo	Fecha	...	Descripción	Disponibles	Precio	
	

Figura 13: Diseño de Interfaz: Módulo de Consultar Disponibilidad.

4.3.2.2 Interfaz del Módulo de Realizar Reservación

En la figura 14 se muestra el módulo de realizar reservación del sistema, este puede verse como una versión más extensa del módulo de consultar disponibilidad ya que este proceso es un pre-requisito para realizar la reservación. Una vez realizada la consulta, explicada anteriormente, se permitirá que el usuario agregue los distintos tipos de habitación deseados por el cliente a una *Lista de Compras*, luego de hacerlo, este podrá cambiar el hotel y las fechas deseadas para seguir añadiendo habitaciones, de esta manera se evita que el usuario al presentarse una reserva de múltiples destinos y distintas fechas, deba realizar varias reservas individuales.

Así, una vez recolectada esta información sobre los datos del hospedaje, se procede a solicitar al usuario que introduzca los datos personales y de contacto del cliente. Para esto se dispone de un campo para cada dato que debe ingresar el usuario, al tener una base de datos conjunta de todos los clientes, si este se ha hospedado anteriormente en cualquiera de las sedes, se auto-rellena todo el formulario de forma automática, para facilitar y agilizar el proceso. Finalizado el ingreso de la información, al pulsar el

botón *Reservar* se guardará la reserva, se emitirá un comprobante en formato *PDF* y se hará el ajuste correspondiente en el inventario de habitaciones disponibles.

Figura 14: Diseño de Interfaz: Módulo de Realizar Reservas.

4.3.2.3 Interfaz del Módulo de Realizar Pagos

En la figura 15 se indica el módulo de realizar pagos, como se observa, el primer campo que debe especificarse es el *Id de Reserva*, este es un número de control auto-generado que se asocia a cada reserva emitida, para facilitar su gestión y localización. Luego de especificarlo, el sistema indica de manera informativa cual es el monto de la estadía de la reserva seleccionada, para posteriormente permitir al usuario llenar el formulario con el resto de datos referentes al pago recibido. Una vez concluido el ingreso de la información en los campos correspondientes, se presiona el botón *Realizar Pago*, el sistema entonces guardará la información en la base de datos y generará un comprobante en *PDF* del pago recibido.

Por otro lado, del proceso de ingreso del pago depende el estatus de la reserva, ya que la fecha de vencimiento que se indica al registrarla, es su fecha límite de validez sin que el cliente realice el pago.

En caso de no ser recibido y cumplido el lapso estipulado, la reserva podría ser anulada, sin embargo, el sistema no lo realiza de forma automática ya que es una cuestión de políticas hoteleras la realización de este proceso.

Navegador Web

Logo de la Empresa	Link 1	Link 2	...	Link n	Nombre de Usuario	Cerrar Sesión
--------------------	--------	--------	-----	--------	-------------------	---------------

Módulo de Pagos

Id de Reserva: (Busqueda)		Monto Estadia:	
Datos del Pago:			
Método de Pago		Monto del Pago	
Número de Referencia:		Comentarios	

Registrar Pago

Figura 15: Diseño de Interfaz: Módulo de Realizar Pagos.

4.3.2.4 Interfaz del Módulo de Gestión de Usuarios

En la figura 16 se muestra el módulo de gestión de usuarios, permite que se agreguen nuevos usuarios al sistema, así como eliminarlos. Para comenzar el proceso de incluir un nuevo usuario, se debe llenar cada campo del formulario, el campo email funciona como *login* para ingresar al sistema, por lo que éste debe ser único para cada usuario, posteriormente se pulsa el botón *Grabar*. Por su parte, para modificar, se hace uso de la búsqueda del sistema, esta permitirá seleccionar de una lista qué usuario se desea modificar y al clicar en él, se permitirá al usuario cambiar la información necesaria y guardar los datos pulsando el botón *Grabar*. Por último, para eliminar un usuario se repite el proceso de búsqueda anterior y se presiona el botón eliminar, de forma inmediata el usuario seleccionado es removido de la base de datos.

Navegador Web							
Logo de la Empresa	Link 1	Link 2	...	Link n	Nombre de Usuario	Cerrar Sesión	
Módulo de Gestión de Usuarios							
Usuario: (Busqueda)				Nombre			
Email (Login)				Contraseña			
<input type="checkbox"/> Admin				Registrar/Modificar/Eliminar Usuario			

Figura 16: Diseño de Interfaz: Módulo de Gestión de Usuarios

www.bdigital.ula.ve

4.3.2.5 Interfaz del Módulo de Cambio de Parámetros del Sistema

En la figura 17 se muestra el módulo de Cambio de Parámetros del Sistema, en él se presentan una serie de pestañas a través de las cuales el usuario podrá navegar. Estas pestañas representan los distintos aspectos modificables del entorno de configuración del sistema, el contenido de cada una permitirá que el usuario consulte, modifique, agregue o elimine los datos según sea el caso. Las variables presentadas en este módulo y su gestión son de vital importancia para el correcto desempeño del sistema, estas describen el comportamiento que tomará frente a distintas situaciones, su acceso debe ser restringido y vigilado.

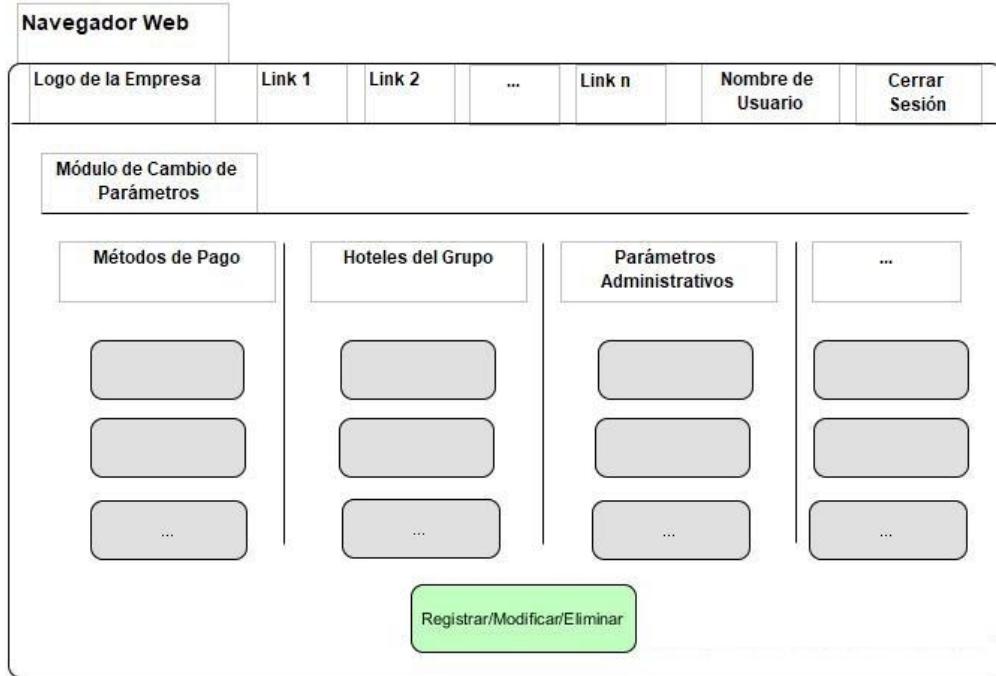


Figura 17: Diseño de Interfaz: Módulo de Cambio de Parámetros

Por último, debe aclararse que para cada uno de estos módulos de registro de información se incluirá un módulo de consultas, básicamente consta de una tabla filtrable para cada entidad en la cual el usuario puede listar la totalidad de operaciones realizadas según los distintos criterios que existen para cada una, permitiendo así la búsqueda y análisis de la información de cada tipo almacenada en la base de datos. En este capítulo han sido definidos todos los aspectos referentes al diseño arquitectónico de la aplicación, se organizó el funcionamiento del sistema dividido en las tres capas antes expuestas, cada una de las cuales con sus funciones bien definidas y el trabajo en conjunto cumplen con todos los requerimientos del proyecto. Se realizó además, el modelo de clases que representa la base de datos relacional que permitirá soportar el almacenamiento de los datos generados al utilizar todas las funcionalidades del sistema y la definición de cada una de las interfaces de los módulos que componen el sistema.

Capítulo 5

Desarrollo de Versiones

La aparición de cada vez más aplicaciones y sitios web ha producido un importante crecimiento en el desarrollo de software sobre dicha plataforma. Ahora bien, desde el punto de vista de la ingeniería de software es importante utilizar los mecanismos adecuados, para que la realización de este tipo de aplicaciones satisfaga las necesidades tanto de los usuarios como de los clientes a los cuales está destinada su utilización. Según afirman los autores, entre ellos Jacobson (2000), el modelo de proceso más adecuado para el desarrollo de este tipo de software es el iterativo e incremental, como el seleccionado en esta investigación. Plantea obtención de diversas versiones del producto final, en la presente investigación se realizaron tres iteraciones que abarcaron todos los aspectos definidos en la fase de desarrollo de requerimientos.

De esta forma, cada una de estas iteraciones permite la retroalimentación de información por parte de los usuarios y permite validar que los requisitos planteados han sido satisfechos a medida que el desarrollo del sistema de información se lleva a cabo, cada iteración contempla un ciclo corto de desarrollo en la cual se obtiene una nueva versión con mejoras sobre la anterior, que debe ser totalmente funcional y completa.

La primera iteración está compuesta en dos partes, la primera de configuración del entorno de trabajo y familiarización con las herramientas a disposición para el desarrollo de la aplicación, este proceso incluye la integración de los *frameworks* utilizados para su trabajo en conjunto, así como la generación de la base de datos. La segunda parte corresponde la codificación de los primeros módulos que darán funcionamiento al sistema, éstos son el módulo de gestión y autenticación de usuarios, el módulo de consultar disponibilidad y el módulo de reservaciones. Además, para permitir la configuración de los hoteles pertenecientes al grupo hotelero, se realizó el *CRUD* de Hoteles, característica perteneciente al módulo de cambio de parámetros del sistema.

La segunda y tercera iteración, comprenden una depuración y progresiva validación de los módulos desarrollados en ciclos anteriores, así como el desarrollo de las nuevas características pertenecientes a la iteración en curso. En estas se realizó la codificación del resto de requerimientos funcionales, así como de todas las características que apoyan estas actividades como lo son módulos de consulta y *CRUD* de distintas entidades de configuración, las cuales serán mostradas más a detalle en este capítulo.

5.1 Primera Iteración

Durante esta fase del desarrollo se realizó en primer lugar una familiarización con las herramientas que serán utilizadas durante todo el desarrollo del sistema. Como fue indicado anteriormente, cada *framework* utilizado tendría funciones bien definidas, durante este proceso se procedió a integrar ambas plataformas en un mismo entorno de trabajo, se decidió utilizar *Visual Studio Code* como editor de código ya que está refinado y optimizado para construir y depurar aplicaciones web, en este editor fue posible albergar la estructura de cada *framework* permitiendo ubicar fácilmente los grupos de archivos según su función, además de generar buenas prácticas de desarrollo de software al tener una organización adecuada del proyecto.

Además, se generó la base de datos a partir del diagrama de clases mostrado en el capítulo anterior, se utilizó una característica de *Laravel* denominada *Migrations*, cuya traducción corresponde con migración. Este tipo de archivo permite albergar en el proyecto una serie archivos que al ejecutarlos, automáticamente genera la base de datos en el manejador de base de datos utilizado, esto es de vital importancia ya que durante el proceso de desarrollo surge la necesidad de borrar la base de datos y generarla de nuevo debido a las pruebas realizadas, y con esta función se realiza este proceso de manera rápida y segura.

La primera versión del producto contaba con las siguientes características desarrolladas:

- Módulo de Autenticación de Usuarios: La figura 18 muestra la primera página que recibe el usuario al iniciar la aplicación, en ella se colocan sus datos de inicio de sesión y el sistema valida que esta información sea correcta para darle entrada al menú principal, el cual puede apreciarse en la figura 19.

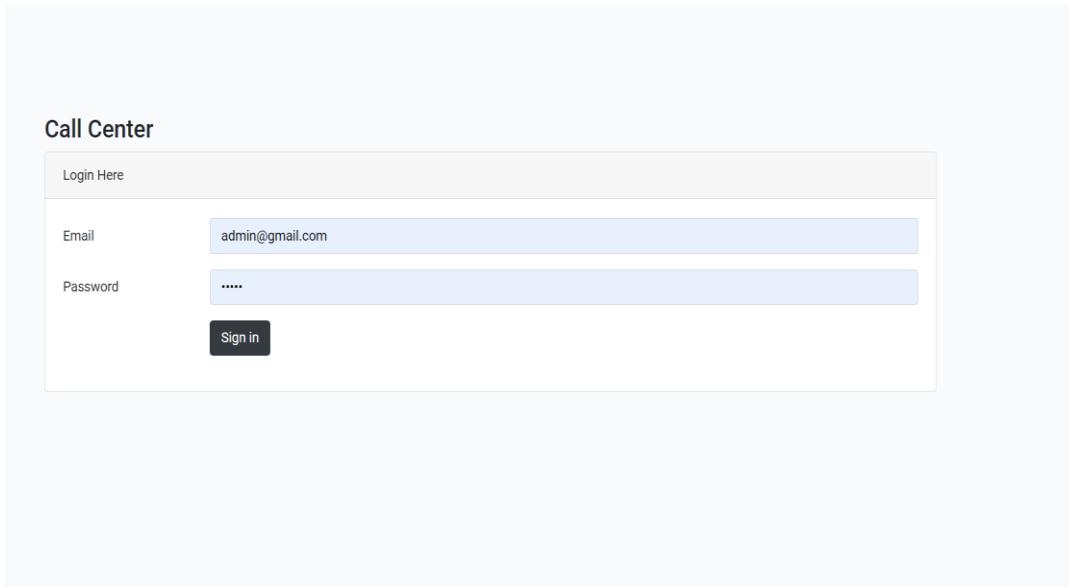


Figura 18: Página de Inicio de Sesión.



Figura 19: Menú Principal del Sistema.

- Módulo de Consulta de Disponibilidad: En la figura 20 se muestra la página de consulta de disponibilidad, en ella el usuario selecciona el hotel deseado, indica un intervalo de fechas y el sistema interpreta un *JSON* proveniente del sistema de gestión *Argus* correspondiente, este

contiene la información sobre el inventario de habitaciones por día, entre otros datos, y la página se encarga de recibir esta información y mostrarla en una tabla para su interpretación por parte del usuario. En esta iteración se realizó la interpretación de un *JSON* generado manualmente, ya que la recepción de estos datos de manera automática no estaba prevista sino hasta una iteración posterior.

Hab	02/10	03/10	04/10	Max-Pax	Pax-Base	Add-Pax-Val	Tarifa Bs	Tarifa USD	Tasa de cambio
DEJE	4	4	4	4	0	0	0	34	0
DOBL	94	94	94	4	0	0	0	31	0
JSUI	6	6	6	4	0	0	0	55	0
KEJE	15	15	15	4	0	0	0	33	0
KING	81	81	81	4	0	0	0	30	0
SLUJ	5	5	5	4	0	0	0	60	0
SPAN	7	7	7	4	0	0	0	60	0
SPRE	1	1	1	5	0	0	0	80	0

Figura 20: Interfaz de Módulo: Consultar Disponibilidad.

- Módulo de Reservación: En la figura 21 se muestra el Módulo de Reservas del sistema, este solicita los datos requeridos del cliente para completar el formulario de reserva, posteriormente, como se observa en la figura 22, el usuario debe repetir el proceso de consulta de disponibilidad explicado anteriormente, en esta tabla el usuario tendrá la posibilidad de añadir las habitaciones que desea el cliente a la lista de compras, indicada en el sistema como “Carrito de Compras”. Al realizar estos dos procesos, se presiona el botón reservar y luego de la validación de los datos, se almacena esta información en la base de datos.

Número de Identificación
V23305337 Historial de Reservas

24 horas 3 días 7 días
 Sin pago

Pago Bolivares Pago Dolares

Español

Nombre * Apellido * Dirección * Cut-Off / Fecha de Validez

Diego Perrone MERIDA

Código Postal * País * Teléfono Móvil * Teléfono Email *

5101 VENEZUELA 0424700000 02742000000 diego@gmail.com

Reservation Notes Private Admin Notes

Figura 21: Interfaz de Módulo: Realizar Reservas.

Buscar hotel HOTEL VENEZUELA

Desde : Hasta :

10/3/2019 10/4/2019

Consultar

Cantidad de habitaciones disponibles por tipo para cada día.				
Hab	03/10	Agregar al carrito	Max-Pax	Pax-Base
DEJE	3		4	0
DOBL	94		4	0
JSUI	6		4	0
KEJE	15		4	0
KING	81		4	0

Figura 22: Botón “Añadir al Carrito” para añadir habitaciones a reserva.

- Módulo de Gestión de Usuarios: Se programó la página con sus funcionalidades completas en la cual se ingresan, modifican, eliminan y consultan los usuarios del sistema. Forma parte del Módulo de Configuración del sistema como se muestra en la figura 23, además, el formulario de creación de usuarios se puede observar en la figura 24.

Figura 23: Interfaz de Módulo: Gestión de Usuarios.

Figura 24: Formulario para registrar nuevos usuarios.

5.2 Segunda Iteración

En esta fase del desarrollo incremental, se trabajó a partir de las funcionalidades programadas en la primera iteración, añadiendo nuevos módulos y funcionalidades, que serán descritas a continuación:

- Módulo Consulta de Reservas: En la figura 25 se muestra la página desarrollada con el fin de consultar las reservas en la base de datos según una serie de parámetros o filtros. Estos filtros que han sido añadidos la página permiten que el usuario realice una consulta tan específica o generalizada como necesite, haciendo posible que combine criterios como: hotel al que pertenece la reserva, fecha de emisión, status de la reserva o id de reserva. Para mostrar la información se utilizó una tabla filtrable cuyas columnas pueden ser ordenadas de manera alfabética para brindar al usuario una mayor cantidad de herramientas en su búsqueda, así como se permite imprimir esta tabla a manera de reporte para labores administrativas. El botón “Ver” en cada fila permite ver la información detallada de cada reserva, como se observa en la figura 26.

ID Reserva	Nombre	Apellido	Emision	Identificación	Hab / Paquetes	Acciones	Argus status	Status	Pago
1	DIEGO	PEREZ	Oct 4, 2019	V23305337	...	<button>Ver</button>	No enviado	Activa	Sin pagos
2	EMILIO	GONZALEZ	Oct 4, 2019	V27777291	...	<button>Ver</button>	No enviado	Activa	Sin pagos

Figura 25: Interfaz de Módulo: Consulta de Reservas.



Figura 26: Consulta de datos de reserva.

- Módulo de Registro de Pagos: En la figura 27 se muestra el módulo para registrar pagos del sistema, este solicita en primer lugar la reserva a la cual se desea añadir el pago, se despliega un diálogo modal del cual se selecciona la reserva deseada y el sistema de forma automática muestra en la pantalla los datos personales del cliente, así como los datos de la estadía. Se puede observar las habitaciones reservadas en cada hotel, así como las fechas de ingreso y salida en una tabla, así como los montos del total general de la reserva y el monto mínimo para realizar la reserva. En la figura 28 se observa el formulario de pago, luego de llenarlo el usuario pulsa el botón “Procesar Pago” para registrarlo en la base de datos. El monto mínimo para reservar es calculado a través de un porcentaje especificado en el módulo de configuración, esta función también fue desarrollada para permitir que se defina este parámetro, así como el módulo para crear y eliminar formas de pago, el cual está especificado en la figura 29.

Formulario de Pago

Id de la reservación:

Nombre del cliente:

Habitaciones/Full Day

Tipo Hab / Actividad	Hotel	Fecha In	Fecha Out	Estadia	Cambio	Pax	Niños	Infante
DOBL	HOTEL VENEZUELA	2019-10-04	2019-10-06	62	0	2	0	0
KING	HOTEL VENEZUELA	2019-10-04	2019-10-06	60	0	1	0	0

Pasajes Aéreos

Línea	Ruta	Fecha	Nombre	Apellido	Cédula	Tarifa USD	Cambio

Total General: **USD \$ 122,00** Bs 0,00
 Mínimo Para Reservar: **USD \$ 30,50** Bs 0,00
 Cancelar

Figura 27: Interfaz de Módulo: Realizar Pagos.

Datos del Pago:

Cantidad a pagar: **30.5 Dollars**

Tipo de pago *: Transferencia Internaci...

Fecha de pago: **10/4/2019**

Código de referencia: **5102948102**

Observación: **Depósito directo. Cliente envió copia de comprobante**

Figura 28: Formulario de Pago.

Configuración

Tipos De Pago:

Crear una nueva forma de pago: **Depósito** **Dólares**

Figura 29: Interfaz de Módulo: Inclusión de formas de pago.

- Módulo de Consulta de Pagos: En la figura 30 se observa el módulo de consulta de pagos, presionando el botón consultar se realiza una búsqueda de todos los pagos registrados en el sistema, con el uso del filtro y del ordenamiento que posee cada columna el usuario posee las herramientas necesarias para ubicar el pago que deseé. Si el usuario utiliza el intervalo de fechas, entonces los pagos reflejados en la tabla corresponderán a los realizados entre las fechas indicadas, como se observa en la figura 31, adicionalmente se programaron en la página totalizaciones por forma de pago y moneda para facilitar del operador y para facilitar los procesos de auditoría.



Figura 30: Interfaz de Módulo: Consulta de Pagos.

Cantidad de pagos: 3									
Filter									
Recibo	Monto	Fecha	Tipo de pago	Código	Usuario	ID Reserva	Huesped	Argus Res Status	Re-imprimir
1	122,00 USD \$	Oct 6, 2019	Transferencia Internacional	5623424	admin	1	DIEGO PEREZ	No enviado	<button>Imprimir</button>
2	52,25 USD \$	Oct 4, 2019	Transferencia Internacional	453121223	admin	3	MARIO LOPEZ	No enviado	<button>Imprimir</button>
3	540,00 USD \$	Oct 2, 2019	Transferencia Internacional	41231231	admin	2	EMILIO GONZALEZ	No enviado	<button>Imprimir</button>

Tipo	Cantidad	Total Tipo	Total USD	Total Bs
			714,25 USD \$	0,00 Bs
Transferencia Internacional	3	714,25 USD \$		

Figura 31: Tabla de Consulta de pagos y totalizaciones.

5.3 Tercera Iteración

Durante la tercera iteración se concluyeron todos los casos de uso definidos en el capítulo de requisitos del proyecto, completando las funcionalidades de cada módulo que ya contaban con cierta operatividad. En primer lugar, se realizó en enlace entre el sistema de gestión *Argus* y el sistema desarrollado, ya que para realizar la codificación de los módulos, en especial la consulta de disponibilidad, se utilizaron objetos *JSON* generados de forma manual. Esto se realizó implementando un método de comunicación entre cliente y servidor llamado *Socket*, se define como una interfaz entrada-salida que permite la comunicación entre procesos para el intercambio de cualquier flujo de datos de manera fiable y ordenada, fue implementado a través de librerías pertenecientes al lenguaje *PHP*.

Así, para lograr la comunicación entre ambos sistemas, se debe especificar la dirección *IP* y puerto destino del servidor de datos de cada hotel en su configuración dentro del sistema, en este servidor se encuentra un programa capacitado para recibir peticiones y responder satisfactoriamente a ellas, con información bajo un formato específico que permite que sea interpretada y entregada al usuario. Se muestra parte de la información contenida en un *JSON* de respuesta en la figura 32, a partir de la cual es posible construir la tabla que indica la disponibilidad de habitaciones de un hotel específico en el intervalo solicitado.

```

"rooms": [
  {
    "type": "DEJE",
    "itip": 1,
    "dtrf": "Rack",
    "description": "DOBLE EJECUTIVA",
    "paxBase": 0,
    "maxPax": 4,
    "rate_dollar": 34,
    "rate_exchange": 0,
    "rate_child_dollar": 13,
    "rate_infant_dollar": 0,
    "can_coupon": 1,
    "add_pax_val": 0,
    "values": [
      {
        "amount": 4
      },
      {
        "amount": 4
      },
      {
        "amount": 4
      },
      {
        "amount": 4
      }
    ]
  },
  {
    "type": "DOBL",
    "itip": 2,
    "dtrf": "Rack",
    "description": "DOBLE ESTANDAR",
    "paxBase": 0,
    "maxPax": 4,
    "rate_dollar": 28,
    "rate_exchange": 0,
    "rate_child_dollar": 13,
    "rate_infant_dollar": 0,
    "can_coupon": 1,
    "add_pax_val": 0,
    "values": [
      {
        "amount": 4
      },
      {
        "amount": 4
      },
      {
        "amount": 4
      },
      {
        "amount": 4
      }
    ]
  }
]

```

Figura 32: Parte de *JSON* de respuesta proveniente de sistema de gestión *Argus*.

Además, se realizó la programación de los comprobantes en PDF que genera cada módulo, esto para su almacenamiento, control y en algunos casos para enviarlos al cliente a manera de garantía. Se detallan a continuación los comprobantes y reportes desarrollados en esta iteración:

- Comprobante de Reserva: En la figura 33 se muestra el comprobante de reserva generado por el sistema, este cuenta con todos los datos personales y relacionados con la estadía del cliente, así como informa el *ID de Reserva*, a través del cual puede realizar cualquier reclamo o solicitar información en general. Este comprobante puede ser enviado vía email, ser impreso o sólo guardarse digitalmente para uso interno del departamento de reservas.



Figura 33: Comprobante de Reserva.

- Comprobante de Pago: En la figura 34 se muestra el comprobante de pago, o recibo de pago, este contiene información referente al cliente que ha realizado el pago, reservación asociada, forma de pago y observaciones. De igual forma, puede enviarse al cliente como garantía de su pago o almacenarse para control interno.

 Recibo de pago	Recibo: 1
	Operador: admin
Huesped: DIEGO PEREZ	
Reservación id: 1	
Tipo de pago: Transferencia Internacional	
Fecha de pago: 2019-10-05	
Código de referencia: 5623424	
Observaciones: Pago directo a cuenta	
Total a pagar: USD 244	Total cancelado: USD 122
Restante: USD 122	

Figura 34: Comprobante de Pago.

- Reporte de Reservas: En la figura 35 se muestra el reporte de reservas emitido desde el módulo de consultas, esta función genera un archivo *PDF* con la lista de reservas que se encuentre en la tabla (Ver Figura 25), luego de haber cumplido los criterios de búsqueda, es posible consultar un hotel en específico, un intervalo de fechas o combinar los criterios para obtener un archivo con la información deseada.



Reporte de Reservas

Criterio de búsqueda: Todas. Cantidad Hab:4/4

Id	Nombre	Apellido	Identificación	Fecha de Reserva
1	ANDRES	GARCIA	V023305240	2019-10-01
2	EMILIO	PEREZ	V25920492	2019-10-08
3	ANDREA	LACRUZ	V015600230	2019-10-10
4	MARIO	RODRIGUEZ	V10392024	2019-10-13

Figura 35: Reporte de Reservas.

De esta manera, se cumplen todos los requerimientos funcionales definidos para el proyecto, aplicando un desarrollo incremental que consistió en tres iteraciones, teniendo como punto de partida la definición de las bases de la aplicación y la configuración del entorno de trabajo, llevando a cabo posteriormente el desarrollo progresivo de los módulos y funcionalidades requeridas para satisfacer los objetivos planteados por los casos de uso definidos en la etapa de análisis del proyecto.

Capítulo 6

Pruebas

Una prueba de software se puede definir como una actividad en la cual un sistema, o alguno de sus módulos se ejecutan bajo condiciones previamente especificadas, que cumplan con los requisitos del cliente. Seguidamente, se realiza un proceso de evaluación en el que los resultados obtenidos se comparan con los resultados esperados, para detectar fallos o conductas inesperadas en el software. Estas fallas conducen a un proceso de depuración en el que es necesario localizar la falta asociada a cada fallo y corregir, este proceso origina una nueva prueba para verificar que se ha rectificado el funcionamiento de la aplicación. Como resultado final se puede obtener un cierto grado de confiabilidad y confianza sobre el software que ha sido probado.

De acuerdo con el modelo presentado por Boehm (1984), existen 4 etapas que deben cumplirse para llevar a cabo un proceso de pruebas de manera integral que abarque todos los aspectos del software desarrollado. La primera de ellas, es la prueba de componentes, en ella el programador verifica que cada módulo funciona de acuerdo a lo esperado de forma unitaria. Luego, la etapa de pruebas de integración comprende la ejecución de una serie de pruebas a los componentes de forma conjunta, donde es validada la interacción entre ellos a través de varias interfaces, tomando en cuenta que la salida de un componente es generalmente la entrada de otro.

Por último, se encuentran las etapas de pruebas de sistema y de aceptación, estas son pruebas integrales las cuales incluyen la interacción del cliente para comprobar que todos los requerimientos planteados han sido satisfechos, su principal objetivo es corroborar que el desarrollo del software se realizó abarcando todas las especificaciones descritas en la etapa de análisis y diseño. De esta manera, se aplicó un plan de pruebas en el cual se encuentran consideradas de manera implícita las etapas antes mencionadas, a continuación, se describen las pruebas realizadas a los módulos del sistema desarrollado.

6.1 Objetivo de las pruebas

El objetivo del plan de pruebas es verificar que el funcionamiento de cada módulo es el esperado, así como comprobar que el flujo de eventos del sistema se desarrolla correctamente

6.2 Técnicas a implementar

Se realizaron pruebas manuales donde se siguió el flujo de eventos necesarios en la interfaz de usuario para realizar las acciones descritas en los casos de uso. Se tuvo en cuenta para las pruebas los módulos cubiertos en cada iteración, para detectar fallos en la nueva codificación, así como se mantuvo una constante revisión de los módulos ya desarrollados luego de la primera iteración para verificar que no se perjudicara las funcionalidades ya probadas. Además, se realizaron consultas directamente en la base de datos para constatar el funcionamiento del sistema y se llevaron a cabo cálculos matemáticos manuales para comprobar los valores mostrados al usuario en la interfaz.

6.3 Criterios de las pruebas

Para cada una de las pruebas realizadas, se determinó que en caso de que la funcionalidad evaluada culmine exitosamente, demostrando un comportamiento esperado, se tomará la prueba como exitosa. En caso contrario, se evaluará el código involucrado en la prueba no exitosa para encontrar el origen del fallo, se corregirá y se volverá a aplicar la prueba nuevamente. Las pruebas se darán como concluidas al garantizarse y comprobarse que cada funcionalidad se comporta de la manera esperada.

6.4 Pruebas Realizadas

6.4.1 Consultar Disponibilidad en un hotel

6.4.1.1 Flujo de la prueba

Se abre en la barra de navegación el menú desplegable “Consultas” y se selecciona la opción “Disponibilidad”. Se ubica con el buscador el hotel deseado y se indica el intervalo de fechas “Desde” y “Hasta”, posteriormente se presiona el botón consultar.

Se ha enfocado esta prueba en verificar que el inventario de habitaciones mostrado al usuario es efectivamente el que proviene del software de gestión *Argus*, ya que este es el propósito de este módulo. Para esto se ha fijado especial atención en un tipo particular de habitación, *JSUI* siglas para “Junior Suite”, en el intervalo de tres noches empezando el primero de octubre al cuatro de octubre del año 2019. Se puede observar en la figura 36 que en las tres columnas correspondientes a cada noche de estadía se muestra el número 6, lo que se interpreta como el número de habitaciones disponibles de ese tipo para cada noche, se puede comprobar esta información observando la figura 37 donde se muestra la disponibilidad en el software de gestión *Argus* de ese hotel para ese mismo tipo de habitación y para el mismo intervalo de fechas.

6.4.1.2 Postcondiciones

Se obtiene la disponibilidad solicitada mostrada al usuario a través de una tabla, que luego de la comparación realizada, permite catalogar la prueba como exitosa.



Cantidad de habitaciones disponibles por tipo para cada dia.						
Hab	01/10	02/10	03/10	Max-Pax	Pax-Base	Add-Pax-Val
DEJE	4	4	4	4	0	0
DOBL	94	94	94	4	0	0
JSUI	6	6	6	4	0	0
KEJE	15	15	15	4	0	0
JUNIOR SUITE	81	81	81	4	0	0

Figura 36: Prueba de Verificación de Disponibilidad.



Figura 37: Disponibilidad verificada en Sistema de Gestión *Argus*.

6.4.2 Realizar una reserva

6.4.2.1 Flujo de la prueba

Se selecciona la barra de navegación la opción “Reservas”, se abre el menú desplegable y se escoge la opción “Reservar”. En primer lugar, se deberá llenar el formulario de datos del cliente mostrado en la figura 38 con el formato requerido, este contiene los siguientes campos: identificación, nombre, apellido, dirección, código postal, país, teléfono móvil, teléfono fijo, email y notas (comentarios). Adicionalmente se debe seleccionar el tiempo de validez de la reserva, así como la moneda deseada para expresar los montos, y el idioma (para la impresión del comprobante).

Luego, se debe aplicar el procedimiento para consultar disponibilidad explicado anteriormente, para seleccionar en la tabla usando el botón “Añadir al carrito” las habitaciones deseadas por el cliente para su reserva, como se detalla en la figura 38. Además, debe indicarse la cantidad de personas, niños e infantes y para culminar se presiona el botón “Procesar Reserva”. El módulo deberá generar correctamente el comprobante en formato PDF, guardar la reserva realizada en la base de datos y entregar esta información al sistema de gestión *Argus*, para actualizar el inventario de habitaciones con la inclusión de esta nueva reserva.

Las verificaciones pertinentes para encontrar un comportamiento esperado del módulo son las siguientes:

- Comprobar que los datos en el comprobante de reserva están bien ubicados y corresponden con los datos introducidos en la página.

- Verificar en el comprobante que el tipo y cantidad de habitaciones seleccionadas, cantidad de personas, niños e infantes, fecha de ingreso y salida es igual al ingresado en la página.
- Comprobar costo de la estadía, para cada habitación se debe hacer el siguiente cálculo, *Costo de la noche x Cantidad de noches*, posteriormente se realiza una sumatoria de todas las estadías por cada tipo de habitación y se obtiene el valor total de la estadía del cliente.
- Comprobar en el módulo de consulta de reservas que la información introducida no ha sufrido ningún cambio, se debe mantener la integridad de los datos. Además, con esta prueba se verifica la integración de los módulos (reservar-consultar reserva).
- Validar que la comunicación entre el sistema de reservas y el sistema de gestión *Argus* ha sido exitosa, debe existir una entrada en el archivo de reservas del sistema *Argus* que coincida con la reserva en cuestión.

Número de Identificación
V1500000 Historial de Reservas
Cotización Pre-reserva 24 horas 3 días 7 días Sin pago Pago Bolívares Pago Dólares Espanol Inglés

Nombre * Apellido * Dirección * RAUL MARTINEZ CARACAS
Código Postal * País * 1000 VENEZUELA Teléfono Móvil * Teléfono 04140000000 04247000000 Email * rauljula@ve Private Admin Notes NA NA

Reservation Notes
NA NA

Consultar Disponibilidad Buscar hotel Puerto la cruz Desde: 10/15/2019 Hasta: 10/19/2019 consultar

Puerto la cruz E-21586351-1 Info

Desde: 10/15/2019 Hasta: 10/19/2019 consultar

Cantidad de habitaciones disponibles por tipo para cada día.

Hab	15/10	16/10	17/10	18/10	Agregar al carrito	Max-Pax	Pax-Base	Add-Pax-Val \$	Tarifa Bs	Tarifa USD	Tasa de cambio
DEJE	2	2	2	2		4	0	0	0	34	0
DOBL	94	94	94	94		4	0	0	0	31	0
JSUJ	6	6	6	6		4	0	0	0	55	0
KEJE	15	15	15	15		4	0	0	0	33	0
KING	81	81	81	81		4	0	0	0	30	0
SLUJ	5	5	5	5		4	0	0	0	60	0
SPAN	7	7	7	7		4	0	0	0	60	0
SPRE	1	1	1	1		5	0	0	0	80	0

Carrito de reservas: Refresca para continuar Tarifa total USD= 34 x 4 = 136

Hotel	Status	In	Out	Tipo	Tarifa total Bs	Tarifa total USD	Tasa de cambio	Nueva tarifa Bs	Nueva tarifa USD	Adultos	Niños	Infante	Agregar cupón	Porcentaje del cupón
Puerto la cruz	2019-10-15	2019-10-19	DEJE	0	136	136	0	-	-	1	0	0	Cupón	-

Cantidad de Noches: 4 Cantidad De Habitaciones: 1 Estadia total en bolívares: 0,00 Estadia total en dólares: 136,00

Procesar reserva

Figura 38: Flujo de la prueba realizar reserva.

6.4.2.2 Postcondiciones

Como se muestra en la figura 39, el comprobante de reserva contiene correctamente todos los datos indicados en el formulario, en la figura 40 se observa el módulo de consulta de reserva con la nueva entrada generada y por último en la figura 41 se muestra que la comunicación con el software de gestión *Argus* ha sido exitosa, ya que este posee una entrada con la reserva realizada. El flujo de la prueba ha conducido a un comportamiento esperado de los módulos en evaluación, por lo tanto se han cumplido todos los aspectos necesarios para catalogar esta prueba como exitosa.

	PARADISE	Comprobante de Reserva	Id Reserva:	5				
HOTELS								
Nombres: RAUL	Apellidos: MARTINEZ							
País: VENEZUELA	Teléfono Móvil: 04140000000		Teléfono:	04247000000				
Dirección: CARACAS								
Email: raul@ula.ve								
Recibo #: N/A	Fecha Emisión: Sin Pago	Monto Cancelado: Sin Pago						
Notas de Reserva: NA								
Habitaciones Reservadas								
Hotel	Tipo	Fecha In	Fecha Out	Costo USD \$	% de Desc	Pax	Niños	Infantes
Puerto la cruz	DEJE	2019-10-15	2019-10-19	136	0	2	0	0
Total a Pagar: USD \$ 136,00					Cupón:	Ninguno		

Figura 39: Prueba de emisión de comprobante de reserva.

Buscar Reservas

Id Reserva: 5

Cantidad de reservas: 5			Nombres: RAUL	Apellidos: MARTINEZ	Pais: VENEZUELA							
Filter			Direccion: CARACAS									
ID Reserva	Nombre	Apellido	Tlf Movil: 04140000000	Telefono: 04247000000	Email: raul@ula.ve							
1	ANDRES	GAR	Recibo #: N/A	Monto Cancelado: Sin Pago	Fecha de Emision: Sin Pago							
2	EMILIO	PER	Notas Reservas: NA									
3	ANDREA	LAC	Cupon Descuento:									
4	MARIO	ROD	Habitaciones Reservadas:									
5	RAUL	MA	Hotel	Tipo	Costo USD \$	Cambio	Fecha In	Fecha Out	% Descuento	Pax	Niños	Infantes
			Puerto la cruz	DEJE	136	0	2019-10-15	2019-10-19	0	2	0	0
			Total a pagar USD \$ 136,00					Total a pagar BS 0,00				
Politicas de Reserva												
Items per page: 50 ▾												

Figura 40: Prueba de consulta de reserva.

<i>Consulta Reservacion Localizador: 4891</i>				
<input type="checkbox"/> Reservas		<i>Buscar</i>		
Localiz.	Call C. Id.	Nombres	Apellidos	In
4890	4	MARIO	RODRIGUEZ	15/10/2019 19/10/2019
4891	5	RAUL	MARTINEZ	15/10/2019 19/10/2019

Figura 41: Prueba de verificación de inserción de reservas en sistema de gestión *Argus*.

6.4.3 Realizar un pago

6.4.3.1 Flujo de la prueba

Se selecciona en la barra de navegación la opción “Pagos” para ir a esta página. Luego, una vez en ella se procede a buscar y seleccionar la reserva a la cual se desea ingresar un pago y el sistema cargará todos los datos de la misma en la pantalla. Debe seleccionarse si el pago es por el monto total de la reserva o el pago corresponde al monto mínimo para reservar (Indicado en el menú configuración), además de indicar la moneda con la cual será registrado el pago a través de los botones “Pagar Bs” o “Pagar \$”, se realiza la conversión a través de una tasa de cambio indicada en la información recibida desde el software de gestión *Argus*. Finalmente, se llena el formulario correspondiente: tipo de pago (configurable en el menú configuración), fecha de pago, código de referencia y comentarios.

En la figura 43 se muestra el flujo de esta prueba, se debe comprobar que el porcentaje mínimo para reservar corresponde al porcentaje indicado en la configuración del sistema (25% como se observa en la Figura 42) aplicado al monto total de la estadía, además debe verificarse que el comprobante incluye todos los datos proporcionados en la página y que están ubicados correctamente. También, en el módulo de consulta de reserva, debe indicarse el monto del pago que ha sido realizado así como el número de recibo correspondiente y por último, se debe verificar la integración con el módulo de consulta de pagos.

Configuración

[Lista de Hoteles](#) [Usuarios del Sistema](#) [Políticas de Reserva](#)

% Minimo de cobro:	Moneda:
25	Tipo de moneda *
Bs	

Guardar

Figura 42: Porcentaje mínimo de cobro según módulo de configuración

Formulario de Pago

Id de la reservacion: 5 5 RAUL MARTINEZ

Reserva realizada en la prueba anterior

Habitaciones/Full Day	Hotel	Fecha In	Fecha Out	Estadia	Cambio
DEJE	Puerto la cruz	2019-10-15	2019-10-19	\$ 136	0

Mínimo para reservar = \$136 x 25% = \$34

Total General:	USD \$ 136,00	Bs 0,00
Minimo Para Reservar:	USD \$ 34,00	Bs 0,00

Cancelar **Total** **Pagar \$** **Pagar Bs**

Datos del Pago:

Cantidad a pagar	34	Dollars	Tipo de pago *	Transf. Zelle	Fecha de pago	10/1/2019
Codigo de referencia *		Observacion				
48129318231		Transferencia via Panamá				

Procesar pago

Figura 43: Flujo de prueba de realizar pago.

6.4.3.2 Postcondiciones

Se puede observar en la figura 44 el recibo de pago con los datos del formulario ubicados correctamente, así como se comprobó manualmente los valores mostrados al usuario, en la figura 45 se muestra el módulo de pagos con la nueva entrada generada y por último, la consulta de reserva muestra correctamente los datos del pago introducido, como se aprecia en la figura 46. Por lo tanto, se considera que se han cumplido todas las condiciones para catalogar la prueba como exitosa.

Recibo de pago	
Recibo: 1	
Operador: admin	
Huesped:	RAUL MARTINEZ
Reservación id:	5
Tipo de pago:	Transf. Zelle
Fecha de pago:	2019-10-01
Código de referencia:	48129318231
Observaciones:	Transferencia via Panamá
Restante = \$136 - 34 = \$102	
Estadia Total	Total cancelado: USD 34
Total a pagar: USD 136	Restante: USD 102

Figura 44: Prueba de emisión de recibo de pago.

Desde	...	Hasta	...																								
<input type="button" value="Consultar"/>			<input type="button" value="Imprimir"/>																								
Cantidad de pagos: 3																											
<input type="text" value="Filter"/>																											
<table border="1"> <thead> <tr> <th>Recibo</th><th>Monto</th><th>Fecha</th><th>Tipo de pago</th><th>Código</th><th>Usuario</th><th>ID Reserva</th><th>Huesped</th></tr> </thead> <tbody> <tr> <td>1</td><td>34,00 USD \$</td><td>Oct 1, 2019</td><td>Transf. Zelle</td><td>48129318231</td><td>admin</td><td>5</td><td>RAUL MARTINEZ</td></tr> <tr> <td>2</td><td>33,00 USD \$</td><td>Oct 3, 2019</td><td>Transf. Zelle</td><td>41231235512</td><td>admin</td><td>3</td><td>ANDREA LACRUZ</td></tr> </tbody> </table>				Recibo	Monto	Fecha	Tipo de pago	Código	Usuario	ID Reserva	Huesped	1	34,00 USD \$	Oct 1, 2019	Transf. Zelle	48129318231	admin	5	RAUL MARTINEZ	2	33,00 USD \$	Oct 3, 2019	Transf. Zelle	41231235512	admin	3	ANDREA LACRUZ
Recibo	Monto	Fecha	Tipo de pago	Código	Usuario	ID Reserva	Huesped																				
1	34,00 USD \$	Oct 1, 2019	Transf. Zelle	48129318231	admin	5	RAUL MARTINEZ																				
2	33,00 USD \$	Oct 3, 2019	Transf. Zelle	41231235512	admin	3	ANDREA LACRUZ																				

Figura 45: Prueba de consulta de pagos.



Figura 46: Prueba de actualización de datos de la reserva al realizar pago.

6.4.4 Gestión de Usuarios

6.4.4.1 Flujo de la prueba

Se abre el menú configuración ubicándolo en la barra de navegación, luego se selecciona la opción “Usuarios del Sistema”, esta pantalla muestra los usuarios registrados en el sistema y permite su edición, eliminación y registro. El flujo de esta prueba consistirá en tres fases: se añadirá un nuevo usuario, se modificarán los datos del mismo luego de su registro y se eliminará de la base de datos.

Para añadir un nuevo usuario debemos presionar el botón “Registrar Usuario”, como se muestra en la figura 47, esta página contiene el formulario necesario para crear un nuevo usuario mostrado en la figura 48: Email, Nombre, Apellido, y contraseña, se debe indicar si el usuario que se está agregando es un administrador y por último se presiona el botón “Registrar Usuario”. Para modificarlo, en la segunda fase del flujo de prueba, se presiona el botón ubicado en la columna “editar” y se procede a llenar los datos del formulario de nuevo, podemos observar esta función en la figura 49. Finalmente, para eliminar un usuario se presiona el botón en la columna “Eliminar” y el sistema deberá eliminar de la base de datos el usuario en cuestión.

Configuración					
	Nombre	Email	Tipo	Eliminar	Editar
1	admin	admin@gmail.com	admin	Usuario principal	Usuario principal
Registrar Usuario 					

Figura 47: Prueba de Registro de Usuario.

Registrar usuario

Email	prueba@gmail.com
Nombre y Apellido	Prueba
Contraseña
Confirmar Contraseña
<input type="checkbox"/> Administrador	Registrar Usuario

Figura 48: Prueba de llenado de formulario para registro de Usuario.

Actualización de usuario: 2

Correo	<input type="text" value="prueba@gmail.com"/>	<input type="radio"/> Admin
Nombre	<input type="text" value="Prueba"/>	<input checked="" type="radio"/> User
Contraseña	<input type="text"/>	
Repita contraseña	<input type="text"/>	
<input type="button" value="Actualizar usuario"/> <input type="button" value="Salir"/>		

Figura 49: Prueba de modificación de usuario.

6.4.4.2 Postcondiciones

Se ha iniciado sesión de manera exitosa con el usuario de prueba, como se observa en la figura 50, también, como se muestra en la figura 51 la lista de usuarios ha sido actualizada y se han verificado todas las funcionalidades que incluye este caso de uso, por lo tanto la prueba se ha tomado como exitosa.

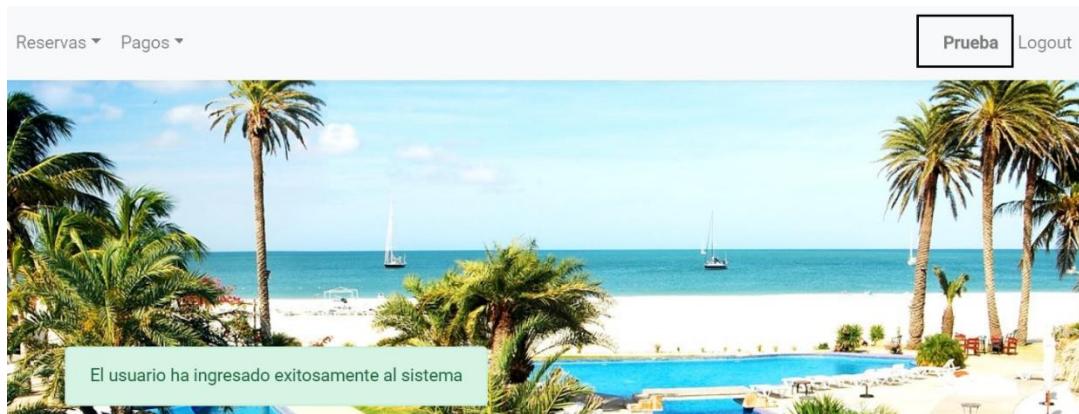


Figura 50: Prueba de Inicio de sesión.

Configuración					
Lista de Hoteles		Usuarios del Sistema		Políticas de Reserva	
#	Nombre	Email	Tipo	Eliminar	Editar
1	admin	admin@gmail.com	admin	Usuario principal	Usuario principal
2	Prueba	prueba@gmail.com	user	X	edit icon

[Registrar Usuario](#)

Figura 51: Prueba de consulta de usuarios del sistema.

Con las pruebas realizadas a lo largo de todo el sistema, se han evaluado las funcionalidades desarrolladas en su totalidad, se ha verificado que cada módulo realice las tareas para las cuales han sido diseñados, se ha constatado la integridad de los datos al guardarse y consultarse de la base de datos y se ha comprobado manualmente los cálculos matemáticos que realiza el sistema. Este proceso ha servido para depurar el software desarrollado de fallos ocurridos en las iteraciones realizadas, concluyendo así de forma exitosa esta fase del proyecto de investigación.

www.bdigital.ula.ve

C.C. Reconocimiento

Capítulo 7

Conclusiones y Recomendaciones

7.1 Conclusiones

En este trabajo de investigación han sido desarrollados los módulos necesarios para elaborar un Sistema Web de Centralización de Reservas, con el objetivo de permitir que se realicen los procesos básicos que componen la operación de los grupos hoteleros, como lo son consultar disponibilidad, realizar reservas e introducir pagos. De esta manera, al llevar a cabo estas operaciones en un mismo módulo de software el grupo hotelero genera información de manera unificada, que permite controlar de una forma adecuada los aspectos relacionados con la gestión, auditoria, mercadeo y comercialización, vitales para su bienestar comercial.

Para lograr este propósito, se ha seguido un marco de trabajo bien definido, que ha pasado por las etapas de especificación de requisitos, diseño arquitectónico, desarrollo de versiones y realización de pruebas, que han aportado los recursos necesarios para cumplir los objetivos planteados en este proyecto. Asimismo, un aspecto fundamental para lograr plasmar el producto de las fases de análisis y diseño en módulos de software, capaces de realizar todas las funciones necesarias para su buen funcionamiento, fue la implementación de los *frameworks* utilizados, ya que estos proveen la infraestructura y capacidades que fueron requeridas para obtener productos de software de calidad en un tiempo de desarrollo limitado.

Ha sido realizado un estudio acerca del entorno y el contexto en el cual estará inmersa la aplicación, de esta manera pudo diagnosticarse el proceso de reservación y se pudieron conocer los demás procesos que conforman la dinámica del funcionamiento del grupo hotelero. Luego de concluida esta etapa y con los datos recolectados, se inició a la fase de análisis de requerimientos de la investigación, esta permitió especificar cada uno de los aspectos funcionales de la aplicación por medio

de los diagramas de caso de uso, los cuales se consideran han descrito todas las funcionalidades necesarias para alcanzar los objetivos del proyecto y para abarcar las necesidades de los usuarios.

Por su parte, el diseño arquitectónico de la aplicación estuvo enmarcado bajo la arquitectura Modelo-Vista-Controlador, esta permitió definir cada una de las funcionalidades específicas para cada tipo de archivo, cada uno manejó la interfaz de usuario, la lógica de negocios y el acceso a datos respectivamente. Parte fundamental de este proceso fue el diseño de la base de datos, especificado mediante un diagrama de clases que permite observar de manera clara todos los atributos que posee cada entidad modelada así como las interacciones que ocurren entre ellas, se establecieron criterios y restricciones que determinan cómo serán almacenados los datos y cómo será el acceso a ellos. Además, se especificó mediante un diagrama cada una de las interfaces que fueron elaboradas posteriormente en la etapa de desarrollo de versiones, esto sirvió para especificar cada una de las características necesarias para que cada interfaz pudiera soportar las actividades que se desempeñarían en ellas, así como para brindar a los usuarios una experiencia positiva al usar la aplicación.

La etapa de desarrollo de versiones consistió en tres iteraciones en las cuales se lograron abarcar todas las necesidades funcionales especificadas en la fase de diseño y análisis del proyecto, durante este proceso se refinaron los requerimientos del software y gracias a la metodología utilizada, se realizaron ajustes y validaciones a medida que el proceso se llevó a cabo. Un plan de pruebas fue ejecutado para validar que cada módulo realizado cumple con los requisitos planteados y cuenta con la calidad necesaria.

Finalmente, cada objetivo específico de la investigación fue alcanzado y cumplido en las fases que componen este proyecto de investigación, logrando diseñar y desarrollar una aplicación adecuada y capaz de realizar todas las actividades que componen el entorno empresarial para el cual fue diseñada.

7.2 Recomendaciones

Se realizan una serie de recomendaciones orientadas a la mejora de la calidad del software y las funcionalidades que ofrece:

- Establecer un marco de pruebas que incluya la comprobar todas las funcionalidades del sistema en un entorno de red, en el que tomen parte los equipos de comunicación entre equipos. La configuración de este tipo de entorno de trabajo depende de las necesidades comunicacionales

de la empresa y es de gran relevancia realizar este tipo de pruebas antes de realizar la implementación del sistema.

- Incluir el envío automático de pagos al sistema de gestión *Argus*, ya que de esta manera se replica en su totalidad la información recibida a través del sistema de reservaciones en cada uno de los sistemas de gestión de las sedes hoteleras.
- Agregar la funcionalidad para que puedan ser observados con gráficos de barra o circulares, los distintos indicadores de operación, como la distribución de ventas por sede y producción monetaria, esto permitirá presentar de manera resumida y visual estos datos numéricos para su interpretación por parte del personal indicado.
- Agregar la funcionalidad para obtener un reporte en formato *PDF* sobre los pagos recibidos. Una gran cantidad de criterios deben ser proporcionados al usuario para poder obtener la información tan específica como sea necesaria.
- Agregar la funcionalidad para permitir crear y aplicar cupones de descuento, estos permiten al grupo hotelero promocionar sus servicios de una manera más atractiva y permite que tenga un mejor posicionamiento en el mercado al ofrecer mejores servicio a un precio más competitivo. Su representación en la base de datos ya se encuentra prevista.
- Agregar la funcionalidad, capacitada por los *frameworks* utilizados, de enviar automáticamente correos informativos al cliente al momento de realizar una reserva o un pago. Esto mejora notablemente el rendimiento del departamento de reservas, ya que libra de este proceso al usuario e informa inmediatamente al cliente que se encuentran bajo gestión sus solicitudes.
- Agregar un módulo de *Dashboard*, que funcione como tablero para ver todas las operaciones realizadas diariamente. Debe permitir ver todas las reservas, su estatus, su hotel destino, operador que la realizó, si la reserva posee un pago, entre otros. Este módulo ofrece un fácil y sencillo seguimiento de todas las operaciones realizadas por los usuarios.

Bibliografía

- Acosta, C. S., Monteza, V. T., & Cabrera, I. M. (2015). ANÁLISIS COMPARATIVO DE FRAMEWORKS PARA EL DESARROLLO DE APLICACIONES WEB EN JAVA. *Ingeniería: Ciencia, Tecnología e Innovación*, 2(1), 60-60.
- Bass, L., Clements, P., & Kazman, R. (2003). Software architecture in practice. Addison-Wesley Professional.
- Becerra, R. L., & Cerveleón, L. J. (2019). ANÁLISIS DE APROPIACIÓN DE LAS TIC EN EL SECTOR HOTELERO DE LA CIUDAD DE PAMPLONA. *SEMILLEROS DE INVESTIGACIÓN*, 1(2).
- Belloch, C. (2012). Las Tecnologías de la Información y Comunicación en el aprendizaje. Departamento de Métodos de Investigación y Diagnóstico en Educación. Universidad de Valencia.
- Bernal Pablo, P. (2018). *La Investigación en Ciencias Sociales: Técnicas de recolección de la información*. Universidad Piloto de Colombia.
- Bierman, G., Abadi, M., & Torgersen, M. (2014, July). Understanding typescript. In European Conference on Object-Oriented Programming (pp. 257-281). Springer, Berlin, Heidelberg.
- Boehm, B. W. (1984). Verifying and validating software requirements and design specifications. *IEEE software*, 1(1), 75.

Buhalis, D. y Law, R. (2008). Progress in information technology and tourism management: 20 years on and 10 years after the Internet—The state of eTourism research. *Tourism Management*, 29(4), 609-623.

Canós, J. H., & Letelier, M. C. P. P. (2012). Metodologías ágiles en el desarrollo de software.

Cardozzo, D. R. (2016). Desarrollo de Software: Requisitos, Estimaciones y Análisis. IT Campus Academy.

Caro Encalada, M. J. (2008). El uso de las tecnologías de la información y comunicación en el sector hotelero de la Península de Yucatán: hacia un modelo explicativo (Tesis doctoral). Universidad Politécnica de Madrid, Madrid, España

Cobo, Á. (2007). Diseño y programación de bases de datos. Editorial Visión Libros.

Date, C. J. (2001). *Introducción a los sistemas de bases de datos*. Pearson Education.

Domecq, C. F., & Ballesteros, P. T. (2006). La conectividad entre TICs para la gestión de la información en el sector hotelero madrileño. In Turitec 2006: VI Congreso nacional turismo y tecnologías de la información y las comunicaciones (pp. 227-241). Universidad de Málaga (UMA).

Heiderich, M., Nava, E. A. V., Heyes, G., & Lindsay, D. (2011). PHP. Web Application Obfuscation, 151–175. doi:10.1016/b978-1-59749-604-9.00006-6

Hernández, J. (2014). Análisis y desarrollo web. Jesús Hernández.

Hills, M., Klint, P., & Vinju, J. (2013, July). An empirical study of PHP feature usage: a static analysis perspective. In Proceedings of the 2013 international symposium on software testing and analysis (pp. 325-335). ACM.

Escalona, M. J., & Koch, N. (2002). Ingeniería de Requisitos en Aplicaciones para la Web—Un estudio comparativo. Universidad de Sevilla.

Falquemberg, C. (2012). Pymes Hoteleras Gestión del Conocimiento, para una mayor Competitividad. Revista Electrónica CECIET, 2(1).

Ivanova, M., & Ivanov, S. (2015). The nature of hotel chains: An integrative framework. International Journal of Hospitality & Tourism Administration, 16(2), 122-142.

Jacobson, I., Booch, G., Rumbaugh, J. (2000). El proceso unificado de desarrollo de Software.

Jain, N., Bhansali, A., & Mehta, D. (2015). AngularJS: A modern MVC framework in JavaScript. Journal of Global Research in Computer Science, 5(12), 17-23.

Law, R., Leung, D., Au, N., & Lee, H. A. (2013). Progress and development of information technology in the hospitality industry: Evidence from Cornell Hospitality Quarterly. Cornell Hospitality Quarterly, 54(1), 10-24.

Melton, J., & Simon, A. R. (1993). Understanding the new SQL: a complete guide. Morgan Kaufmann.

Oliveros Contreras, Diana; Martínez, Gabriel Mauricio Efecto de las TIC sobre la gestión de las empresas hoteleras afiliadas a Cotelco de Bucaramanga Santander, Colombia Revista EAN, núm. 83, Julio-Diciembre, 2017, pp. 15-30 Universidad EAN.

Pantoja, L., & Pardo, C. (2016). Evaluando la Facilidad de Aprendizaje de Frameworks MVC en el Desarrollo de Aplicaciones Web. Publicaciones e Investigación, 10, 129-142.

PHP Usage Statistics. <http://w3techs.com/technologies/details/pl-php/all/all>

Pressman, R. S. (2002). Ingeniería del Software: Un enfoque práctico 5a. Edición.

Ramos, M., Valente, M. T., Terra, R., & Santos, G. (2016, November). AngularJS in the wild: A survey with 460 developers. In Proceedings of the 7th International Workshop on Evaluation and Usability of Programming Languages and Tools (pp. 9-16). ACM.

Rivera, F. L., & León, F. (2008). Bases de Datos relacionales: Teoría y Práctica. Editorial Textos Académicos del Instituto Tecnológico Metropolitano. Medellín, Colombia.

Solis, C. (2015). Manual Del Guerrero: AngularJS.

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1). Upper Saddle River: Prentice Hall.

Sánchez, C., Tuesta, V. y Mejía, I., Análisis comparativo de framework para el desarrollo de aplicaciones web en java– Rev. Ingeniería: Ciencia, Tecnología e Innovación VOL 1/Nº 2, ISSN: 2313-1926/Julio 2015

Stauffer, M. (2019). Laravel: Up & Running: A Framework for Building Modern PHP Apps. O'Reilly Media.

TIOBE Programming Community Index.

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.