



UNIVERSIDAD
SAN IGNACIO
DE LOYOLA

FACULTAD DE INGENIERÍA

Carrera de Ingeniería Informática y de Sistemas

IMPLEMENTACION DE UN PMS PARA MEJORAR LA GESTIÓN DE EMPRESAS DEL SECTOR HOTELERO

**Trabajo de Suficiencia Profesional para optar el Título
Profesional de Ingeniero Informático y de Sistemas**

**CARLOS ALBERTO NORABUENA ARANDA
(0000-0001-5291-956X)**

**Asesor:
Mg. SILVIA CASTRO BURGOS
(0000-0003-2493-8362)**

**Lima – Perú
2019**

Dedicatoria

Este trabajo está dedicado a Fátima Norabuena Guerra por inspirarme una filosofía de vida. Que la felicidad del prójimo es el motivo más grande de sacrificio que tiene la persona, superando a cualquier motivo material o intelectual.

INDICE

Introducción	5
Generalidades de la Empresa.....	6
Datos Generales.....	6
Razón Social de la Empresa	6
Ubicación de la empresa	6
Giro de la Empresa.....	7
Tamaño de la Empresa	7
Reseña histórica de la Empresa	7
Organigrama de la empresa.....	8
Misión, visión y políticas.....	8
Políticas de la Empresa.....	8
Reglas de la Empresa	8
Productos.....	9
Clientes.....	10
Relación de la empresa con la sociedad	10
Planteamiento del problema que fue abordado.....	11
Caracterización del área	11
Definición del Problema	14
Objetivos.....	18
Justificación	18
Alcances y limitaciones	18
Marco teórico.....	20
Property Management System (PMS)	20
Arquitectura de software.	23
Lenguaje de programación.....	25
Framework de desarrollo.....	25
Software as a Service (SAAS)	27
Infraestructura como Servicio (IAAS).....	27
Latencia de Red	29
Diseño y desarrollo de software.....	32
Desarrollo del proyecto	34
Análisis del problema	34
Diseño de software.....	36
Diseño de la base de datos.....	47
Implementación	55

Pruebas.....	67
Despliegue o pase a producción.....	73
Análisis económico.....	74
Análisis y resultados.....	78
Conclusiones	85
Recomendaciones.....	86
Referencias.....	87
Bibliografía.....	90

Introducción

Desde la aparición de los ordenadores, éstos han sido usados en los sistemas de información de las empresas para facilitar el trabajo de gestión gracias a la capacidad de recopilar y procesar gran cantidad de datos. Martínez, Majó y Casadesús (2006) señalan que las inversiones en tecnologías de la información incrementan de forma significativa la competitividad de la empresa.

En el caso de los hoteles, el sistema más recomendado es un sistema de gestión de establecimientos (PMS por sus siglas en inglés) porque permite ahorrar costos de mano de obra al reducir tiempos en los procesos principales, incrementar ventas al conectar el negocio con canales de venta externos y facilitar los procesos de gestión.

En ciudades como España el nivel de penetración de los PMS bordea el 75% (Grant Thornton, 2018) mientras que en países como Perú el nivel desciende al 45%. Cifras del Ministerio de Comercio Exterior y Turismo (MINCETUR) en un estudio del 2018 señalan que la población de establecimientos hoteleros en Perú fue de 22,000. Ese estudio solo incluyó a los establecimientos formales.

Esa necesidad desatendida motivó el emprendimiento empresarial del que se basa este informe. El emprendimiento consistió en el diseño y desarrollo de un software PMS denominado "XAFIRO, Sistema de Gestión Hotelera". Y fue implementado en diez establecimientos hoteleros ubicados en los departamentos de Arequipa, Lima, Trujillo, Amazonas y Tumbes. Todos estos establecimientos tenían una gestión orientada a procesos manuales.

Los resultados demostraron un ahorro de tiempo de hasta un 80% en los procesos de reservas, cierre de caja y gestión de inventarios. Además, por primera vez los administradores disponían de indicadores de gestión de sus áreas principales.

El documento consta de ocho capítulos: Generalidades de la Empresa (capítulo dos), Planteamiento del problema (capítulo tres), Marco teórico (capítulo cuatro), Desarrollo del proyecto (capítulo cinco), Análisis de Resultados (capítulo seis), Conclusiones (capítulo siete) y Recomendaciones (capítulo ocho).

Generalidades de la Empresa

Datos Generales

ABBOST es una empresa peruana dedicada al diseño y desarrollo de sistemas informáticos. Desde el año 2013 ABBOST ha desarrollado software para diversas industrias como hotelería, gastronomía, retail, educación, logística y otros.

ABBOST también ha desarrollado sistemas con marca propia como XAFIRO: Sistema de gestión hotelera, FACTURACIÓN ELECTRÓNICA PYME, y XAFIRO CLUB: Sistema de fidelización.

Razón Social de la Empresa

ABBOST EIRL.

Ubicación de la empresa

Dirección.

Calle Forestales 525, distrito de La Molina, provincia de Lima.

Teléfono.

+511 677 2782

Mapa de ubicación.



Figura 1. Ubicación de oficinas de Abbost. Fuente: Google Maps

Giro de la Empresa

Diseño y desarrollo de software.

Tamaño de la Empresa

ABBOST es una microempresa (MYPE) y cuenta en la actualidad con 5 colaboradores.

Reseña histórica de la Empresa

Sistema Hotelero Xafiro se creó durante los años 2014 y 2015 por iniciativa de Alberto Norabuena Aranda. Se desarrolló como un sistema de gestión para hoteles con opciones básicas. Conforme pasó el tiempo se fueron agregando más funcionalidades al sistema según el aporte de los usuarios.

En el 2017 el equipo de Xafiro participó en la competencia StartUp USIL 4G. En esta competencia no se llegó a instancias finales, pero se pudo obtener importantes comentarios de expertos que fueron la base para la actualización del software.



Figura 2. Logo del sistema XAFIRO. Fuente: www.sistemahoteleroxafiro.com

A partir del 2018 se han ido haciendo cambios y actualizaciones al sistema XAFIRO. Un cambio importante fue la integración con el portal de reservas BOOKING.COM, la implementación de la facturación electrónica y el rediseño del logotipo por uno más minimalista.



Figura 3. Logo del sistema XAFIRO. Fuente: www.sistemahoteleroxafiro.com

Organigrama de la empresa

El organigrama de la empresa tiene una estructura horizontal y está conformada por las siguientes áreas.

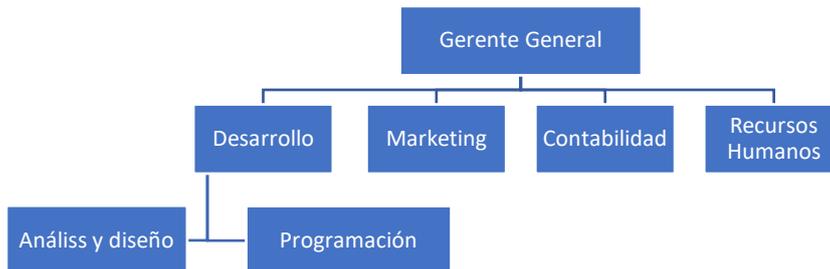


Figura 4. Organigrama de la empresa ABBOST al 2019. Fuente: Elaboración propia.

Misión, visión y políticas

Misión.

ABBOST es una fábrica de software. Tiene la misión de entregar a los clientes soluciones que agreguen valor a su negocio, que los incluyan tecnológicamente y los vuelvan más competitivos en el mercado.

Visión.

ABBOST pretender ser una fábrica de software líder en Perú y Latinoamérica. Que sea reconocida por su innovación, tiempo de respuesta y atención post-venta.

Políticas de la Empresa

No existe un documento establecido de políticas de la empresa.

Reglas de la Empresa

Para elaborar las reglas de la empresa se tomaron como referencia los estatutos de ITIL (2019) y las recomendaciones de seguridad del Cloud Security Guidance (2017).

Reglas de Trabajo.

- No desarrollar ninguna línea de código hasta que el documento Alcance del Proyecto no esté aprobado y firmado por el cliente.
- Publicar los cambios realizados en los proyectos en el repositorio de la empresa.
- Testear cualquier cambio en el servidor de prueba de la empresa

Reglas de Atención al Cliente.

- Priorizar las incidencias de clientes en prioridad alta, media y baja.
- La prioridad alta se asigna cuando un servicio se ve afectado de manera severa impidiendo su uso y afectando a actividades críticas de negocio. Tiempo de atención de hasta 1 hora.
- La prioridad media se asigna cuando un servicio se ve afectado impidiendo su uso, pero no afectando a actividades críticas de negocio. Tiempo de atención de hasta 4 horas.
- La prioridad baja se asigna cuando un servicio se ve afectado, pero no impide su uso. Tiempo de atención de hasta 24 horas.

Reglas de seguridad.

- Cambiar obligatoriamente las contraseñas cada 30 días.
- Usar contraseñas seguras compuestas por letras y símbolos.
- Encriptar las contraseñas y códigos de verificación antes de su almacenamiento en la base de datos.

Productos

ABBOST ofrece al mercado 2 líneas de producción: Diseño y desarrollo de software con marca propia, y diseño y desarrollo de software con marca de terceros.

Tabla 1

Tipos de productos comercializados con marca propia

Nombre	Descripción
Sistema Hotelero Xafiro	Sistema Web para gestionar los procesos dentro un hotel.
Facturación Electrónica Pyme	Sistema Web para la emisión de comprobantes electrónicos para SUNAT.
Xafiro Club: Sistema de Fidelización	Aplicación móvil que incentiva o premia las recomendaciones de los clientes por su marca favorita.

Tabla 2

Tipos de productos desarrollados para terceros a partir del año 2017

Nombre	Descripción
Páginas Web	Desarrollo de sitios web que buscan mostrar información de productos y medios de contacto.
E-Commerce	Desarrollo de carritos de compra y formularios de pagos con pasarelas de pago con VISA y/o MASTERCARD.
ERP	Sistema para planificar o gestionar los recursos de una empresa. Permite gestionar el área de ventas, compras, almacén, atención al cliente y auditoría.
Aplicaciones móviles	Desarrollo de aplicaciones móviles para las tiendas virtuales de PlayStore y AppStore.

Clientes

ABBOST tiene un portafolio amplio de clientes en diversas industrias como por ejemplo retail, educación, retail, salud, entretenimiento, hotelería, turismo y gastronomía. Los clientes se encuentran ubicados en varios distritos de Lima y Provincia.

Relación de la empresa con la sociedad

ABBOST busca -a través de sus trabajos realizados- mejorar la calidad de vida de los usuarios. Una forma de hacerlo es reduciendo tiempo en actividades operativas manuales para que el usuario lo invierta en actividades de mayor valor. Otra forma de hacerlo es fomentando el teletrabajo a través de sus sistemas web para que los usuarios puedan trabajar desde cualquier lugar con conexión a internet.

Planteamiento del problema que fue abordado

Caracterización del área

El turismo es uno de los principales motores económicos a nivel mundial. Representa el 10.4% del PBI mundial y genera 1 de cada 10 empleos según la asociación empresarial World Travel & Tourism Council (WTTC). El sector turismo reúne las actividades de transporte, gastronomía, hotelería y afines.

Precisamente el sector de hotelería, conformado -en su mayoría- por empresas pequeñas y medianas (PYMES), desempeña un papel importante dentro del turismo y su correcta gestión es clave para sostener la economía de un país. En países como Reino Unido, España, Italia, Francia y Estados Unidos el 70% de los administradores de hoteles apoyan su gestión con el uso de un sistema informático (Grand Thornton, 2018)

La realidad en Perú es por el contrario distinta, la participación del sector hotelero tan solo representa el 0.55% del PBI nacional (MINCETUR, 2018) y solo el 45% de estas empresas usa algún sistema de información (DIARIO GESTION, 2017).

De los diez establecimientos hoteleros que participaron de este informe el 100% de ellos utilizaba métodos manuales para la gestión de sus procesos de reservas, checkin, checkout, arqueo de caja y gestión de inventarios. A continuación, se menciona una breve descripción de cada una de las empresas.

Hotel Sunset.

Ubicado en el distrito de Chaclacayo, departamento de Lima. Es una empresa familiar MYPE. Cuenta con 18 habitaciones y sus tarifas van entre 90 a 150 soles. Cuenta con un equipo de 6 colaboradores entre personal de recepción, lavandería, cocina y administración.

Hostal Albareda.

Ubicado en el distrito de Ate, departamento de Lima. Es una empresa familiar MYPE. Cuenta con 32 habitaciones. Sus tarifas oscilan entre 70 a 120 soles. Cuenta con un equipo de 7 colaboradores entre personal de recepción, lavandería y administración.

Hotel Queen.

Ubicado en el distrito de Ate, departamento de Lima. Es una empresa familiar MYPE. Cuenta con 30 habitaciones. Sus tarifas oscilan entre 70 a 200 soles. Cuenta con un equipo de 7 colaboradores entre personal de recepción, lavandería y administración.

Hotel Washington.

Ubicado en el distrito de San Juan de Lurigancho, departamento de Lima. Es una empresa pequeña. Cuenta con 96 habitaciones. Sus tarifas oscilan entre 50 a 120 soles. Cuenta con un equipo de 10 colaboradores entre personal de recepción, lavandería y administración.

Hotel Norte.

Ubicado en la provincia de Trujillo, departamento de Trujillo. Es una empresa familiar MYPE. Cuenta con 32 habitaciones. Sus tarifas oscilan entre 100 a 180 soles. Cuenta con un equipo de 8 colaboradores entre personal de recepción, lavandería, cocina y administración.

Hotel Victoria.

Ubicado en el distrito de La Victoria, departamento de Lima. Es una empresa pequeña. Cuenta con 78 habitaciones. Sus tarifas oscilan entre 90 a 200 soles. Cuenta con un equipo de 10 colaboradores entre personal de recepción, lavandería, cocina y administración

Hotel Molina.

Ubicado en el distrito de La Molina, departamento de Lima. Es una empresa familiar MYPE. Cuenta con 38 habitaciones. Sus tarifas oscilan entre 70 a 120 soles. Cuenta con un equipo de 7 colaboradores entre personal de recepción, lavandería y administración.

Hotel Misti.

Ubicado en el departamento de Arequipa. Es una empresa familiar MYPE. Cuenta con 14 habitaciones. Sus tarifas oscilan entre 70 a 150 soles. Cuenta con un equipo de 7 colaboradores entre personal de recepción, lavandería y administración.

Hotel Frontera.

Ubicado en el departamento de Tumbes. Es una empresa pequeña. Cuenta con 20 habitaciones. Sus tarifas oscilan entre 120 a 240 soles. Cuenta con un equipo de 6 colaboradores entre personal de recepción, lavandería, cocina y administración.

Hotel Gocta.

Ubicado en la provincia de Chachapoyas, departamento de Amazonas. Es una empresa familiar MYPE. Cuenta con 24 habitaciones. Sus tarifas oscilan entre 60 a 100 soles. Cuenta con un equipo de 6 colaboradores entre personal de recepción, lavandería y administración.

Antecedentes

Martínez, Majó y Casadesús en su estudio “El uso de las tecnologías de la información en el sector hotelero” comentan que el primer PMS fue instalado en el Hotel Sheraton de Waikiki en 1970. En un primer momento estaban pensados para dar solución a las necesidades de los procesos del “front office”, es decir los que tienen relación directa con el cliente. Pero con los años han ido aglutinando más funcionalidades tanto del “front office” como del “back office” (2006)

Kiyamu en su tesis para obtener la licenciatura, “La calidad de diseño de un Property Management System y la utilidad en hoteles tres estrellas de Miraflores”, señala que la mejora del rendimiento de los colaboradores tiene relación con la calidad de la información; pues si la información que obtienen con el PMS es toda la necesaria, exacta, actualizada y fácil de entender; entonces el rendimiento de los trabajadores mejora, pues no retrasan sus funciones diarias y finalizan sus tareas en el tiempo adecuado. (2018).

La automatización que ofrecen los PMS permite recopilar datos (fechas, tarifas, preferencias, datos demográficos, etc) de manera estática a medida que se producen como parte de un procedimiento estándar o bien de manera dinámica, actualizándose en cada periodo determinado de tiempo. Posteriormente, estos datos, permitirán a través de herramientas de Business Intelligence (BI) la generación de informes para la toma de decisiones. (Palao D. 2018)

Definición del Problema

La mayor parte de las PYMES en Perú no superan el segundo año de funcionamiento debido entre otras cosas a la falta de innovación tecnológica (Ponce Fátima y Zevallos Emilio, 2017).

Esta carencia de innovación se comprobó en los hoteles elegidos para este informe. El 100% de ellos usaban hojas para sus registros de datos con limitaciones de capacidad, análisis y obtención de datos específicos. Los administradores reclamaban a sus recepcionistas una información ordenada y legible que le permitiera analizarla fácilmente. En la práctica no se daba esta situación.

Debido a su administración manual no se podía llevar un control frecuente de productos vendidos, no existía un cárdex. Según entrevistas hecha a los administradores las pérdidas de productos eran frecuentes y no se tenía un estimado de lo perdido. Asimismo, la pérdida de dinero era recurrente por los errores ocasionados por cálculos manuales.

Otro problema que requería la atención inmediata de la mayoría de los administradores era la alta rotación de personal. Esta rotación -en el caso de los recepcionistas- era motivada principalmente por el sobretiempo no remunerado y los descuentos salariales que debían asumir cada vez que su caja no les cuadraba. La tasa de rotación promedio era del 61% y estaba por encima del estudio realizado por IPSOS APOYO (2014), que indicaba que el sector hotelero presentaba una tasa de rotación del 18%.

Se encontró también que la relación hotel-huésped era muy débil porque el historial de consumo o preferencias quedaba en la mente de quien lo atendió. Con esto se reducían notablemente las oportunidades de ventas dirigidas y de empatía con los huéspedes.

Un último problema, pero no menos importante fue que el administrador no tenía indicadores de gestión actualizados. No tenía métricas de venta totales, ni por tipo de habitación, ni por productos. Esto ocasionaba que las decisiones que tomaba sean muy improvisadas y empíricas.

Con esos datos se pudo identificar el problema central. Para ello se necesitó el uso de 3 herramientas de análisis: el diagrama causa-efecto conocido como Ishikawa, el árbol de problemas y la matriz FODA. La conclusión a la que se llegó fue que el problema era la ineficiente gestión del administrador, propiciada por el mal uso de los recursos y exceso de prácticas manuales.

Diagrama Ishikawa.

Según las bases de elaboración del diagrama Ishikawa se listaron todas las posibles causas del problema -a modo de espinas- como por ejemplo los cálculos manuales, el tiempo extra no remunerado, las decisiones improvisadas, entre otros. Llegando a la conclusión que la gestión de los recursos del hotel se estaba haciendo de manera ineficiente.

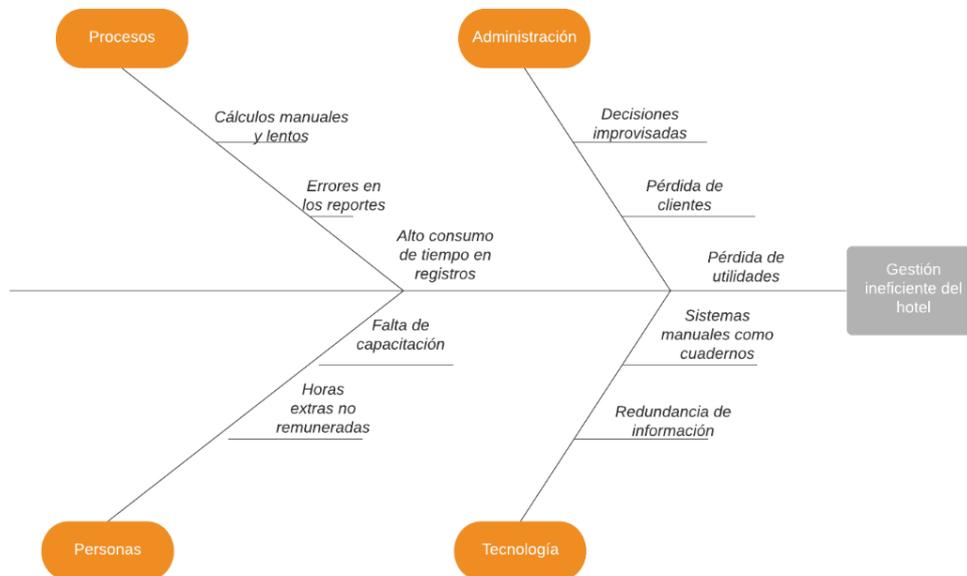


Figura 5. Diagrama Ishikawa. Elaboración propia

Árbol de Problemas.

Para confirmar y reforzar el diagrama Ishikawa se diseñó un árbol de problemas. De este modo no solo identificaríamos al problema, sino que se evaluaría qué impactos podría tener en la empresa.

Los efectos del problema serían las hojas mientras que las causas serían las raíces del árbol. Se pudo confirmar que el problema principal era la "Gestión hotelera ineficiente" motivada por el exceso de operaciones manuales como sumas y cálculos, y errores manuales como datos incompletos y pérdida de apuntes.

Como consecuencia de esa gestión ineficiente se estaba desmotivando al personal, de ahí se explica la alta rotación. Se estaba perdiendo competitividad y por tanto era una oportunidad para que las empresas competidoras se hagan de los clientes del hotel. Y lo peor podría llegar en cualquier momento si es que no estaba pasando, la pérdida de utilidades.

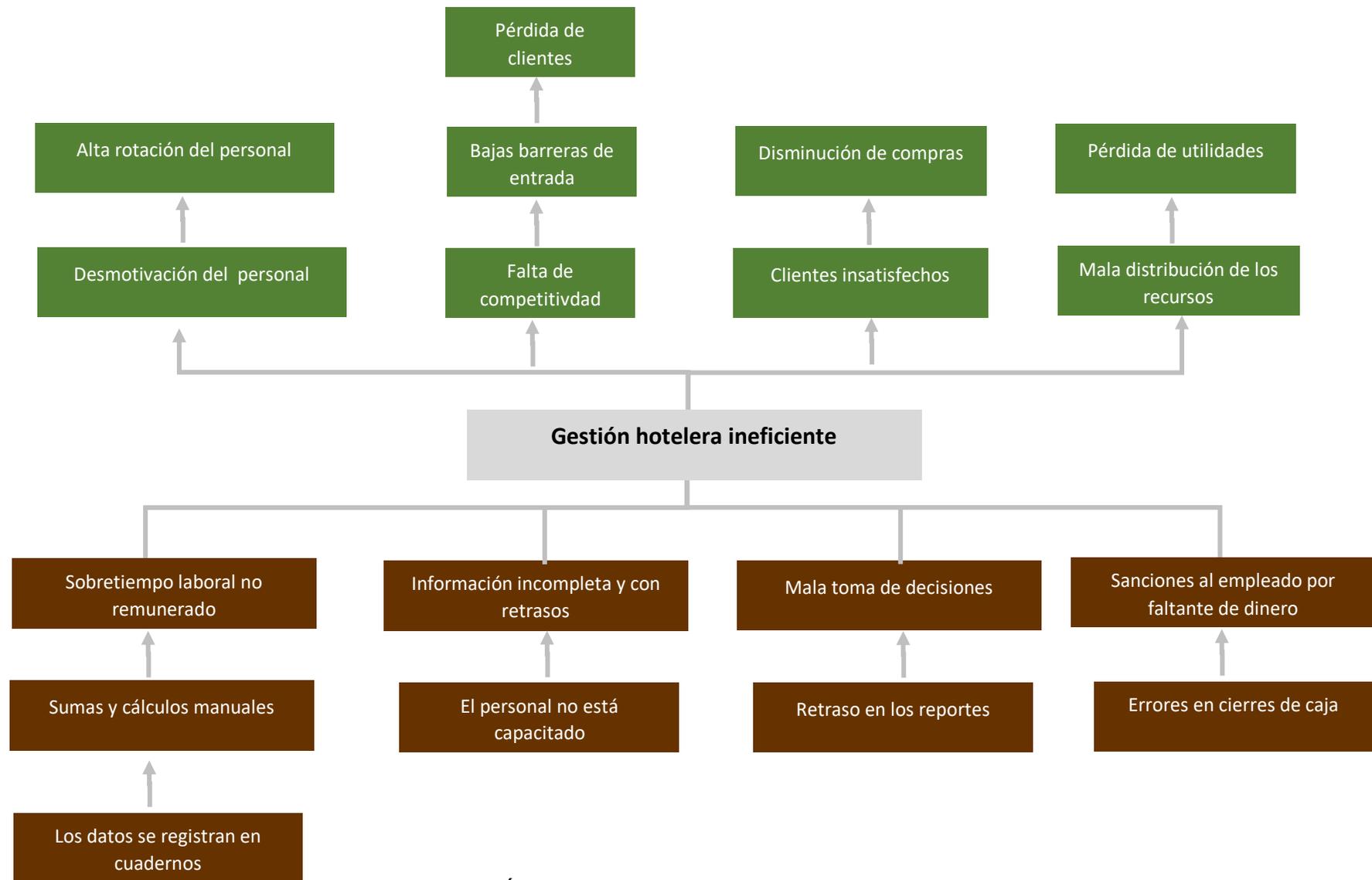


Figura 6. Árbol de problemas. Fuente: Elaboración propia

Matriz FODA.

La matriz FODA fue útil para plantear alguna solución o soluciones a la problemática antes mencionada.

Tabla 3

Matriz FODA para los hoteles en investigación

	OPORTUNIDADES 1. Impulsar las ventas por internet 2. Aceptar pagos con tarjeta de crédito o débito.	AMENAZAS 1. La tasa de crecimiento del sector hotelero en la zona. 2. Taxistas llevan a los pasajeros a otros hoteles.
FORTALEZAS: 1. Muy buena atención al cliente. 2. Imagen corporativa (uniformes, decoración y utilería). 3. Venta de servicios adicionales.	ESTRATEGIAS FO <ul style="list-style-type: none">• Desarrollar un sistema informático para gestionar las reservas por internet (O1).• Desarrollar una página web mostrando testimonios de clientes (F1, O1).• Contratar el servicio de pago con VISA (O2).	ESTRATEGIAS FA <ul style="list-style-type: none">• Dar comisión a los taxistas para poner marca del hotel en los taxis.
DEBILIDADES: 1. Rotación alta del personal. 2. Usan libros y cuadernos para registro de datos. 3. Muchas incidencias negativas de clientes. 4. Pérdidas de dinero por cálculos erróneos. 5. Pérdida de información de clientes. 6. Uso de un único canal de ventas. 7. Falta de capacitación o instrucción del administrador.	ESTRATEGIAS DO <ul style="list-style-type: none">• Implementar un sistema informático (D2, D4, D5 y D6).• Implementar una central de alertas en el hotel (D3).• Crear una página web para realizar reservas en el hotel (D6, O1).	ESTRATEGIAS DA <ul style="list-style-type: none">• Crear promociones para retener clientes (A1).

Fuente: Elaboración propia

De este análisis se concluye que un sistema informático podría corregir las debilidades de los procesos manuales y aprovechar las oportunidades que da el mercado.

Objetivos

Objetivo general.

- Implementación de un PMS para mejorar la gestión de empresas del sector hotelero.

Objetivos específicos.

- Reducción de tiempo en el proceso de reservas.
- Reducción de tiempo en el proceso de cierre de caja.
- Reducción de tiempo en la gestión de inventarios.
- Reducción de la tasa de rotación.
- Creación de indicadores de gestión.

Justificación

La importancia de mejorar la gestión de las empresas del sector hotelero se sustenta por los siguientes aspectos:

Conveniencia.

Se aprecia que existe una necesidad real insatisfecha de parte de los administradores. Si se logra satisfacer esa necesidad se podrá confirmar que Xafiro es el sistema ideal para los hoteles. Por tanto, habrá más interesados en adquirir Xafiro.

Implicancias prácticas.

Existe un consumo alto de tiempo en la generación de reportes elaborados manualmente. Se demostrará que con Xafiro una operación que normalmente toma varios minutos se puede realizar en un par de minutos. Asimismo, los errores aritméticos se podrían evitar con cálculos automatizados.

Alcances y limitaciones

Es necesario precisar que la tecnología y metodologías utilizadas en el informe corresponden a las que el autor estimo conveniente durante el año 2014 y 2016, años en el cual fue desarrollado el proyecto.

El proyecto estuvo enfocado en el diseño, desarrollo e implementación de un sistema informático tipo PMS. Contempla la automatización de los procesos de reservas, checkin,

checkout, cierre de caja, gestión de inventarios, reportes y analíticas. Están fuera del alcance los procesos relacionados a compras, finanzas, recursos humanos, contabilidad, channel manager y CRM.

Asimismo, no fueron incluidas las capacitaciones al administrador en temas de interpretación de indicadores y elaboración de promociones.

Una restricción final respecto a los clientes fue no revelar el nombre real de los hoteles en la muestra de resultados.

Marco teórico

Property Management System (PMS)

En el sentido literal, un PMS es un sistema informático que administra una propiedad o propiedades. Tiene como principales objetivos la reducción del costo de mano de obra y facilitar la gestión de los procesos del hotel.

El primer PMS fue instalado en el hotel Sheraton de Wakiki en 1970. Tenía como función gestionar el inventario de habitaciones libres. Fue después de 1980 que los PMS evolucionaron categóricamente hasta convertirse en la poderosa herramienta de gestión que ahora son (Martínez, Majó y Casadesús, 2006).

A continuación, los principales módulos y procesos que gestiona un PMS

- Clientes: Perfiles, historial y contacto.
- Recepción: Reservas, checkin, checkout y facturación.
- Limpieza: Horarios, asignación de tareas, inventario de insumos.
- Back-office: Libro mayor, cuentas por cobrar, cuentas por pagar y depósitos. Reportes y analítica

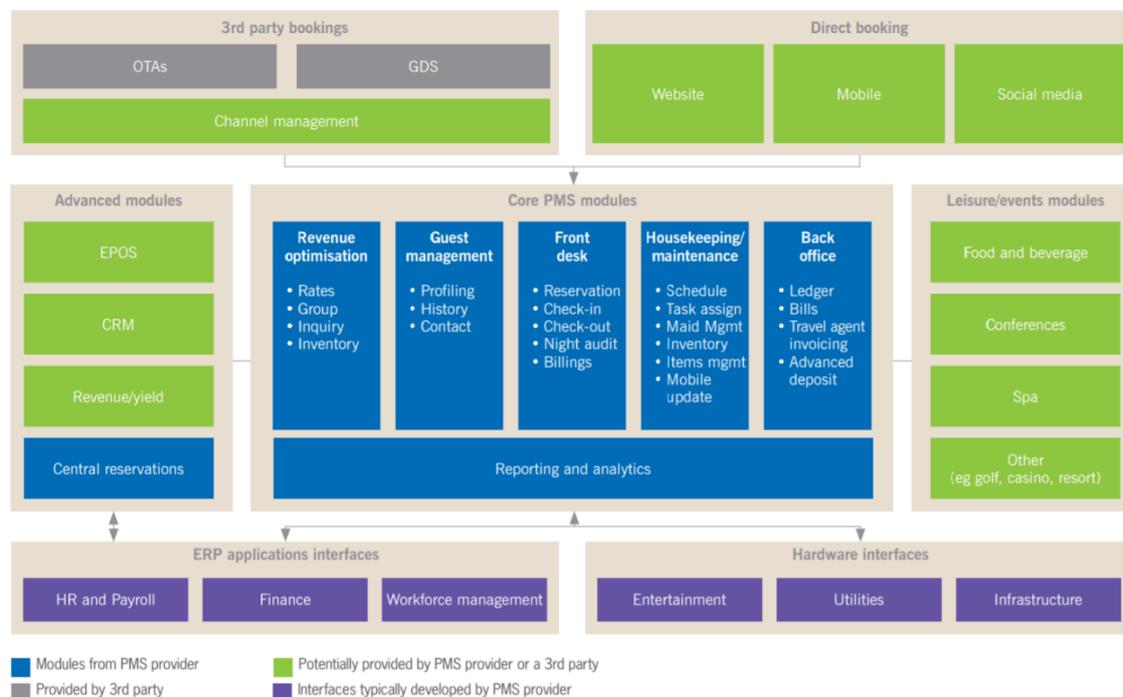


Figura 7. Módulos de un PMS. Fuente: Grand Northon (2018)

Para comprender mejor la utilidad de un PMS se cita a Carlos Diez de La Lastra, CEO de la universidad Les Roches Marbella catalogada como la tercera mejor a nivel mundial en formación de servicios hoteleros: “El PMS es un sistema imprescindible en los hoteles. hoy en día no se puede gestionar correctamente un hotel sin tener un PMS activo” (2018).

Larrañaga también afirma que un PMS permite automatizar las funciones del hotel como las reservas de clientes, reservas online, puntos de venta, teléfono, cuentas por cobrar, ventas y marketing, banquetes, alimentos y bebidas, costos y administración, gestión de materiales, recursos humanos y nómina, gestión de mantenimiento, gestión de calidad y otros servicios para un mejor flujo de la información dentro de la organización (2015).

Según Mark Hurd, CEO de la compañía estadounidense ORACLE (2018) las principales ventajas de un PMS son las siguientes:

- Menos costo de personal: Esto se justifica en la capacidad que tiene el sistema para automatizar y simplificar todas aquellas funciones que son más lentas y repetitivas. De ese modo el personal puede ser asignado a tareas que generen mayor valor.
- Mas clientes: Permite gestionar todas las reservas que lleguen al hotel por los diferentes canales de atención.
- Checkin / Checkout: Permite registrar las entradas y salidas de una manera rápida. Asimismo, permite agilizar las tareas de limpieza y mantenimiento.
- Control y seguridad: Aporta accesos al sistema en base a permisos o tipos de usuario. El acceso a la información es restringido.
- Habitaciones: Se puede consultar el estado de las habitaciones de una manera sencilla.
- Reportes: Se puede obtener diferentes tipos de informes para analizarlos, compararlos y enviarlos a la gerencia.

PMS in-house vs PMS in-cloud

Existen dos caminos para implementar un PMS en el hotel. Hacerlo de forma local (in-house), es decir que toda la información se almacene en una base de datos ubicado en el establecimiento hotelero o hacerlo en modo externo (in-cloud) donde la base de datos se encuentra en un servidor de internet. Las diferencias y ventajas de cada modalidad se presentan en la tabla 4.

Tabla 4

Comparativo entre un modelo cloud y un modelo in House

	PMS in Cloud	PMS in House
Requerimientos del sistema	Dispositivo con un navegador web y una conexión a internet.	Estación de trabajo (PC), servidor de datos, servidor extra para backup, sistema operativo (generalmente Windows).
Instalación del sistema y acceso de usuarios	El software PMS y los datos son almacenados en un servidor compartido del centro de datos del proveedor. Los usuarios acceden en cualquier momento y lugar a través de un navegador web.	El software PMS y los datos son almacenados en un servidor local ubicado en las instalaciones del cliente. El sistema es instalado en cada PC que tendrá acceso al sistema.
Seguridad de datos y mantenimiento	La seguridad, el mantenimiento del software y los servidores son responsabilidad del proveedor incluyendo los backups automáticos.	La seguridad y mantenimiento del software y de los servidores son responsables del administrador del negocio.
Conocimientos de informática	No	Si
Presupuesto	Varios planes de suscripción, comúnmente de pago mensual (SAAS)	Normalmente un pago único por licencia y un plan de soporte anual para el software PMS.
Acceso Remoto	Si	Posible. Requiere hardware adicional y una configuración adicional de la red.
Actualizaciones gratuitas	Si	No
Ventajas	No requiere inversión en Hardware.	No depende de una conexión a internet.
Desventajas	No recomendable para hoteles sin una estable conexión a internet.	Costoso y complejo mantenimiento (hardware y actualizaciones de software)

Generalmente las empresas hoteleras de 4 y 5 estrellas prefieren un PMS In-house debido a que existe un staff amplio de colaboradores con responsabilidades asignadas que permanecen mayormente dentro del hotel. Por el contrario, para los administradores de hoteles pequeños la movilidad es un valor muy importante. Ellos se encargan de prácticamente todo en su negocio, por tanto, deben tener acceso al sistema desde cualquier parte y lugar sin invertir fuertes cantidades de dinero.

En la tabla 5 mostramos las marcas con mayor antigüedad en el mercado de los PMS.

Tabla 5

Relación de empresas proveedoras de PMS a nivel mundial

				Booking		Core PMS			Leisure		Advanced PMS			ERP Modules				
	On-Premise	Hybrid	Web-native	Channel Management	Online Booking	Front Desk	Revenue Optimisation	Housekeeping / Maintenance	Back Office	Reporting Analytics	Food & Beverage	Conferences	EPOS	CRM	Central reservation	Revenue yield management	Procurement HR & Payroll	Accounting & Finance
Oracle Hospitality	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Infor	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Protel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Itesso	✓		✓	✓	✓	✓	✓	✓	✓			✓	✓	✓				
Sabre			✓	✓	✓	✓	✓	✓	✓				✓	✓				
PAR Springer-Miller	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓				
Little Hotelier			✓	✓	✓			✓	✓									
Clock Software			✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓			
CMS Hospitality	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Genium	✓	✓		✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓
IDS Next	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
Agilysys	✓		✓		✓	✓	✓	✓	✓			✓	✓	✓			✓	
Eviivo			✓	✓	✓													
Hotelogix			✓	✓	✓	✓	✓	✓	✓	✓		✓		✓				

Fuente: Grant Thornton (2018)

Arquitectura de software.

De acuerdo con el Software Engineering Institute (2019), la Arquitectura de Software se refiere a “las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.” Se puede considerar como un diseño del sistema a desarrollar, pero al más alto nivel, sin detalles técnicos. En este diseño se aprecian la estructura, funcionamiento e interacción de los componentes del sistema.

Dentro de las arquitecturas más conocidas figuran la arquitectura cliente-servidor y la arquitectura de capas.

Arquitectura cliente-servidor.

La arquitectura cliente servidor se compone de dos partes. La primera parte es el servidor o equipo donde reside el software y la base de datos. El servidor debe ser un computador bastante potente para procesar todas las órdenes del cliente. La otra parte es el cliente o grupo de clientes que se conectan al servidor para consultar, procesar o registrar datos.

El servidor debe responder a las solicitudes hechas por los clientes o usuarios a través del protocolo HTTP o HTTPS.

Arquitectura de tres capas o niveles.

Según la empresa International Business Machines Corporation (IBM) la arquitectura en tres capas divide el sistema en tres partes diferenciadas, de tal forma que cada capa solo se comunique con la inferior. Esas tres capas se denominan:

Capa de Persistencia.

En esta capa se encuentra la base de datos. Aquí se procesan todas las tareas relacionados al registro, modificación o eliminación de datos. Se llama persistencia porque estos datos quedan grabados en el disco duro del servidor.

Capa de Negocio.

Es la capa intermedia y está encargada de procesar las órdenes o peticiones de los usuarios. Aquí se encuentran programadas todas las reglas y políticas del negocio, así como las políticas de seguridad. Por ejemplo, cuando un usuario intente realizar una venta, esta capa se encarga de consultar los descuentos a usar, los precios por temporadas, si el usuario tiene los permisos necesarios, las promociones vigentes, los datos obligatorios para registrar un cliente, etc.

Capa de Presentación.

Es la capa que tiene contacto con el usuario y está compuesta por toda la parte gráfica (botones, formularios y ventanas). Su única función es pasarle las acciones que realice el usuario a la capa de negocio.

Lenguaje de programación

Un lenguaje de programación es un lenguaje formal conformado por letras, símbolos y reglas gramaticales que permiten al programador escribir una secuencia de órdenes o algoritmos para controlar el aspecto físico y lógico de una computadora.

Los 10 lenguajes de programación más demandados a junio del 2015 según la página RedMonk ubicaban a PHP en el puesto 3. Este ranking fue obtenido por la popularidad del lenguaje en plataformas como GitHub y StackOverflow.

Tabla 6

Ranking de los lenguajes más demandados a junio del 2015

Junio 2015	Lenguaje	Ejecución
1	JavaScript	Cliente / Servidor
2	Java	Cliente
3	PHP	Cliente
4	Python	Cliente / Servidor
5	C#	Cliente / Servidor
6	C++	Cliente / Servidor
7	Ruby	Cliente / Servidor
8	CSS	Servidor
9	C	Servidor
10	Objective - C	Cliente

Fuente: redmonk.com

Framework de desarrollo.

Un framework es una estructura conceptual y tecnológica de soporte definido normalmente con artefactos o módulos de software concretos. Puede servir de base para la organización y desarrollo de software. Dentro de las principales ventajas de un framework figuran:

No redundancia en el código.

La mayoría de los proyectos tienen funciones básicas como la conexión a base de datos, validación de formularios o seguridad. Un framework contiene funciones programadas que ahorran muchas líneas de programación.

Buenas prácticas de desarrollo.

Los frameworks tienen una estructura definida para la programación. Por lo general usan el patrón Modelo-Vista-Controlador (MVC) que separa la programación en 3 capas. La capa de datos, la capa controladora (quien recibe las órdenes) y la de interfaz de usuario. Esto permite hacer más entendible y escalable la programación. Se corresponde a la arquitectura de 3 capas, pero también se podría tener una arquitectura de 3 capas sin necesidad de tener una programación MVC.

Desarrollo más rápido.

Debido a la reutilización de componentes y las funciones preprogramadas, la programación a través de un framework se realiza de manera rápida y fácil.

Cada lenguaje de programación tiene sus propios frameworks, en el caso de PHP el ranking de los 10 frameworks más demandados en el 2015 se muestra en la tabla 6

Tabla 7

Ranking de los frameworks más usados a mayo del 2015

Noviembre 2018	Nombre	Creación
1	Laravel	2011
2	Symfony	2005
3	Codeigniter	2006
4	Yii	2008
5	PHPPixie	2012
6	Zend	2005
7	CakePHP	2000
8	Phalcon	2012
9	Slim	2011
10	FuelPHP	2011

Fuente: sitepoint.com

Software as a Service (SAAS)

SAAS es la modalidad de adquisición de software más popular de la última década. Surge como consecuencia de la computación en la nube (cloud computing). Donde los usuarios no son propietarios del sistema informático que usan, sino que acceden a él a través de una conexión a internet. En contraprestación deben pagar una suscripción que generalmente es mensual.

El proveedor del sistema cloud es quien controla la seguridad, almacenamiento y la disponibilidad del sistema. Es más rentable adquirir un sistema cloud porque el usuario se ahorra gastos de hardware, mantenimiento y licencias de software.

Se puede decir que los sistemas cloud han tomado mucha fuerza y popularidad en América Latina en la última década. Como lo indica el estudio VMWare Cloud Adoption (2013) que el 45% de los presupuestos de las Tecnologías de Información y Comunicación (TICs) se asignaron a la computación en la nube en el 2012, superando incluso a las demás regiones del planeta.

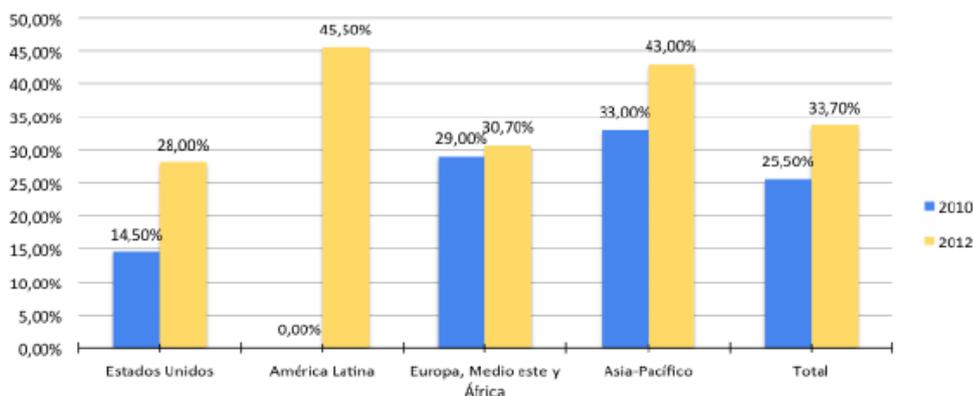


Figura 8. Presupuesto para cloud computing en el 2010 y 2012.

Fuente: VMWare Cloud Adoption Study

Estos resultados fueron confirmados por International Data Corporation (IDC) que indicó que las soluciones SAAS se estaban acercando al corazón del negocio en las empresas. Según sus propias cifras, el crecimiento de SAAS en Latinoamérica fue de 53% durante el 2013, y la perspectiva era que el 46% de las empresas invertirán hasta 25% de su presupuesto en Soluciones Cloud durante el 2014.

Infraestructura como Servicio (IAAS)

Las empresas que comercializan sistemas SAAS pueden alojar la aplicación en servidores propios (nube privada) o en servidores de terceros (nube pública). El último caso

ha ganado mucha popularidad porque existen empresas altamente calificadas que proveen servidores por costos bajos (IAAS). De ese modo las empresas desarrolladoras solo deben enfocarse en el diseño y desarrollo de soluciones.

IAAS es un modelo en el que un proveedor mantiene una infraestructura central, que incluye hardware, software, servidores y almacenamiento. Esto generalmente incluye el alojamiento de aplicaciones en un entorno altamente escalable, donde los clientes solo pagan por la infraestructura que utilizan.

De acuerdo con un estudio de Gartner, el mercado de IAAS al 2016 lo lideraban 2 empresas: Amazon y Microsoft. Estas empresas ponían a disposición de los usuarios recursos virtuales para que puedan publicar o desplegar sus aplicaciones a un precio bajo.



Figura 9. Cuadrante de Gartner (2016). Fuente: Gartner

Tabla 8

Comparación entre los proveedores de IAAS

	Amazon Web Services	Azure	Google Cloud Platform	Gtd
Propietario	Amazon	Microsoft	Google	Gtd
País	EEUU	EEUU	EEUU	CHILE
Lanzamiento	2006	2010	2011	1990
Data Centers en el mundo	66	54	20	8
Data Center Sudamérica	Brasil, Argentina	Brasil	Chile	Chile, Perú, Colombia
Market Share mundial	30%	16%	10%	-
Principales clientes	Netflix, Airbnb, Unilever, BMW, Samsung, etc.	Johnson, Fujifilm, HP, Honeywell, Apple, etc.	HSBC, Paypal, 20 th Century Fox, Bloomberg, etc.	-
Servicios Web	200+	100+	60+	0
Precio Instancia más pequeña	\$8/mes	\$15/mes	\$10/mes	\$20/mes

Fuente: Intellipaat y Amazon.com

Latencia de Red

La latencia es el tiempo que tarda en transmitirse un paquete en la red y es un factor clave en las conexiones a internet. Si se está vendiendo un SAAS, este punto es muy importante. Para medir la latencia se usa el comando “ping” que mide el tiempo de descarga de un paquete de datos desde un equipo remoto a una red local. Y se toma como referencia el valor promedio “average” expresado en milisegundos (ms). La latencia es afectada por tres factores: el ancho de banda, la distancia entre los puntos de comunicación y la capacidad del dispositivo emisor.

```

Cmder
λ ping www.google.com

Pinging www.google.com [2607:f8b0:4008:813::2004] with 32 bytes of data:
Reply from 2607:f8b0:4008:813::2004: time=91ms
Reply from 2607:f8b0:4008:813::2004: time=85ms
Reply from 2607:f8b0:4008:813::2004: time=80ms
Reply from 2607:f8b0:4008:813::2004: time=82ms

Ping statistics for 2607:f8b0:4008:813::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 80ms, Maximum = 91ms, Average = 84ms
    
```

Figura 10. Medición de la latencia de los servidores de Google descargando 32 bytes desde una red de Perú. Fuente: Elaboración propia

El ancho de banda o la tecnología de acceso a internet.

El ancho de banda es la medida de datos y recursos de comunicación disponible expresado en Megabits por segundo (Mbps).

Para llevar internet a un hogar o empresa se utilizan tres medios. Por un lado, se tiene la conexión ADSL con una velocidad de descarga de datos máxima de 20 Mbps. Del cual, según el Reglamento de Calidad de los Servicios Públicos de Telecomunicaciones en su artículo 6 establece que los proveedores de internet están obligados a cumplir solo el 40%.

Por otro lado, se tiene la conexión HFC o cable coaxial que puede llevar internet con una velocidad de descarga de hasta 100 Mbps como lo demuestra el estudio de Ookla (2016).

Por último, se tiene la conexión con fibra óptica que tiene una capacidad de transmisión de hasta 1,000 Mbps. (El Peruano, 2018).

A pesar de las diferencias en velocidades, en el 2016 gran parte del Perú aún contaba con conexión ADSL ofrecida por la empresa Telefónica tal como lo muestra la Tabla 9.

Tabla 9

Participación de mercado por operador en 2016

Operador	Propietario	Tecnología	% de mercado
Claro	América Móvil	HFC	18.1
Entel	Entel Chile	Inalámbrica	5.2
Telefónica	Telefónica	ADSL / HFC	74.9
Otros			1.8

Notas: Datos extraídos de Panorama de Mercado – Perú. Fuente: <https://www.telesemana.com/panorama-de-mercado/peru/>

En la tabla 10 se muestra el tiempo de espera del usuario con diferentes anchos de banda y diferentes tamaños archivos de descarga.

Tabla 10

Tiempo de descarga de datos usando diversos servicios de conectividad ADSL (Telefónica)

Ancho de Banda \ Tamaño de archivo	Plan 2 Mbps	Plan 4 Mbps	Plan 8 Mbps
	0.25 MBps al 100% 0.10 MBps al 40%	0.50 MBps al 100% 0.20 MBps al 40%	1.00 MBps al 100% 0.40 MBps al 40%
0.5 MB	5.00 s	2.50 s	1.25 s
0.7 MB	7.00 s	3.50 s	1.75 s
1 MB	10.00 s	5.00 s	2.50 s

Notas: 1 Megabyte (MB) equivale a 8 Megabits (Mb). Para calcular el tiempo de descarga se usó la fórmula MB/MBps. Fuente: Elaboración propia

La distancia entre los puntos de comunicación.

Otro factor que incrementa la latencia es la distancia. Cuanto mayor sea la distancia por recorrer entre los 2 puntos de emisión y recepción, mayor será la latencia. Por ejemplo, si estamos ubicados en Perú, la latencia de un servidor ubicado en Europa será mayor que un servidor ubicado en Norteamérica.

Capacidad del dispositivo desde el que nos conectamos.

La capacidad del dispositivo del que nos conectamos viene determinada por la capacidad de la tarjeta de red y de su nivel de procesamiento. La mayoría de las tarjetas pueden procesar hasta 1'000,000 de bits por segundo.

Diseño y desarrollo de software

Para poder construir software fue necesario tener una metodología de trabajo para que el equipo pueda alinearse. Si el alcance del proyecto es conocido y se prevé que no sufrirá cambios frecuentes durante el proceso, la metodología recomendable es una metodología pesada como cascada o RUP (Rational Unified Process).

La metodología en cascada (del inglés waterfall) descrita en la publicación *Gestionando el desarrollo de grandes sistemas de software* (Winston W. Royce 1970) es el modelo tradicional que ordena estrictamente las etapas del proceso para el desarrollo de software, donde el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Rubén Lazarte (s.f.) menciona que es conveniente usar una metodología de cascada cuando el equipo de trabajo tiene poca experiencia, cuando el cliente no tiene suficiente tiempo para el proyecto y cuando el alcance ya está definido desde el comienzo. Por estas razones se optó por la metodología Cascada.

Según el modelo cascada, las etapas en el desarrollo de software son cinco: Análisis del problema, diseño de la solución, implementación, pruebas y despliegue.

Análisis del problema.

En esta etapa se hace un diagnóstico de la problemática del cliente. El analista de negocio debe realizar una serie de entrevistas y reuniones con el cliente y los usuarios del sistema a fin de entender sus necesidades. Es en este proceso que se obtienen los requisitos funcionales y no funcionales del proyecto.

Diseño de la solución.

En esta etapa se reúne el analista con el equipo de desarrollo a fin de plantear una solución a la problemática del paso anterior. De esta reunión se obtiene la propuesta técnica de solución que será expuesta al cliente. Esta propuesta debe contener lo siguiente:

- Requerimientos funcionales y no funcionales.
- Arquitectura del sistema.
- Diseño de la base datos.
- Calendario de actividades.

Al exponer la propuesta técnica al cliente pueden surgir más ideas y funcionalidades que no fueron comentadas al inicio. Si fuera el caso, esta etapa debe reiniciarse las veces que sean necesarias hasta que el cliente este conforme con el alcance del proyecto.

Implementación.

En este proceso se programan los módulos identificados en el paso anterior para convertir la propuesta de solución en un sistema informático.

Aquí se programa el frontend que es la parte que interactúa con los usuarios y el backend que es la parte que se conecta con la base de datos y el servidor. Se hace uso del framework elegido para poder tener una mejor gestión de la programación.

Pruebas.

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funcione correctamente. Se buscan sistemáticamente y se corrigen todos los errores antes de ser entregado al usuario final.

Despliegue.

Cuando se tienen los errores levantados y subsanados se inicia el pase a producción del software. Se instala el software en el servidor de producción y queda lista para ser utilizada por el usuario final.

Desarrollo del proyecto

De acuerdo con el análisis de Ishikawa y del árbol de problemas se pudo concluir que los problemas obedecían a un problema central: La manera de gestionar los recursos por parte del administrador estaba siendo ineficiente. La solución partía por capacitación intensiva y contar con una buena herramienta de gestión.

De acuerdo con el alcance del proyecto no se consideró la capacitación en temas de gestión. Por tanto, la solución partió por construir un sistema informático que ayude a mejorar su gestión.

Para la construcción del software se usó la metodología Cascada y se dividió el proyecto en 5 etapas: Análisis del problema, diseño de solución, implementación, pruebas y despliegue.

Análisis del problema

El objetivo de esta etapa fue obtener la lista de requerimientos que debería satisfacer el sistema. Por ello fue necesario dividir el problema en subproblemas:

Tiempos excesivos.

Los procesos principales tenían un alto consumo de mano de obra. En el caso de las reservas el tiempo promedio empleado era de 9 minutos, en el cierre de caja se podrían tardar hasta 34 minutos, ya que al tener registros en papel estos se extraviaban o no sumaban correctamente y en el control de inventarios tardaban hasta 25 minutos en promedio.

Tabla 11

Tiempo empleado en el proceso de reserva (expresado en minutos)

SUNSET	ALBAREDA	QUEEN	WASHI- NGTON	NORTE	VICTORIA	MOLINA	MISTI	FRONTERA	GOCTA
11	9	10	8	8	11	10	9	8	6

Fuente: Elaboración propia

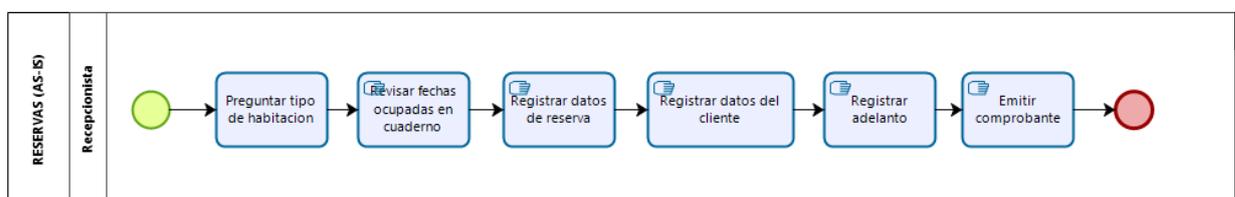


Figura 11. Proceso de reservas antes de implementarse el sistema. Elaboración propia

Tabla 12

Tiempo empleado en el proceso de cierre de caja (expresado en minutos)

SUNSET	ALBAREDA	QUEEN	WASHINGTON	NORTE	VICTORIA	MOLINA	MISTI	FRONTERA	GOCTA
23	35	26	18	20	30	40	16	20	19

Fuente: Elaboración propia

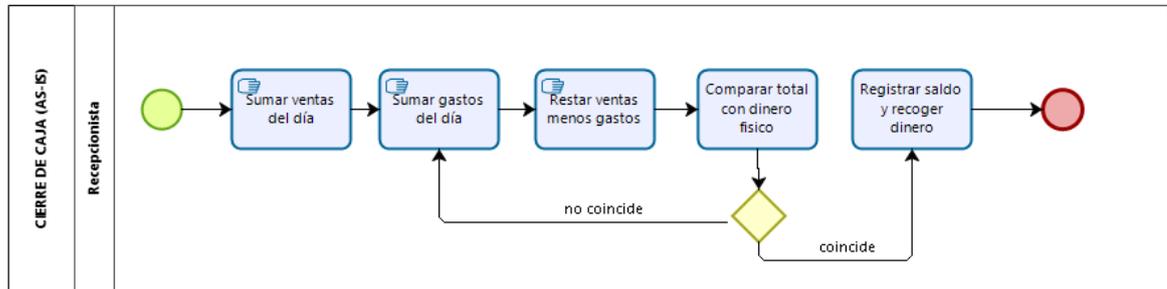


Figura 12. Proceso de cierre de caja. Elaboración propia

Tabla 13

Tiempo empleado en el proceso de control de inventarios (expresado en minutos)

SUNSET	ALBAREDA	QUEEN	WASHINGTON	NORTE	VICTORIA	MOLINA	MISTI	FRONTERA	GOCTA
30	25	35	41	32	30	45	34	31	40

Fuente: Elaboración propia

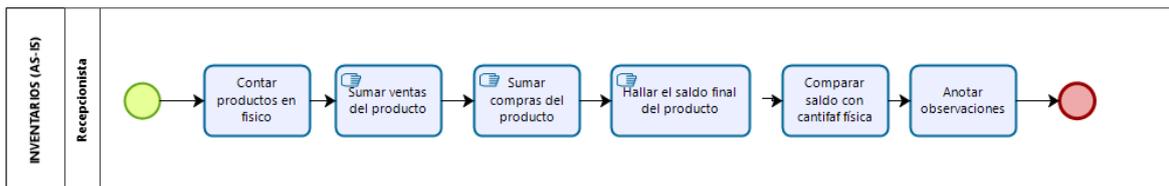


Figura 13. Proceso de gestión de inventarios. Elaboración propia

Alta rotación del personal.

De acuerdo con las entrevistas realizadas a los administradores, la tasa de rotación promedio fue de 62%. La fórmula empleada para el cálculo fue la siguiente:

$$TR = \frac{N^{\circ} \text{ de trabajadores renunciando}}{N^{\circ} \text{ promedio de empleados}} \times 100$$

Tabla 14

Nivel de rotación del personal en los establecimientos hoteleros

	Contrataciones	N° Empleados	TR
SUNSET	7	6	116%
ALBAREDA	8	7	114%
QUEEN	7	7	100%
WASHINGTON	5	10	50%
NORTE	2	8	25%
VICTORIA	1	10	10%
MOLINA	3	7	43%
MISTI	3	7	43%
FRONTERA	3	6	50%
GOCTA	4	6	67%

Fuente: Elaboración propia

Carencia de Indicadores de gestión.

Ningún establecimiento que formó parte del estudio tenía indicadores de gestión actualizados. Entre las razones que se encontraron fue la cantidad de datos difíciles de procesar por el administrador y la información dispersa en varios cuadernos de control.

Diseño de software

Con la definición de los problemas se concluyó que era necesario la construcción de un sistema informático de gestión. En el caso de la hotelería lo más recomendable era diseñar y desarrollar un PMS para cubrirlos siguientes procesos:

- Clientes: Para la gestión de los perfiles y el historial de compras de clientes.
- Recepción: Para la gestión de reservas, checkin, checkout y caja.
- Backoffice: Para la gestión de gastos e inventarios de productos.
- Reportes y analítica: Para la generación de reportes e indicadores.

Documento de requerimientos.

Este documento señaló el alcance del proyecto. Solo se desarrollaría lo que se encuentre dentro del documento de requerimientos. Estos documentos a su vez se dividieron en requerimientos funcionales que definen lo que el sistema tiene que hacer. Y requerimientos no funcionales que definen cómo se va hacer. Un punto de partida fue realizar casos de uso del negocio para conocer a los stakeholders y sus principales actividades.

Diagramas de caso de uso del negocio.

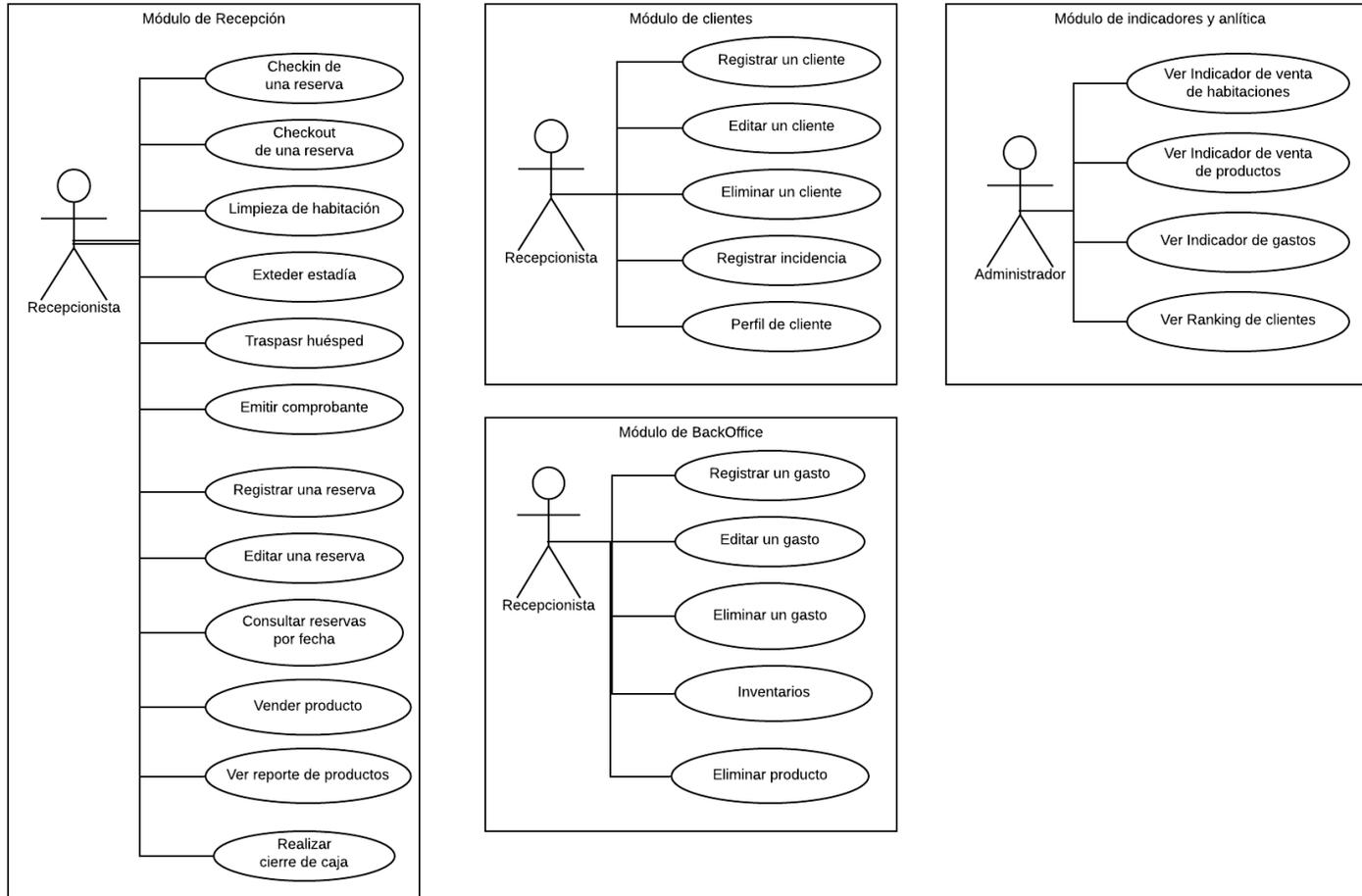


Figura 14. Casos de uso del sistema. Fuente: Elaboración propia

Requerimientos funcionales.

Los requerimientos funcionales fueron todas aquellas acciones que debería hacer o permitir hacer el sistema. La forma de redactar un requerimiento es indicando el mayor detalle para que pueda ser programado. Cada requerimiento tuvo un código único.

Tabla 15

Lista de requerimientos funcionales

Código	RF01
Nombre	Acceso al sistema
Descripción	Acceder de forma segura al sistema
Entradas	Nombre del negocio, usuario y password
Salidas	Sesión creada

Código	RF02
Nombre	Checkin de una reserva
Descripción	Se registra el ingreso de un huésped a una habitación
Entradas	Fecha de reserva, tipo de habitación, documento y nombres del huésped y pagos realizados
Salidas	Registro de reserva actualizado y estado de habitación actualizado.

Código	RF03
Nombre	Checkout de una reserva
Descripción	Se registra la salida de un huésped de una habitación
Entradas	Número de habitación
Salidas	Registro de reserva actualizado y estado de habitación actualizado

Código	RF04
Nombre	Limpieza de habitación
Descripción	Se procede a dar inicio a la limpieza de la habitación
Entradas	Número de habitación y nombre del trabajador de limpieza
Salidas	Estado de habitación actualizado

Código	RF05
Nombre	Extender estadía
Descripción	Ampliar las fechas de reserva de un huésped
Entradas	Código de reserva y nueva fecha de salida
Salidas	Registro de reserva actualizado

Código	RF06
Nombre	Traspasar a un huésped
Descripción	Cambiar de habitación a un huésped
Entradas	Número de habitación actual y número de habitación a traspasar
Salidas	Registro de reserva actualizado

Código	RF07
Nombre	Emitir la liquidación de un huésped
Descripción	Calcular el total vendido de un cliente y proceder al cobro.
Entradas	Número de habitación actual y monto pagado
Salidas	Registro de reserva actualizado

Código	RF08
Nombre	Registrar una reserva
Descripción	Crear una reserva
Entradas	Número de habitación, fecha de ingreso, fecha de salida, documento y nombres del huésped.
Salidas	Registro de reserva creado

Código	RF09
Nombre	Editar una reserva
Descripción	Cambiar los datos de una reserva
Entradas	Código, número de habitación, fecha de ingreso, fecha de salida, documento y nombres del huésped.
Salidas	Registro de reserva actualizado

Código	RF10
Nombre	Consultar reservas por fecha
Descripción	Buscar las reservas registradas en una fecha en particular
Entradas	Fecha de ingreso
Salidas	Lista de reservas

Código	RF11
Nombre	Vender un producto a un huésped
Descripción	Buscar un producto en el inventario y cargarlo a la cuenta del huésped.
Entradas	Número de habitación, código de producto, cantidad y medio de pago
Salidas	Registro de venta creado

Código	RF12
--------	------

Nombre	Vender un producto a un trabajador
Descripción	Buscar un producto en el inventario y cargarlo a la cuenta del trabajador
Entradas	Número de habitación, código de producto, cantidad y medio de pago
Salidas	Registro de venta creado

Código	RF13
Nombre	Consultar stock de producto
Descripción	Buscar si un producto tiene stock disponible en el inventario
Entradas	Código de producto y cantidad
Salidas	Si o No

Código	RF14
Nombre	Sumar pagos realizados
Descripción	Sumar los pagos realizados en el turno actual del recepcionista y clasificarlos por medio de pago: Efectivo, tarjeta o transferencia
Entradas	Fecha del último cierre
Salidas	Total de pagos realizados

Código	RF15
Nombre	Sumar venta de habitaciones
Descripción	Sumar la venta de habitaciones del turno actual y clasificarlos por tipo de habitación.
Entradas	Fecha del último cierre
Salidas	Total de venta de habitaciones

Código	RF16
Nombre	Sumar gastos
Descripción	Sumar los gastos registrados en el turno actual.
Entradas	Fecha del último cierre
Salidas	Total de gastos

Código	RF17
Nombre	Cierre de caja
Descripción	Calcular la cantidad de efectivo disponible en caja y registrar el saldo de inicio para el siguiente turno. Para finalizar debe enviarse un correo al administrador.
Entradas	Fecha del último cierre
Salidas	Registro de cierre creado, correo electrónico.

Código	RF18
Nombre	Registrar un cliente
Descripción	Crear un cliente en la base de datos
Entradas	Documento, nombres, apellidos, dirección y correo.
Salidas	Registro de cliente creado

Código	RF19
Nombre	Editar un cliente
Descripción	Cambiar los datos de un cliente
Entradas	Código, documento, nombres, apellidos, dirección y correo.
Salidas	Registro de cliente actualizado

Código	RF20
Nombre	Eliminar un cliente
Descripción	Eliminar un cliente de la base de datos
Entradas	Código y documento del cliente
Salidas	Registro de cliente actualizado

Código	RF21
Nombre	Registrar incidencia de clientes
Descripción	Agregar una incidencia positiva o negativa a un cliente determinado
Entradas	Documento del cliente y descripción de la incidencia
Salidas	Registro de incidencia creado

Código	RF22
Nombre	Perfil del cliente
Descripción	Buscar información de un cliente
Entradas	Documento del cliente
Salidas	Información de un cliente

Código	RF23
Nombre	Registrar gasto
Descripción	Registro de un gasto en la base de datos
Entradas	Descripción, categoría y precio
Salidas	Registro de gasto creado

Código	RF24
Nombre	Editar gasto

Descripción	Cambiar los datos de un gasto registrado
Entradas	Código, descripción, categoría y precio
Salidas	Registro de gasto actualizado

Código	RF25
Nombre	Eliminar gasto
Descripción	Eliminar un gasto de la base de datos
Entradas	Código de gastos
Salidas	Estado de gasto actualizado

Código	RF26
Nombre	Registrar producto
Descripción	Crear un producto en el inventario
Entradas	Nombre, categoría, precio y stock
Salidas	Registro de producto creado

Código	RF27
Nombre	Actualizar producto
Descripción	Cambiar los datos de un producto
Entradas	Código, Nombre, categoría, precio y stock
Salidas	Registro de producto actualizado

Código	RF28
Nombre	Eliminar producto
Descripción	Quitar un producto del inventario
Entradas	Código, Nombre, categoría, precio y stock
Salidas	Registro de producto actualizado

Código	RF29
Nombre	Reponer producto
Descripción	Registrar la compra de productos y actualizar el stock
Entradas	Código, cantidad y precio
Salidas	Inventario de producto actualizado

Código	RF30
Nombre	Cardex de un producto
Descripción	Generar el reporte de un producto entre fechas donde se vea la entrada y salida de productos.
Entradas	Código o nombre de producto

Salidas	Cardex de producto
---------	--------------------

Código	RF31
Nombre	Indicador de venta de habitaciones
Descripción	Mostrar un gráfico con las ventas de habitaciones por mes clasificado por tipo de habitación
Entradas	Número de mes y año
Salidas	Gráfico de venta de habitaciones

Código	RF32
Nombre	Indicador de venta de productos
Descripción	Mostrar un gráfico con las ventas de productos por mes clasificado por categoría de producto
Entradas	Número de mes y año
Salidas	Gráfico de venta de productos

Código	RF33
Nombre	Ranking de clientes
Descripción	Mostrar un gráfico comparando el monto de venta por cliente
Entradas	Monto de ventas, mes y año
Salidas	Gráfico comparativo de ventas por cliente

Fuente: Elaboración propia

Requerimientos No funcionales.

Es la manera de cómo el sistema va cumplir con los requerimientos funcionales Por ejemplo, capacidad del servidor, capacidad del hosting, software usado, etc. Para comprender mejor cada requerimiento se elaboraron tablas de detalle por cada uno.

Tabla 16

Lista de requerimientos No funcionales

Código	RNF01
Nombre	Responsive Design
Descripción	El sistema debe verse correctamente en pantallas de 7 pulgadas en adelante. Debe verse en celulares, tablets, laptops y desktops.

Código	RNF02
Nombre	Cloud computing
Descripción	El sistema debe verse desde cualquier dispositivo conectado a internet.

Código	RNF03
Nombre	Programación Front-End
Descripción	La tecnología requerida es: Jquery 1.10 para el manejo de Javascript HighCharts para la creación de indicadores Bootstrap 3 para el diseño responsive FontAwesome 3 para los iconos

Código	RNF04
Nombre	Programación Back End
Descripción	La tecnología requerida es: PHP 7.0 PHPMailer para el envío de correo PHPExcel para exportar los reportes a Excel QRCode para generar códigos QR Codeigniter 3

Código	RNF05
Nombre	Servidor de base de datos
Descripción	En este servidor estará disponible la base de datos y se conectará con el servidor de aplicaciones. Software Sistema Operativo Amazon Linux 2016.03 Base de datos MySQL Server 15.1 Hardware Instancia Amazon t2.micro Procesador INTEL XEON 3.3 GHz Disco Duro 16GB Memoria RAM 1GB Tarjeta de red virtual 1 Gbps

Código	RNF06
Nombre	Servidor de aplicaciones
Descripción	En este servidor estará el código del proyecto. Para la obtención de datos se conectará a la base de datos. Software Amazon Linux 2016.03 NGINX 1.12 PHP 7 COMPOSER 1.7.2 CODEIGNITER 3.0 GIT 2.14 Hardware Instancia Amazon t2.micro Procesador INTEL XEON 3.3 GHz Disco Duro 8GB

	Memoria RAM 1GB Tarjeta de red virtual 1 Gbps
Código	RNF07
Nombre	Computador cliente
Descripción	Para tener una buena experiencia de xafiro considerar la siguiente tecnología. Software Sistema operativo Windows 7+ o Linux Google Chrome 30.0, Mozilla Firefox 27.0, Opera 10.0 o Safari 7.0 Hardware Pantalla de 14+ pulgadas Mouse y Teclado Memoria RAM 4GB+ Disco duro de 8GB+ Tarjeta de red de 1Gbps Ancho de banda de 4Mbps

Fuente: Elaboración propia

Arquitectura del PMS

La arquitectura que se necesitó para el proyecto fue la de cliente-servidor de 3 capas. Se requirieron 2 instancias A y B. Una para la aplicación y la otra para la base de datos.

Estas instancias generarían reportes de rendimiento a través del servicio ofrecido por Amazon Web Services llamado Amazon CloudWatch. Asimismo, se realizarían los backups automáticos para almacenarlos en contenedores o “buckets” llamados S3.

Los usuarios de los hoteles se podían conectar a través del protocolo seguro HTTPS mientras que los administradores de Xafiro se podían conectar a través del protocolo SSH. Cada hotel tendría su propia base datos para asegurar la independencia de datos.

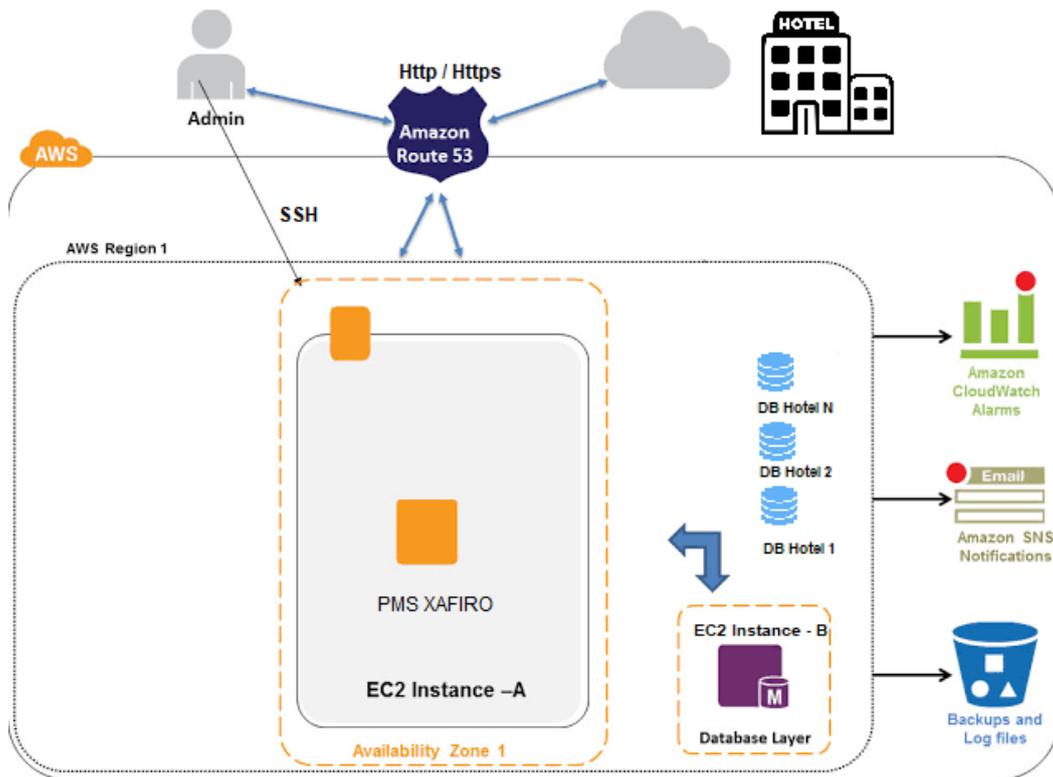


Figura 15. Arquitectura de 3 capas usando AWS. Fuente: Elaboración propia

Diseño de la base de datos

Modelo conceptual.

El modelo conceptual del PMS es un modelo de alto nivel que permite ver el alcance global de todo el sistema.

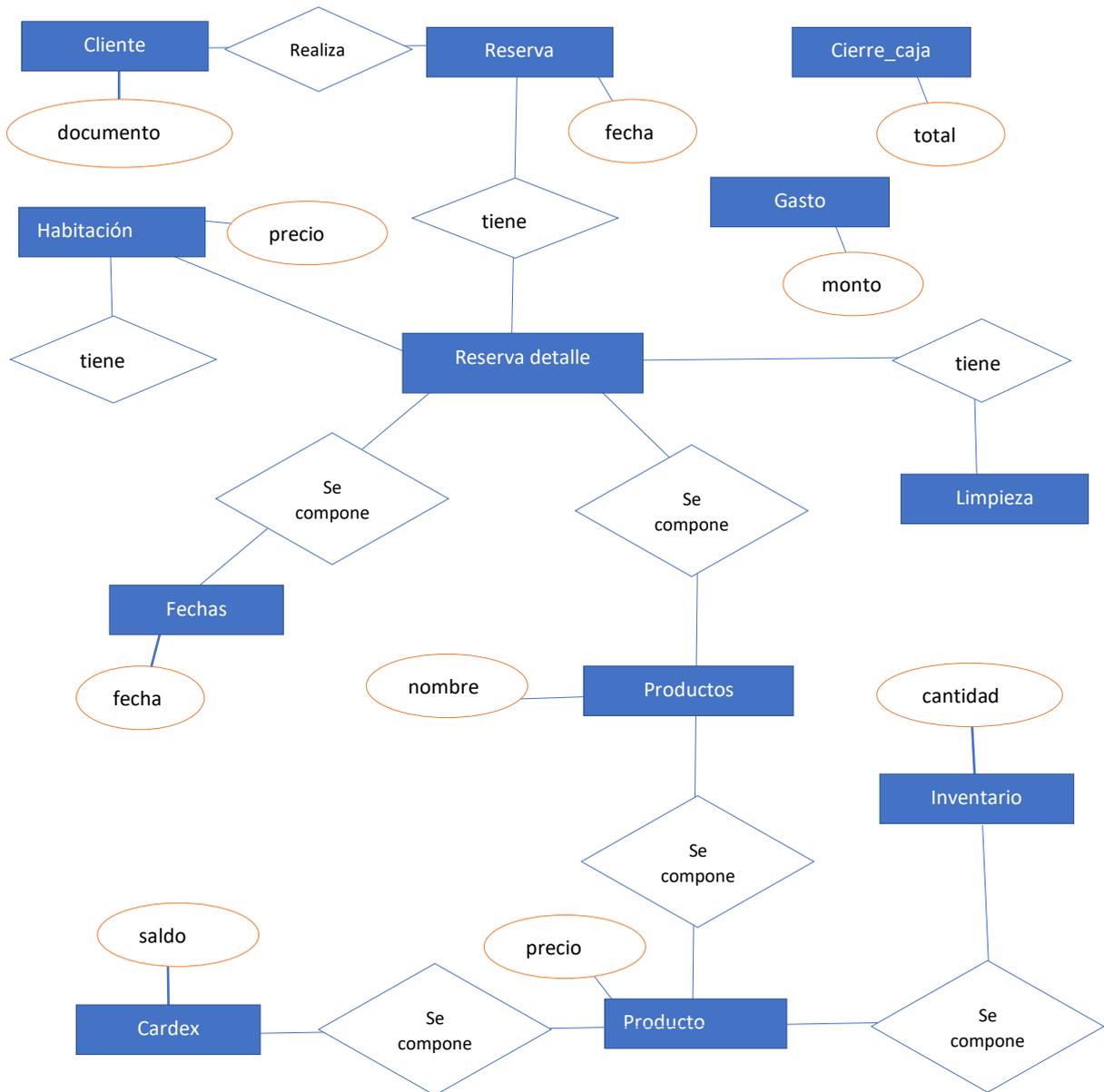


Figura 16. Diseño conceptual de la base de datos del PMS

Modelo lógico.

El modelo lógico incluyó la notación UML para representar las relaciones entre las entidades del sistema. Además, en esta etapa se redujeron las redundancias en el diseño de base de datos aplicando 3 formas normales. 1era Forma Normal, 2da Forma Normal y 3era Forma Normal. De todo este proceso resultaron un total de 15 entidades.

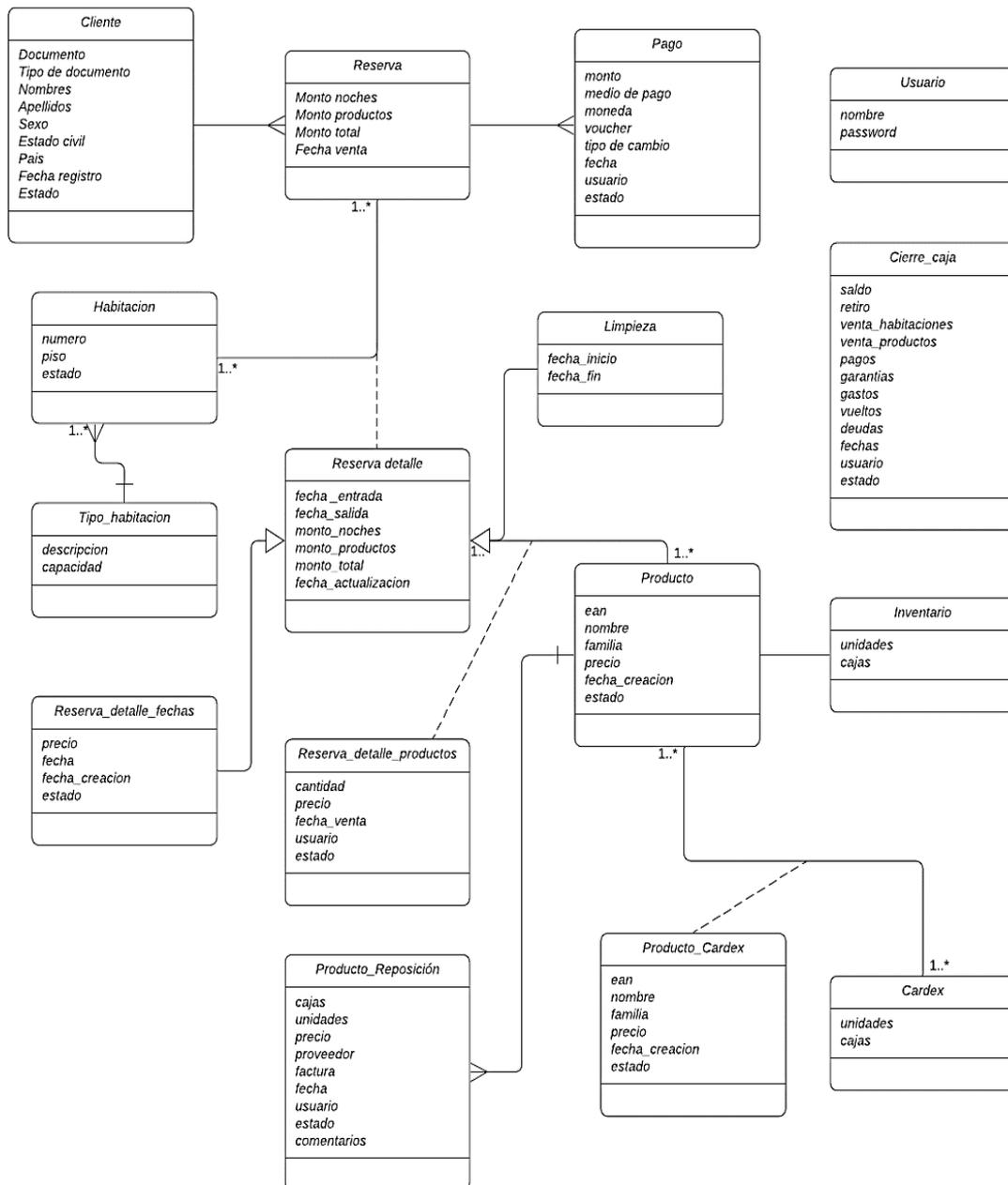


Figura 17. Diseño lógico de la base de datos del PMS

Modelo físico.

Para el modelo físico se indicó el tipo de dato que tendría cada atributo de las entidades mencionadas. El resultado se muestra a continuación

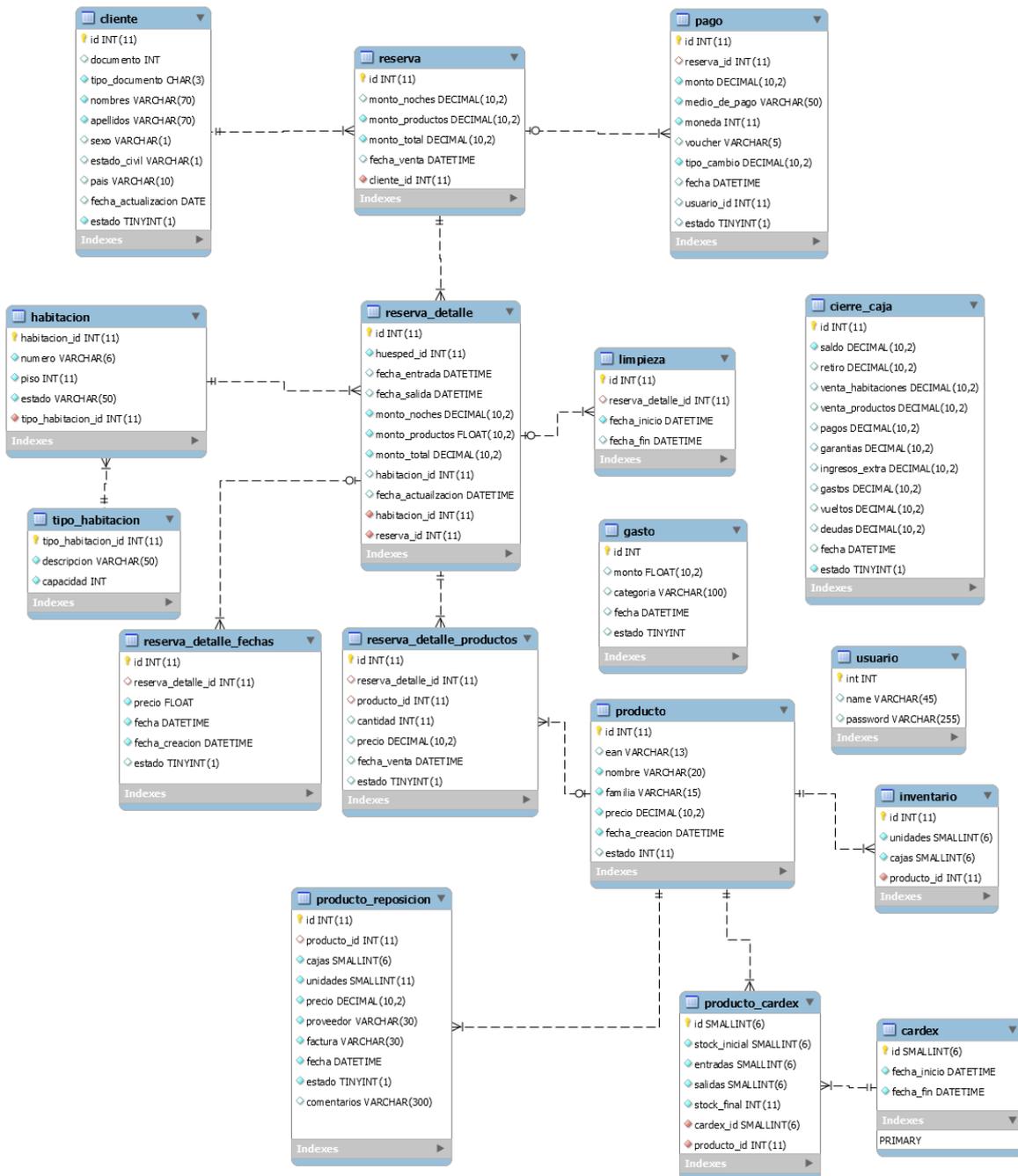


Figura 18. Diseño físico de la base de datos del PMS

Una vez definido el diseño de la base de datos se procedió a generar el script SQL correspondiente. Para ello se utilizó el software MySQL Workbench.

```
CREATE TABLE IF NOT EXISTS `usuario` (
```

```
  `int` INT NOT NULL,  
  `name` VARCHAR(45) NULL,  
  `password` VARCHAR(255) NULL,  
  PRIMARY KEY (`int`));
```

```
CREATE TABLE IF NOT EXISTS `cliente` (
```

```
  `id` INT(11) NOT NULL,  
  `documento` INT NULL DEFAULT NULL,  
  `tipo_documento` CHAR(3) CHARACTER SET 'utf8' NOT NULL DEFAULT 'DNI',  
  `nombres` VARCHAR(70) CHARACTER SET 'utf8' NOT NULL,  
  `apellidos` VARCHAR(70) CHARACTER SET 'utf8' NOT NULL,  
  `sexo` VARCHAR(1) CHARACTER SET 'utf8' NULL DEFAULT NULL,  
  `estado_civil` VARCHAR(1) CHARACTER SET 'utf8' NULL DEFAULT NULL,  
  `pais` VARCHAR(10) CHARACTER SET 'utf8' NULL DEFAULT NULL,  
  `fecha_actualizacion` DATE NULL DEFAULT NULL,  
  `estado` TINYINT(1) NOT NULL DEFAULT '1',  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `documento` (`documento` ASC));
```

```
CREATE TABLE IF NOT EXISTS `tipo_habitacion` (
```

```
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `descripcion` VARCHAR(50) CHARACTER SET 'utf8' NOT NULL,  
  `capacidad` INT NOT NULL,  
  PRIMARY KEY (`id`));
```

```
CREATE TABLE IF NOT EXISTS `habitacion` (
```

```
  `habitacion_id` INT(11) NOT NULL AUTO_INCREMENT,  
  `numero` VARCHAR(6) CHARACTER SET 'utf8' NOT NULL,  
  `piso` INT(11) NOT NULL,  
  `estado` VARCHAR(50) CHARACTER SET 'utf8' NOT NULL DEFAULT 'disponible',  
  `tipo_habitacion_id` INT(11) NOT NULL,  
  PRIMARY KEY (`habitacion_id`),  
  CONSTRAINT `fk_habitacion_tipo_habitacion1`  
  FOREIGN KEY (`tipo_habitacion_id`) REFERENCES `tipo_habitacion` (`id`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `reserva` (
```

```
  `id` INT(11) NOT NULL,  
  `monto_noches` DECIMAL(10,2) NULL DEFAULT '0.00',  
  `monto_productos` DECIMAL(10,2) NOT NULL DEFAULT '0.00',  
  `monto_total` DECIMAL(10,2) NOT NULL DEFAULT '0.00',  
  `fecha_venta` DATETIME NULL DEFAULT NULL,  
  `cliente_id` INT(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `fk_libro_habitaciones_cliente1`  
  FOREIGN KEY (`cliente_id`) REFERENCES `cliente` (`id`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `reserva_detalle` (
```

```
  `id` INT(11) NOT NULL,  
  `huesped_id` INT(11) NOT NULL,  
  `fecha_entrada` DATETIME NULL DEFAULT NULL,  
  `fecha_salida` DATETIME NULL DEFAULT NULL,  
  `monto_noches` DECIMAL(10,2) NOT NULL DEFAULT '0.00',  
  `monto_productos` FLOAT(10,2) NOT NULL DEFAULT '0.00',  
  `monto_total` DECIMAL(10,2) NOT NULL DEFAULT '0.00',  
  `habitacion_id` INT(11) NULL DEFAULT NULL,  
  `fecha_actuailzacion` DATETIME NULL DEFAULT NULL,  
  `habitacion_id` INT(11) NOT NULL,  
  `reserva_id` INT(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `fk_reserva_detalle_habitacion1`  
  FOREIGN KEY (`habitacion_id`)  
  REFERENCES `habitacion` (`habitacion_id`)  
  ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT `fk_reserva_detalle_reserva1`  
FOREIGN KEY (`reserva_id`) REFERENCES `reserva` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `producto` (  
`id` INT(11) NOT NULL,  
`ean` VARCHAR(13) CHARACTER SET 'utf8' NULL DEFAULT NULL,  
`nombre` VARCHAR(20) CHARACTER SET 'utf8' NOT NULL,  
`familia` VARCHAR(15) CHARACTER SET 'utf8' NOT NULL,  
`precio` DECIMAL(10,2) NOT NULL DEFAULT '0.00',  
`fecha_creacion` DATETIME NOT NULL,  
`estado` INT(11) NULL DEFAULT NULL,  
PRIMARY KEY (`id`));
```

```
CREATE TABLE IF NOT EXISTS `reserva_detalle_productos` (  
`id` INT(11) NOT NULL,  
`reserva_detalle_id` INT(11) NULL DEFAULT NULL,  
`producto_id` INT(11) NULL DEFAULT NULL,  
`cantidad` INT(11) NULL DEFAULT NULL,  
`precio` DECIMAL(10,2) NULL DEFAULT NULL,  
`fecha_venta` DATETIME NULL DEFAULT NULL,  
`estado` TINYINT(1) NULL DEFAULT NULL,  
PRIMARY KEY USING BTREE (`id`),  
CONSTRAINT `detalle_lh_prod_ibfk_1`  
FOREIGN KEY (`producto_id`) REFERENCES `producto` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION,  
CONSTRAINT `fk_detalle_lh_prod_detalle_lh_hab1`  
FOREIGN KEY (`reserva_detalle_id`) REFERENCES `reserva_detalle` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `reserva_detalle_fechas` (  
`id` INT(11) NOT NULL,  
`reserva_detalle_id` INT(11) NULL DEFAULT NULL,  
`precio` FLOAT NOT NULL,  
`fecha` DATETIME NOT NULL,  
`fecha_creacion` DATETIME NOT NULL,  
`estado` TINYINT(1) NULL DEFAULT NULL,  
PRIMARY KEY (`id`),  
CONSTRAINT `fk_detalle_noches_detalle_lh1`  
FOREIGN KEY (`reserva_detalle_id`) REFERENCES `reserva_detalle` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `limpieza` (  
`id` INT(11) NOT NULL,  
`reserva_detalle_id` INT(11) NULL DEFAULT NULL,  
`fecha_inicio` DATETIME NOT NULL,  
`fecha_fin` DATETIME NULL DEFAULT NULL,  
PRIMARY KEY (`id`),  
CONSTRAINT `fk_hab_limpiadas_detalle_lh1`  
FOREIGN KEY (`reserva_detalle_id`) REFERENCES `reserva_detalle` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `inventario` (  
`id` INT(11) NOT NULL AUTO_INCREMENT,  
`unidades` SMALLINT(6) NOT NULL DEFAULT '0',  
`cajas` SMALLINT(6) NOT NULL DEFAULT '0',  
`producto_id` INT(11) NOT NULL,  
PRIMARY KEY (`id`),  
CONSTRAINT `fk_inventario_producto1`  
FOREIGN KEY (`producto_id`) REFERENCES `producto` (`id`)  
ON DELETE NO ACTION ON UPDATE NO ACTION));
```

```
CREATE TABLE IF NOT EXISTS `cierre_caja` (  
`id` INT(11) NOT NULL,  
`saldo` DECIMAL(10,2) NOT NULL,  
`retiro` DECIMAL(10,2) NULL DEFAULT NULL,
```

```

`venta_habitaciones` DECIMAL(10,2) NULL DEFAULT NULL,
`venta_productos` DECIMAL(10,2) NULL DEFAULT NULL,
`pagos` DECIMAL(10,2) NULL DEFAULT NULL,
`garantias` DECIMAL(10,2) NULL DEFAULT NULL,
`ingresos_extra` DECIMAL(10,2) NULL DEFAULT NULL,
`gastos` DECIMAL(10,2) NULL DEFAULT NULL,
`vueltos` DECIMAL(10,2) NULL DEFAULT NULL,
`deudas` DECIMAL(10,2) NULL DEFAULT NULL,
`fecha` DATETIME NULL DEFAULT NULL,
`estado` TINYINT(1) NOT NULL DEFAULT '1',
PRIMARY KEY (`id`));

CREATE TABLE IF NOT EXISTS `pago` (
  `id` INT(11) NOT NULL,
  `reserva_id` INT(11) NULL DEFAULT NULL,
  `monto` DECIMAL(10,2) NOT NULL DEFAULT '0.00',
  `medio_de_pago` VARCHAR(50) CHARACTER SET 'utf8' NOT NULL,
  `moneda` INT(11) NOT NULL DEFAULT '1',
  `voucher` VARCHAR(5) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `tipo_cambio` DECIMAL(10,2) NOT NULL DEFAULT '1.00',
  `fecha` DATETIME NULL DEFAULT NULL,
  `estado` TINYINT(1) NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_pago_libro_habitaciones1`
  FOREIGN KEY (`reserva_id`) REFERENCES `reserva` (`id`)
  ON DELETE NO ACTION ON UPDATE NO ACTION);

CREATE TABLE IF NOT EXISTS `gasto` (
  `id` INT(11) NOT NULL,
  `monto` DECIMAL(10,2) NOT NULL DEFAULT '0.00',
  `categoria` VARCHAR(100),
  `fecha` DATETIME NULL DEFAULT NULL,
  `estado` TINYINT(1) NULL DEFAULT NULL,
  PRIMARY KEY (`id`));

CREATE TABLE IF NOT EXISTS `cardex` (
  `id` SMALLINT(6) NOT NULL AUTO_INCREMENT,
  `fecha_inicio` DATETIME NOT NULL,
  `fecha_fin` DATETIME NOT NULL,
  PRIMARY KEY (`id`));

CREATE TABLE IF NOT EXISTS `producto_cardex` (
  `id` SMALLINT(6) NOT NULL AUTO_INCREMENT,
  `stock_inicial` SMALLINT(6) NOT NULL,
  `entradas` SMALLINT(6) NOT NULL,
  `salidas` SMALLINT(6) NOT NULL,
  `stock_final` INT(11) NOT NULL,
  `cardex_id` SMALLINT(6) NOT NULL,
  `producto_id` INT(11) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_producto_cardex_detalle_producto_cardex1`
  FOREIGN KEY (`cardex_id`) REFERENCES `cardex` (`id`)
  ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_producto_cardex_detalle_producto1`
  FOREIGN KEY (`producto_id`) REFERENCES `producto` (`id`)
  ON DELETE NO ACTION ON UPDATE NO ACTION);

```

Figura 19. Script escrito en SQL para crear la base de datos

Calendario de actividades.

El proyecto inició en noviembre del 2014 y finalizó en abril del 2015. Tuvo una duración de 6 meses.

Tabla 17

Descripción y duración de tareas del proyecto

		Analista	Programador 1	Programador 2
ANALISIS DEL PROBLEMA				
Kick off meeting	40	✓	✓	
Levantamiento de información	40	✓	✓	
DISEÑO DE LA SOLUCION				
Elaboración de propuesta de solución				
Definición de requerimientos funcionales	40	✓		
Definición de requerimientos no funcionales	32	✓		
Definición de arquitectura del sistema	8	✓		
Definición de casos de prueba	32	✓		
Revisión de propuesta con cada administrador	20	✓		
Levantamiento de observaciones	16	✓		
Aprobación de administradores	10	✓		
Diseño de la base de datos	80	✓	✓	✓
IMPLEMENTACION DE LA SOLUCION				
Programación del módulo de Acceso				
Login	16			✓
Programación del módulo Recepción				
Crear una reserva	24		✓	✓
Editar una reserva	8			✓
Eliminar una reserva	8			✓
Registrar checkin	16		✓	✓
Registrar venta de producto	32		✓	✓
Registrar checkout	16		✓	✓
Registrar limpieza de habitación	16		✓	✓
Registrar extensión de estadia	16		✓	✓
Registrar traspaso de huésped	16		✓	✓
Calcular liquidación del huésped	32		✓	✓
Crear reporte de reservas	16		✓	✓
Ver resumen pagos realizados en el turno	8		✓	
Ver resumen de venta de habitaciones en el turno	8		✓	
Ver resumen de gastos en el turno	8		✓	
Registrar Cierre de caja	24		✓	✓
Programación del módulo Cliente				
Registrar un cliente	8		✓	
Editar un cliente	8		✓	
Eliminar un cliente	8		✓	
Registrar incidencias de clientes	8		✓	
Buscar un cliente por documento	8		✓	
Programación del módulo Back Office				
Registrar un gasto	8			✓
Editar un gasto	8			✓
Eliminar un gasto	8			✓

Registrar un producto	8	✓	
Editar un producto	8	✓	
Eliminar un producto	8	✓	
Reponer un producto	8	✓	
Generar cárdex de un producto	16	✓	✓
Programación del módulo de reportes e Indicadores			
Mostrar gráfico de indicador de venta de habitaciones	24	✓	✓
Mostrar gráfico de indicador de venta de productos	24	✓	✓
Mostrar gráfico de indicador de gastos	24	✓	✓
Mostrar gráfico de ranking de clientes	24	✓	✓
Integración de Módulos	80	✓	✓
PRUEBAS			
Configuración del servidor de pruebas	48	✓	✓
Pruebas de funcionalidad			
Validación con los usuarios	40	✓	✓
Pruebas de desempeño			
Verificación del uso del procesador	4	✓	
Verificación de descarga de paquetes de datos	4	✓	
DESPLIEGUE			
Configuración de servidor de producción	16	✓	
Instalación de la aplicación	8	✓	
Configuración de copias de seguridad	8	✓	
Total horas	982		
Horas mensuales (22 días * 8 hrs)	176		
Cantidad meses	5.58		

Fuente: Elaboración propia

Implementación

Luego de tener los requerimientos y el diseño de la base de datos aprobado se creó una base de datos por cada hotel. Según la arquitectura del sistema cada negocio tendría su propia base de datos.

Para el desarrollo del software se usó la metodología del Kanban de David Anderson que consistió en el uso de un tablero para organizar las tareas en tres grupos: las tareas pendientes, las que se encuentren terminadas y las que ya se terminaron.

Según el requisito no funcional RNF03 y RNF04 los lenguajes de programación a utilizar serían PHP y Javascript para el backend y frontend respectivamente. La programación se dio en el orden que figura en el calendario de actividades.

Programación del módulo de Acceso.

Para el módulo de acceso se trabajó con la tabla “usuario” y se satisfizo el requerimiento funcional RF02. Como se trataba de un sistema con muchas bases de datos fue necesario también identificar a la empresa para conectarse a su base de datos correspondiente. El tiempo empleado para el desarrollo de este módulo fue de dos días.

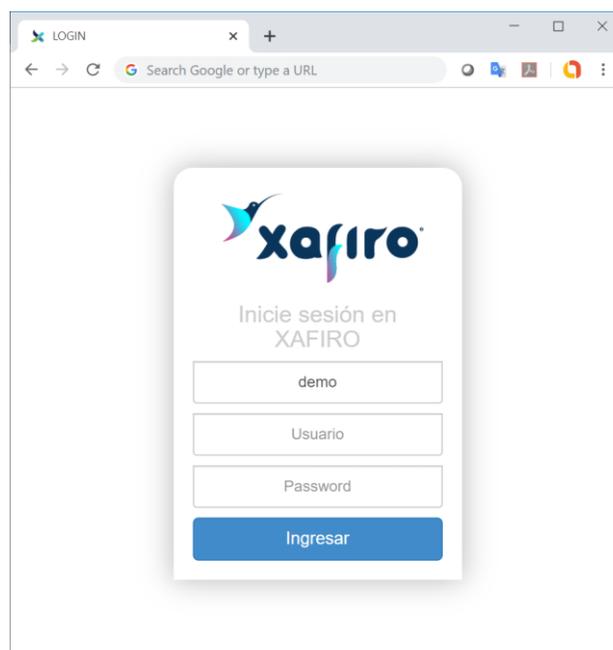


Figura 20. Vista del login de la aplicación. Fuente: Elaboración propia

Para evitar riesgos de seguridad respecto a ataques de fuerza bruta se programó un código de seguridad (captcha) luego de tres intentos fallidos.

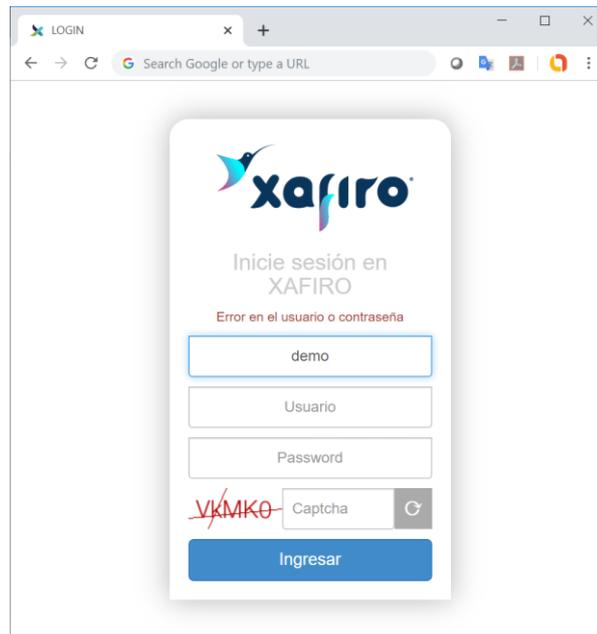


Figura 21. Vista del login tras intentos fallidos. Fuente: Elaboración propia

Módulo de Recepción.

Para el módulo de recepción se trabajaron varios requerimientos. Desde el requerimiento RF02 hasta el RF16. La programación de este módulo tomo 31 días. Se muestra un ejemplo de las tareas correspondientes a este módulo en el tablero Kanvan

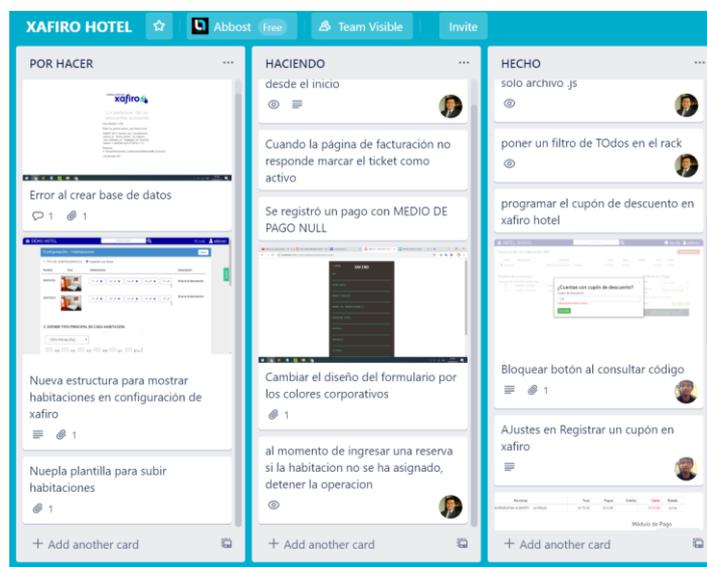


Figura 22. Tablero Kanvan para el desarrollo del módulo de recepción. Fuente: Trello.

El proyecto utilizó el framework codeigniter para facilitar el desarrollo del sistema. Una de las ventajas principales fue la separación de la capa vista, de la capa controlador y de la capa modelos. En la figura 23 se puede apreciar la sintaxis para obtener el reporte de huéspedes.

```
13
14 //LIBRO DE HUESPEDES
15 public function index(){
16     $data['title'] = 'Libro de huéspedes';
17     $data['entrada'] = date('Y-m-d',strtotime('-3days'));
18     $data['salida'] = date('Y-m-d',strtotime('+30days'));
19     $data['canales'] = $this->model_canal->getAllCanales(1);
20     $data['page'] = 'book';
21     $this->load->view('plantillas/header_menu',$data);
22     $this->load->view('reserva/index',$data);
23     $js['page'] = "lay";
24     $this->load->view('plantillas/footer',$js);
25 }
26
27 public function listar_estancia_habitaciones($entrada,$salida){
28     $salida = get_limite_fecha($entrada,$salida);
29     $all_negocio = config_load();
30     $data['doble_moneda'] = $all_negocio['doble_moneda'];
31     $data['canales'] = $this->model_canal->getAllCanales(true);
32     $data['listar_estancia_habitaciones'] = $this->model_reserva_detalle->
33         m_cargar_estancia_habitaciones($entrada,$salida,$all_negocio['
34         doble_moneda']);
35     $this->load->view('reserva/listar',$data);
36 }
37 }
```

Figura 23. Vista del controlador Reserva. Fuente: Elaboración propia.

El diseño del rack fue inspirado en el tablero de llaves que usan los recepcionistas. En el que cada llave es identificable rápidamente.



Figura 24. Vista de un tablero de llaves tradicional. Fuente: Google.

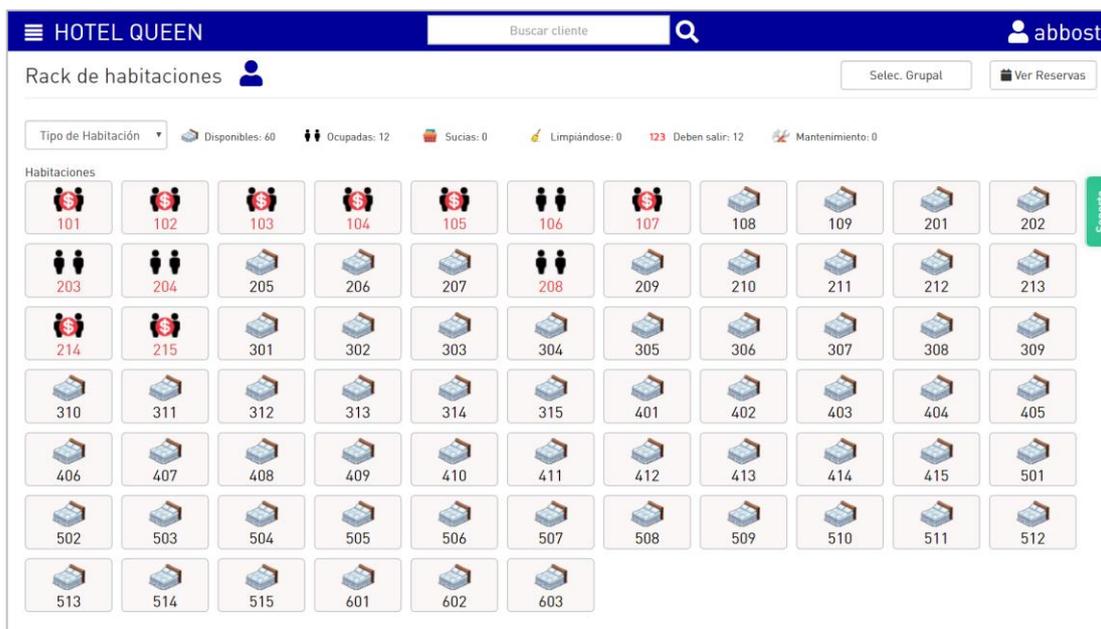


Figura 25. Vista del rack de habitaciones del PMS Xafiro. Fuente: Elaboración propia.

En el caso del planning la programación se realizó usando funciones asíncronas de carga (AJAX) en Javascript para aprovechar los recursos del cliente y no descargar todo el html que arma el planning desde el servidor.

```

2202
2203
2204 function planning(date1){
2205     $('#myPleaseWait').modal('show');
2206     var data = {};
2207     data.date1 = date1;
2208
2209     // Aquí llenan la tabla
2210     $.post(base_url + 'reserva/load/', data, function(response){
2211         for(var i = 0, l = response.length; i < l; i++){
2212             date1 = new Date(response[i].date1.getTime());
2213             date2 = new Date(response[i].date2.getTime());
2214             room = response[i].rid;
2215             dlh = response[i].id;
2216             shift = Math.ceil(Math.abs(date2 - date1) / (1000 * 3600 * 24));
2217             cell = $('td.day[data-room="'+room+'"][data-date="'+response[i].date1+'"]').first();
2218
2219             if(!exists(cell)){
2220                 date1 = $('#first_date').val();
2221                 //date1 debe ser la fecha inicial de la tabla
2222                 date1 = new Date( date1_ .getTime());
2223                 date2 = new Date( response[i].date2 ).getTime();
2224                 shift = Math.ceil(Math.abs(date2 - date1) / (1000 * 3600 * 24));
2225                 cell = $('td.day[data-room="'+room+'"][data-date="'+ date1_ +'"]').first();
2226                 shift++;
2227             }
2228
2229             /* NO BORRAR */
2230             cellPointer = cell
2231             x = 1;
2232             while( x < shift ){
2233                 cellPointer.removeClass('toselect');
2234                 cellPointer = cellPointer.next();
2235                 x++;
2236             }
2237
2238             if(response[i].tipo == 'reserva'){
2239                 canal = "reserva-"+response[i].canal;
2240                 cell.append($('<a href="#" class="full" +canal+" style="width: '+$(cell).width()*shift+'px"></a>').text(response[i].nombre.toUpperCase()));
2241                 cell.removeClass('toselect').addClass('reserva');
2242             }else{
2243                 cell.append($('<a href="#" class="full" style="width: '+$(cell).width()*shift+'px"></a>').text((response[i].nombre?response[i].nombre:'.').t
2244                 cell.removeClass('toselect').addClass('ocupado');
2245             }
2246         }
2247         $('#myPleaseWait').modal('hide');
2248     });
2249 }

```

Figura 26. Vista de la función asíncrona para generar el planning de reservas.

HOTEL QUEEN		Buscar cliente							abbo
Planning		Vie	Sab	Dom	Lun	Mar	Mier	Jue	Vie
		18	19	20	21	22	23	24	25
101	RICKET MARTIN								
102	JOSE TRIGOSO ...	JUAN PEREZ							
103	CARLOS ALBER...								
104	NORAH ROXAN...			LUIS SUAREZ					
105	CARLOS ALBER...								
106	NORA AGUEDIT...								
107	CARLOS ALBER...								

Figura 27. Vista del planning de reservas del PMS Xafiro. Elaboración propia.

Para el punto de venta se realizó la verificación del stock de forma asíncrona. Esto evitó que la página se tenga que recargar y el usuario pierda los datos que ya tenía cargados

```

project Preferences Help
product.js x check stock.js
1 $("#product-cart").livequery(function(){
2   $(this).autocomplete({
3     minLength: 4,
4     source: function( request, response ) {
5       $.ajax({
6         type: "POST",
7         url: base_url+"producto/check_stock",
8         dataType: "json",
9         data: {
10          codigo: encodeURIComponent(request.param),
11          cantidad: $("#cantidad").val()
12        },
13        success: function( data ) {
14          response( data );
15        }
16      });
17    },
18    focus: function() {return false;},
19    response: function( event, ui ) {
20      item = ui.content[0];
21      brokenStock = item.brokenStock;
22      itemId = item.itemId;
23      descripcion = item.descripcion;
24      precio = item.precio;
25      cantidad = item.cantidad;
26      if(brokenStock)
27        alertify.alert("No hay stock del producto");
28      else
29        addCarrito(itemId,descripcion,precio,cantidad);
30    }
31  }

```

Figura 28. Vista de la función "consultar stock" hecha en javascript.

Punto de Venta

Tipo de Complemento: PRODUCTO RECEPCIC *almacén

Tipo de Comprador: HUESPED

Número de habitación: 203 ✓ CARLOS ALBERTO NORABUENA ARANDA

Detalle de Compra

- 1 + Nombre del Producto

Acción	Cant.	Item	Precio Unit	Precio Total
X	1	coca cola 1l 1/2	S/ 10.00	S/ 10.00
X	1	agua san luis	S/ 2.00	S/ 2.00
X	1	agua san mateo	S/ 2.00	S/ 2.00
X	1	afeitador	S/ 4.00	S/ 4.00

4 item(s) en la compra

Módulo de Pago

Medio de pago: Pagar al salir

Total: **S/ 18.00**

REGISTRAR VENTA

Soporte

Figura 29. Vista del módulo de venta del Xafiro. Fuente: Elaboración propia

El cierre de caja fue un reporte de todo lo transcurrido en el turno actual (RF14, RF15, RF16 y RF17). Este reporte consistió en sumar todos los ingresos y restar los egresos. Además, se podía consultar el detalle de las transacciones realizadas.

Arqueo de caja diario

Históricos

Elige fecha Ver

>> Ver caja activa

>> Ver caja mensual

RECEPCION

Ultimo cierre por ROCIO DEL PILAR el 16 de Octubre a las 05:23 pm

Arqueo de Caja 1

Saldo inicial :	S/ 100.00
Ingresos	S/ 346.00
Movimientos :	S/ 346.00
Ingresos extra :	S/ 0.00
Egresos	S/ 100.00
Gastos corrientes :	S/ 100.00
Total	S/ 346.00

Resumen:

Venta de habitaciones	S/ 1,218.00
Venta de productos	S/ 50.00
Venta de servicios	S/ 0.00
Otros	S/ 30.00

COMPOSICION:

Efectivo:	S/ 346.00
Tarjeta :	S/ 0.00
Depósitos :	S/ 0.00

CERRAR CAJA

Soporte

Figura 30. Vista del cierre de caja del PMS Xafiro. Fuente: Elaboración propia

Desarrollo de módulo de clientes.

El módulo de clientes se desarrolló en un tiempo de 60 días. Los requerimientos que definían este módulo fueron el RF18, RF19, RF20, RF21 y RF22. Todas las tareas fueron ubicadas en el tablero Kanvan.

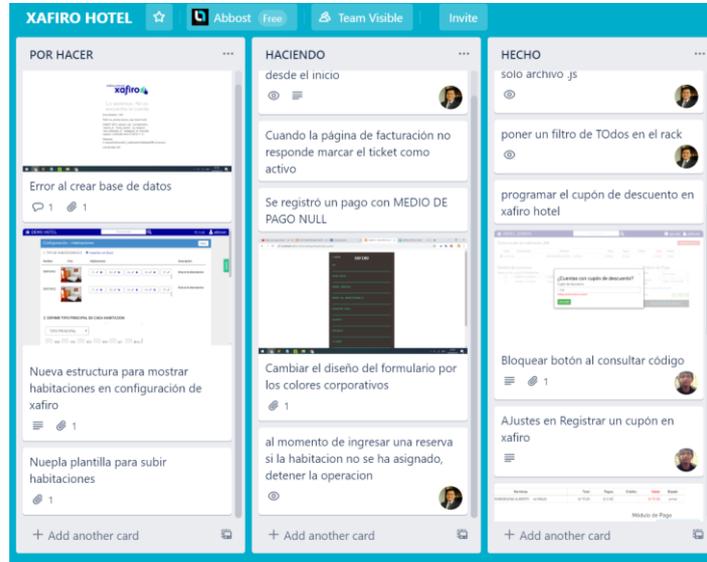


Figura 31. Tablero Kanvan para el desarrollo del módulo de clientes. Fuente: Trello

Se diseñó el perfil del cliente para mostrar la mayor cantidad de información del cliente en una sola pantalla.

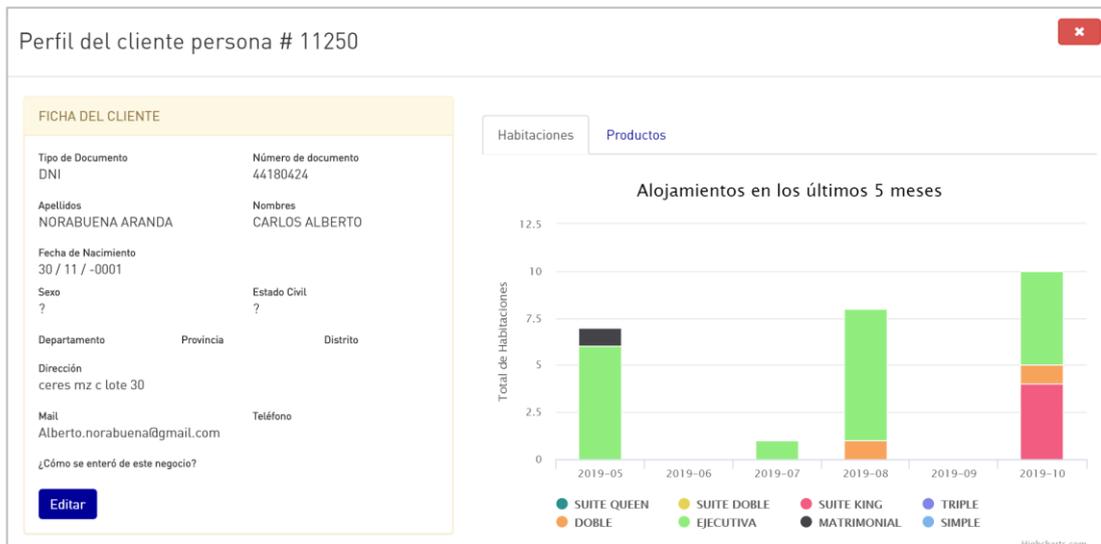


Figura 32. Vista del perfil del cliente del PMS Xafiro. Fuente: Elaboración propia

Desarrollo del módulo de backoffice

Al igual que en los otros módulos se registraron las tareas en el tablero Kanvan

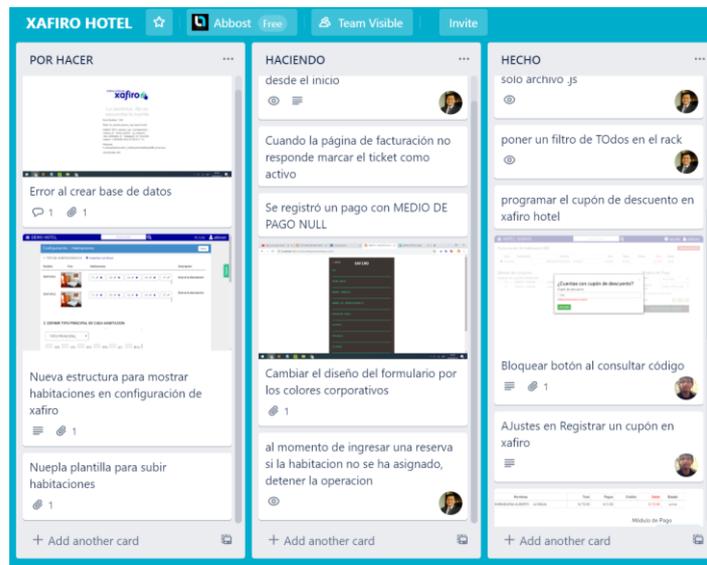


Figura 33. Tablero Kanvan para el desarrollo del módulo de backoffice. Fuente: Trello

Para el registro de gastos se programó una ventana que tenga la lista de gastos del día pero que también tenga un botón para registrar gastos.

Reporte de Gastos / Extras para el día 2019-10-20. Se muestran 4 registros.

Acción	Fecha	Descripción	Comprobante	Costo	Caja	Categoría	Responsable	Estado
	20 Oct - 10:07 pm	compra de diario		S/ 1.50	RECEPCION	otros	abbost	ACTIVO
	20 Oct - 10:06 pm	compra de foco ahorrador	13924	S/ 3.50	RECEPCION	otros	abbost	ACTIVO
	20 Oct - 10:06 pm	compra de ambientador	12345	S/ 45.00	RECEPCION	limpieza	abbost	ACTIVO
	20 Oct - 10:05 pm	compra de detergente	122939	S/ 50.00	RECEPCION	limpieza	abbost	ACTIVO

Figura 34. Vista del reporte de gastos del PMS Xafiro. Elaboración propia

Figura 35. Vista del formulario para registrar gastos

Para los inventarios se diseñó un reporte que tenga el nivel de stock de los productos. Desde esta misma pantalla se podía acceder al Kardex de productos y reponer productos.

Acciones	Código	Nombre	Familia	Cajas	Unidades	Stock Total	Precio Unit.	Precio Caja	Precio Trab.
	00	bonawell	aseo	0	40	40	S/ 1.00	S/ 0.00	S/ 1.00
	00	desodorante mujer	aseo	0	0	0	S/ 1.00	S/ 0.00	S/ 1.00
	00	desodorante hombre	aseo	0	9	9	S/ 1.00	S/ 0.00	S/ 1.00
	00	toallas higienicas	aseo	0	4	4	S/ 1.00	S/ 0.00	S/ 1.00
	00	jabon camay	aseo	0	9	9	S/ 2.50	S/ 0.00	S/ 2.50
	00	afeitador	aseo	0	6	6	S/ 4.00	S/ 0.00	S/ 4.00
	00	sporade	bebidas	0	0	0	S/ 3.00	S/ 0.00	S/ 3.00
	00	cepillo dental	aseo	0	2	2	S/ 2.00	S/ 0.00	S/ 2.00
	00	shampu hys	aseo	5	17	77	S/ 1.50	S/ 0.00	S/ 1.50
	00	bolsas negra 20x30	limpieza	0	9	9	S/ 0.00	S/ 0.00	S/ 0.00
	00	bolsas de 220 lt.	limpieza	0	2	2	S/ 0.00	S/ 0.00	S/ 0.00

Figura 36. Vista del reporte de inventario del PMS Xafiro. Elaboración propia

Desarrollo del módulo de reportes

Los trabajos en este módulo comenzaron actualizando el tablero de avance de tareas.

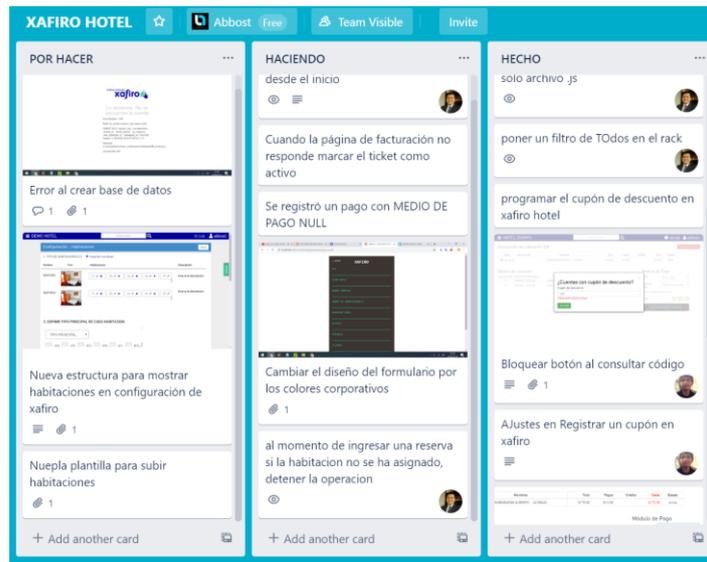


Figura 37. Tablero Kanvan para el desarrollo del módulo de reportes. Fuente: Trello

Según el requerimiento RF31 fue necesario medir el nivel de venta de habitaciones. Para ello se programó un gráfico de barras mensual usando la herramienta HighCharts.

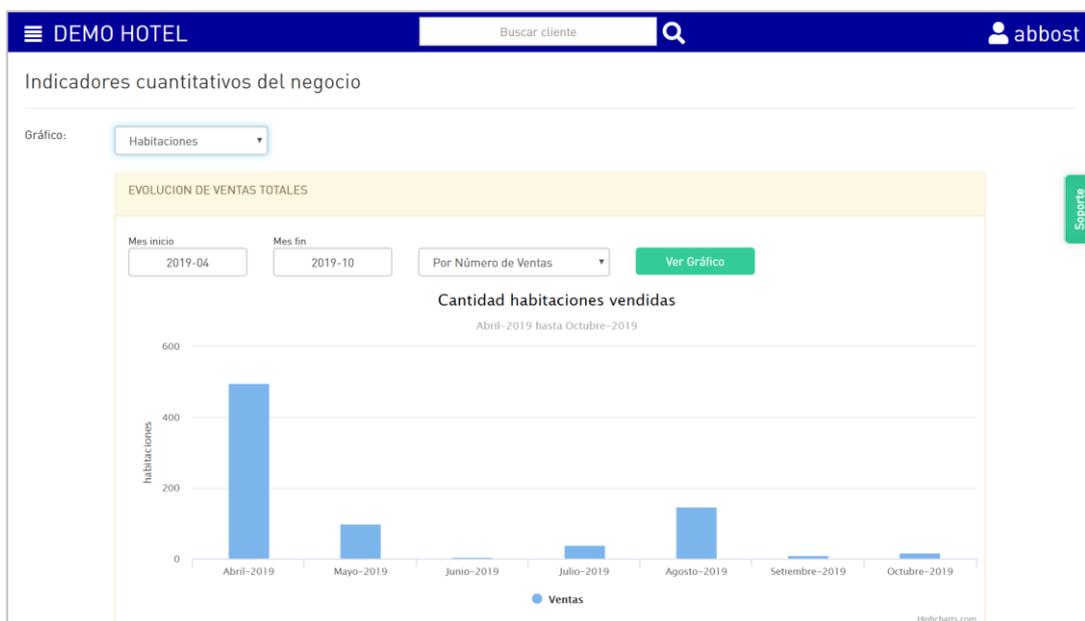


Figura 38. Vista del indicador de venta total de habitaciones del PMS Xafiro.

Fuente: Elaboración propia

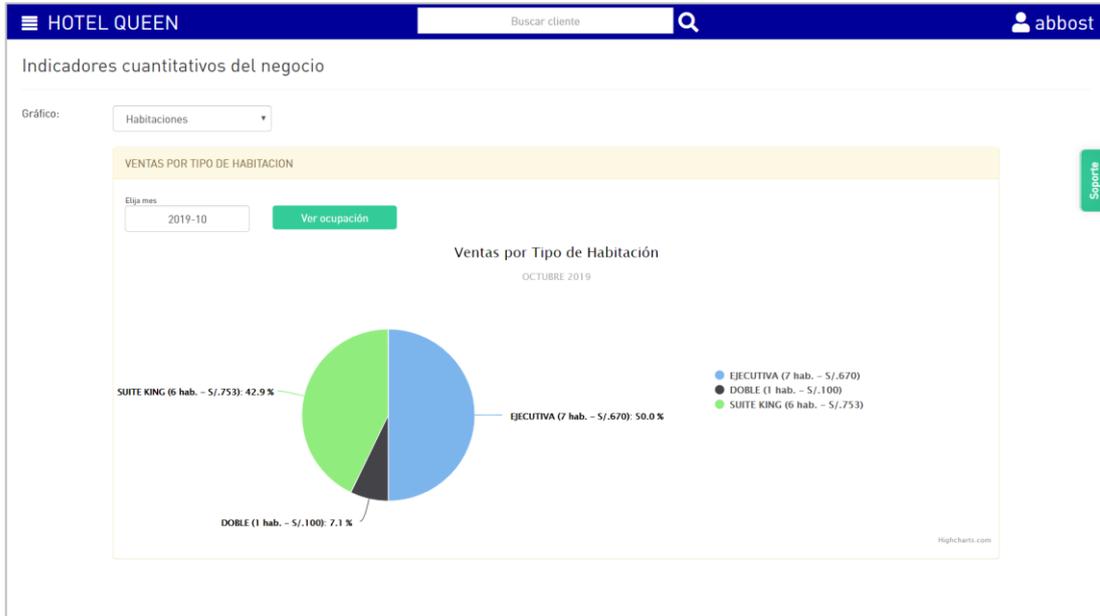


Figura 39. Vista del indicador de venta por tpo de habitación del PMS Xafiro. Fuente: Elaboración propia

Según el requerimiento RF32 fue necesario medir el nivel de venta de productos por categoría. Para ello se programó un gráfico tipo pie usando la herramienta HighCharts.

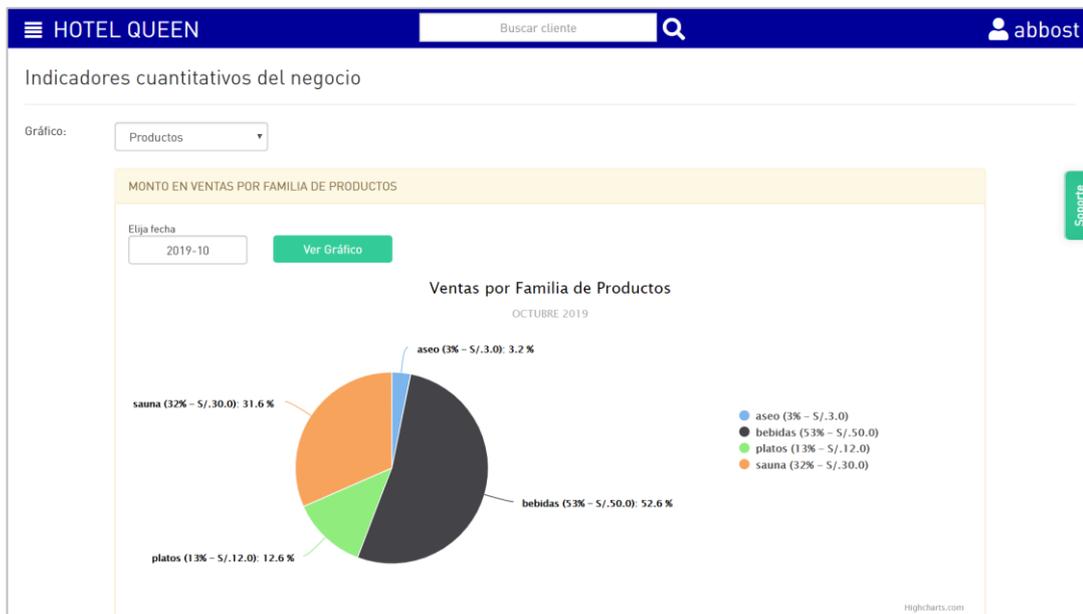


Figura 40. Vista del indicador de venta por tipo de producto del PMS Xafiro. Fuente: Elaboración propia

El requerimiento RF33 indicaba que se necesitaba ver el monto de gastos desembolsado mes a mes incluso por categoría.

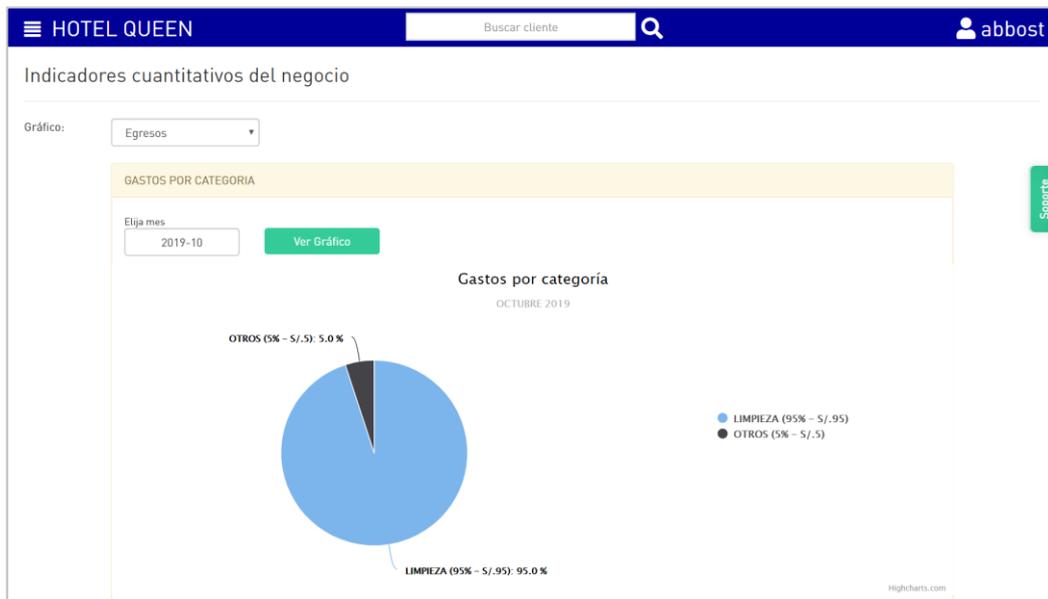


Figura 41. Vista del indicador de gasto por categoría del PMS Xafiro. Fuente: Elaboración propia

Según el requerimiento RF34 fue necesario tener un ranking de clientes frecuentes. Para ello se programó un gráfico de barras que recibía como input el nivel de facturación deseable.

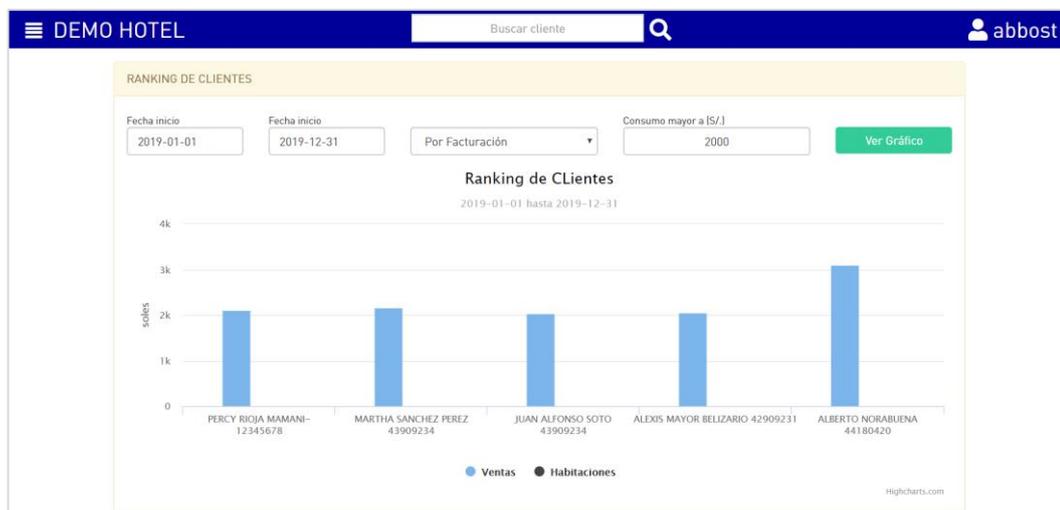


Figura 42. Vista del indicador de venta total de habitaciones del PMS Xafiro. Fuente: Elaboración propia

Integración de módulos.

Antes de pasar a las pruebas con los usuarios se hicieron pruebas integrales de todo el sistema. De esa forma se pudieron encontrar errores como links rotos, desborde del tamaño permitido por la base de datos, entre otros.



Figura 43. Vista responsiva del PMS adaptable a múltiples pantallas.

Fuente: Elaboración propia

Pruebas

Hasta este punto todas las pruebas habían sido realizadas en un entorno local. Sin embargo, era momento de solicitar la validación del usuario. El plan era que los usuarios se conecten a través de un navegador web al PMS y puedan validar los requerimientos.

Para que el PMS pueda estar en la nube se tuvo que elegir una región donde configurar el servidor de pruebas.

AWS poseía 60 datacenters en diversos países y regiones del mundo. Considerando el factor de la distancia para reducir la latencia se evaluó el datacenter ubicado en Sao Paulo (Brasil) y el ubicado en Oregón (Estados Unidos) por su proximidad a Perú-



Figura 44. Imagen de la conexión establecida entre los usuarios de Perú y los datacenter de AWS en Brasil y Estados Unidos. Fuente: Amazon.com

Fue necesario evaluar la latencia en diferentes horarios. Mañana y tarde. El horario de mañana (antes de las 12pm) era más liviano en comparación al horario tarde por la cantidad de transacciones que se daban. Como por ejemplo checkin y checkout de pasajeros.

Se hicieron pruebas de latencia en ambos horarios con el protocolo ICMP descargando paquetes de 32 bytes de datos del servidor.

<pre> Cmder λ ping 35.165.230.16 Pinging 35.165.230.16 with 32 bytes of data: Reply from 35.165.230.16: bytes=32 time=2203ms TTL=233 Reply from 35.165.230.16: bytes=32 time=275ms TTL=233 Reply from 35.165.230.16: bytes=32 time=366ms TTL=233 Reply from 35.165.230.16: bytes=32 time=297ms TTL=233 Ping statistics for 35.165.230.16: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 275ms, Maximum = 2203ms, Average = 785ms </pre>	<pre> Cmder λ ping 35.165.230.16 Pinging 35.165.230.16 with 32 bytes of data: Reply from 35.165.230.16: bytes=32 time=152ms TTL=234 Reply from 35.165.230.16: bytes=32 time=150ms TTL=234 Reply from 35.165.230.16: bytes=32 time=163ms TTL=234 Reply from 35.165.230.16: bytes=32 time=152ms TTL=234 Ping statistics for 35.165.230.16: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 150ms, Maximum = 163ms, Average = 154ms </pre>
---	---

Figura 45. Comando ping a un servidor AWS ubicado en Oregón (Estados Unidos). A la izquierda horario mañana y a la derecha horario tarde. Fuente: Elaboración propia

```

Cmder
λ ping 18.228.129.67

Pinging 18.228.129.67 with 32 bytes of data:
Reply from 18.228.129.67: bytes=32 time=97ms TTL=237
Reply from 18.228.129.67: bytes=32 time=95ms TTL=237
Reply from 18.228.129.67: bytes=32 time=95ms TTL=237
Reply from 18.228.129.67: bytes=32 time=98ms TTL=237

Ping statistics for 18.228.129.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 95ms, Maximum = 98ms, Average = 96ms

Cmder
λ ping 18.228.129.67

Pinging 18.228.129.67 with 32 bytes of data:
Reply from 18.228.129.67: bytes=32 time=200ms TTL=240
Reply from 18.228.129.67: bytes=32 time=202ms TTL=240
Reply from 18.228.129.67: bytes=32 time=199ms TTL=240
Reply from 18.228.129.67: bytes=32 time=196ms TTL=240

Ping statistics for 18.228.129.67:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 196ms, Maximum = 202ms, Average = 199ms

```

Figura 46. Comando ping a un servidor AWS ubicado en Sao Paulo Brasil). A la izquierda horario mañana y a la derecha horario tarde

Mientras que en Brasil la latencia promedio -en horario tarde- fue de 199 milisegundos, en Estados Unidos fue de 154 milisegundos. A pesar de que ambos tenían 21 redes o puntos intermedios, la diferencia en tiempo favoreció al datacenter de Estados Unidos.

Esa diferencia era consecuencia de la congestión de las redes intermedias que tenían que atravesar los paquetes para llegar desde el servidor de AWS al usuario ubicado en Perú.

```

Cmder
λ tracert 35.165.230.16

Tracing route to ec2-35-165-230-16.us-west-2.compute.amazonaws.com [35.165.230.16]
over a maximum of 30 hops:
  0  6 ms   1 ms   2 ms  192.168.0.1
  1  16 ms  18 ms  15 ms  10.137.128.1
  2  15 ms  15 ms  15 ms  10.150.144.57
  3  17 ms  13 ms  18 ms  10.95.156.46
  4  121 ms 200 ms 221 ms las-bb1-link.telia.net [62.115.35.125]
  5  152 ms 195 ms 144 ms a100-ic-315571-las-b21.c.telia.net [213.248.104.234]
  6  *      *      *      Request timed out.
  7  *      *      *      Request timed out.
  8  *      *      *      Request timed out.
  9  *      *      *      Request timed out.
 10 *      *      *      Request timed out.
 11 *      *      *      Request timed out.
 12 *      *      *      Request timed out.
 13 *      *      *      Request timed out.
 14 *      *      *      Request timed out.
 15 *      *      *      Request timed out.
 16 *      *      *      Request timed out.
 17 *      *      *      Request timed out.
 18 *      *      *      Request timed out.
 19 *      *      *      Request timed out.
 20 *      *      *      Request timed out.
 21 339 ms 230 ms 155 ms ec2-35-165-230-16.us-west-2.compute.amazonaws.com [35.165.230.16]

```

Figura 47. Comando “Tracert” desde Lima (Perú) a un servidor AWS ubicado en Oregón (Estados Unidos). Fuente: Elaboración propia

Configuración del servidor de pruebas.

Una vez elegido la región donde configurar los servidores, se continuó con la configuración de estos. De acuerdo con la arquitectura del sistema se configuró un servidor de base de datos y un servidor de aplicaciones.

Servidor de base de datos.

Según el requerimiento RNF05 se configuró el gestor de base de datos MySQL. Se configuró en un servidor independiente para asegurar la seguridad y estabilidad del sistema. De haber una caída en el servidor de aplicación el servidor de datos seguiría funcionando. Se reiniciaría el servidor de aplicación o se crearía otro rápidamente y bastaría con enlazar la IP del servidor de base datos para que el sistema siga funcionando.

Servidor de aplicaciones.

Este servidor tendría instalada la aplicación desarrollada. Según el requerimiento RNF06 se instaló Codeigniter con PHP versión 7.0. Este servidor recibiría todas las peticiones de los usuarios, las procesaría y si fuera necesario se conectaría al servidor de base de datos para obtener o registrar datos.

Después de haber configurado correctamente los servidores se consultó el panel de control de AWS para verificar su estado. Se verificó que ambos servidores tenían el estado “running” lo que indicó que ambos servidores se encontraban en línea y listos para usarse.

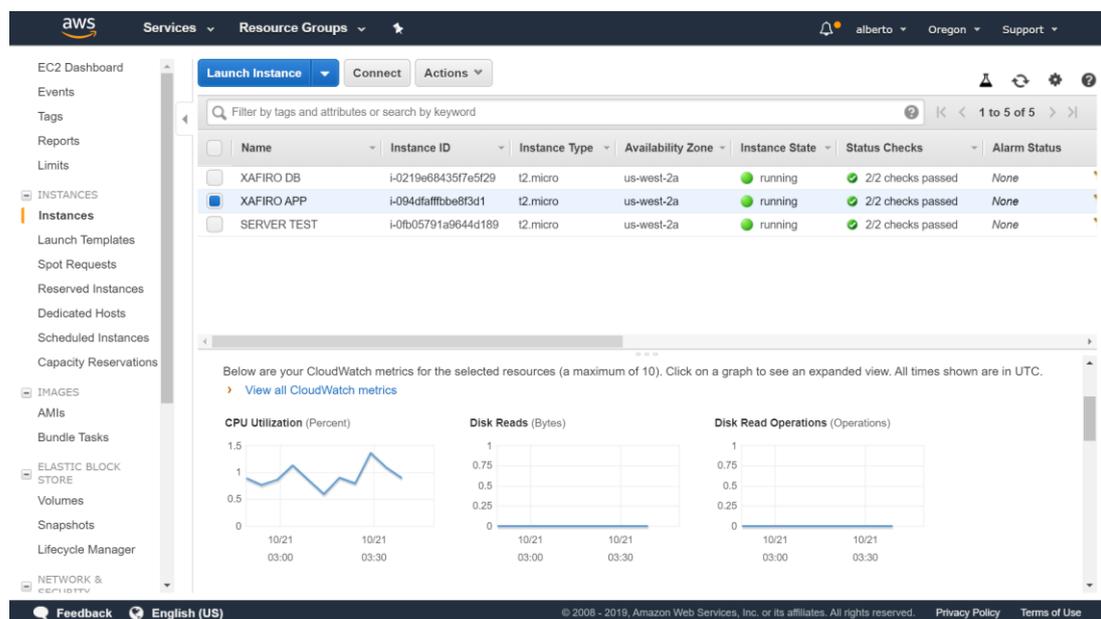


Figura 48. Panel de administración de instancias EC2 Amazon.

Fuente: Amazon Web Services

Pruebas de funcionalidad.

Una vez configurado los servidores y habiéndose instalado el aplicativo, se contactó con los usuarios de los hoteles para que puedan hacer la validación de los requerimientos funcionales. De esta etapa se encontraron 20 observaciones que fueron incluidas en la lista de tareas pendientes.

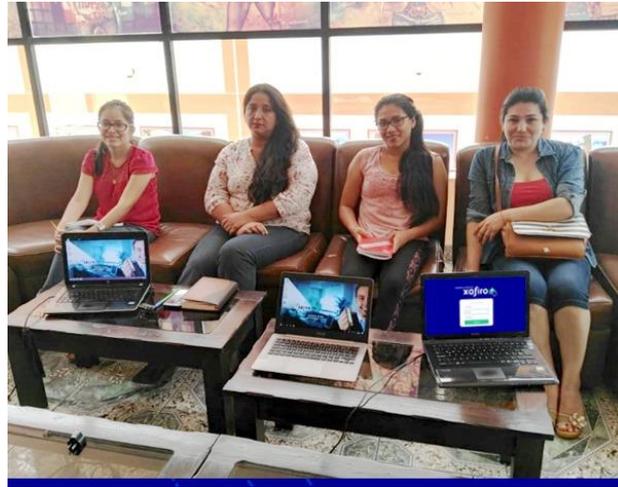


Figura 49. Trabajadores de recepción probando el sistema Xafiro. Elaboración propia

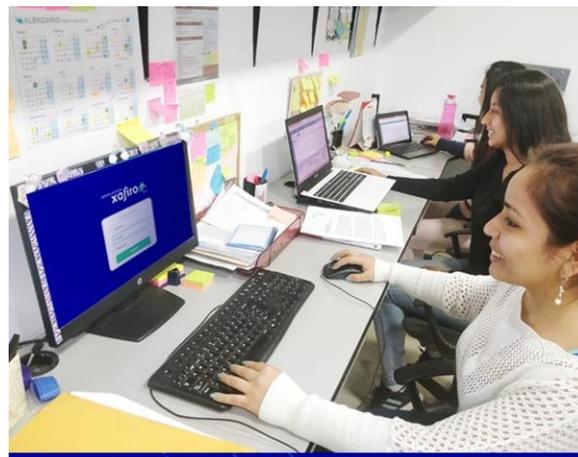


Figura 50. Trabajadores de recepción probando el sistema Xafiro. Elaboración propia

Pruebas de desempeño.

Para evaluar el desempeño del hardware se utilizó el mínimo recomendado. Tal como lo indica la tabla 11.

Tabla 18

Hardware recomendado para el funcionamiento del PMS

PC cliente	Servidor AWS
Procesador Intel 1.6 GHz	Procesador INTEL XEON 3.3 GHz
Memoria RAM 4GB	Memoria RAM 1GB
Disco duro 80GB	Disco Duro 8GB
Tarjeta de red 1Gbps	Tarjeta de red virtual 1 Gbps
Ancho de banda ADSL 4Mbps	

Fuente: Elaboración propia

Las pruebas de desempeño sobre las transacciones se hicieron por un periodo de 15 días. Se midió el uso del procesador y el tamaño de los datos descargados. En los resultados se observó que el procesador apenas se utilizó al 8% de su capacidad y que los paquetes descargados tuvieron un peso promedio de 0.2 MB. Con ese peso y considerando el ancho de banda del cliente y la latencia de la comunicación se confirmó que los paquetes de datos se descargarían en un tiempo no mayor a 5 segundos.

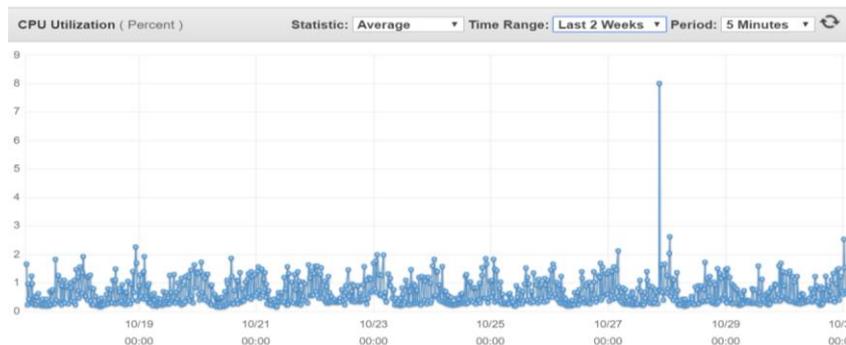


Figura 51. Uso del procesador del servidor. Fuente: Cloudwatch de AWS

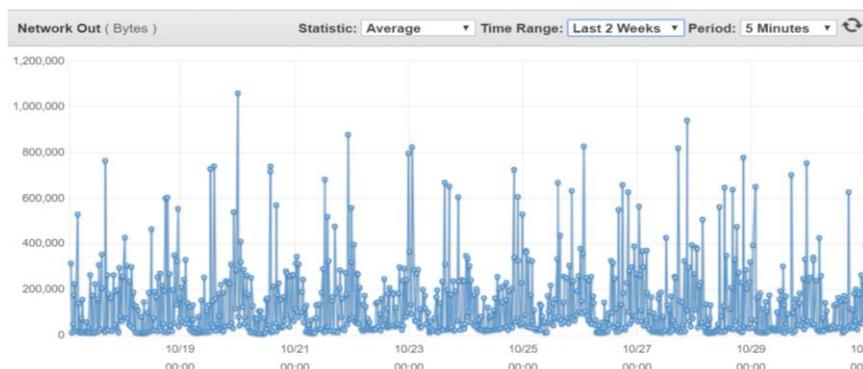


Figura 52. Paquetes descargados del servidor. Fuente: Cloudwatch de AWS

Despliegue o pase a producción

Luego que el servidor de pruebas superó las pruebas de desempeño se configuraron los dos servidores de producción con las mismas características: Un servidor de aplicación y un servidor de base de datos. Además, se configuraron las copias de respaldo (backup) y se aseguró de confirmar el nivel de servicio (SLA) ofrecido por Amazon Web Services.

Copias de Respaldo.

Una tarea muy importante fue asegurar la seguridad de la información y su disponibilidad de forma inmediata. Por ello fue necesario elaborar un plan de respaldo o copia de seguridad (backups) cada 6 horas.

En este proceso, se usó el servicio de SNAPSHOT de Amazon Web Services que consiste en hacer un full backup al disco duro de la instancia EC2. Este servicio fue automatizado a través de una tarea programada dentro del panel de control de AWS.

Si en algún momento el servidor de base de datos tuviera algún desperfecto se podría restaurar la información casi de manera inmediata. Solo bastaría con encontrar el último SNAPSHOT generado y convertirlo en un Amazon Machine Image (AMI). Posteriormente se configuraría un nuevo servidor partiendo del AMI anteriormente creado. El servidor estaría listo y disponible en aproximadamente 10 minutos.

Acuerdo de Servicio con Amazon (SLA)

En la modalidad SAAS era importante que físicamente los servidores estén disponibles el 100% de veces. Por ello, según el Acuerdo de Servicio (SLA por sus siglas en inglés) establecido entre Abbot y Amazon, este último se comprometía a que el servicio *Amazon Elastic Compute Cloud (Amazon EC2)* esté disponible el 99.99% de veces durante el mes. En caso no se cumpliera con este acuerdo las sanciones se muestran en la tabla 12.

Tabla 19

Escala de descuentos por caídas del servicio de Amazon Web Services EC2

Tiempo en línea al mes	Descuento del mes
Menos que 99.99% pero mayor igual que 99.00 %	10%
Menos que 99.00% pero mayor igual que 95.00 %	30%
Menos que 95%	100%

Fuente: <https://aws.amazon.com/compute/sla/>

Análisis económico

Un aspecto importante para la continuidad del proyecto fue la rentabilidad. Según los resultados el proyecto resultó ser rentable a partir del tercer año. Se muestra a continuación el detalle de la inversión, ventas y estados de utilidades.

Inversión

El PMS Xafiro fue un servicio que se vendió en modalidad SAAS con un cobro mensual prepago. La inversión del proyecto fue de S/ 22,315 y estuvo destinada principalmente al pago de mano de obra como lo muestra la tabla 20

Tabla 20

Inversión del sistema XAFIRO

Descripción	Costo unitario S/.	
Desarrollo de la aplicación	S/.	16,564.00
Programador 1 x 6 meses	S/.	9,000.00
Programador 2 x 6 meses	S/.	7,200.00
Licencia de Windows 7	S/.	364.00
Oficina y equipos	S/.	2,000.00
Alquiler de 2 laptops x 6 meses *	S/.	1,200.00
Alquiler de oficina x 6 meses **	S/.	600.00
Artículos decorativos	S/.	200.00
Constitución de la Empresa	S/.	682.00
Elaboración de la Minuta	S/.	292.00
Inscripción en SUNARP	S/.	178.00
Trámite SUNAT- obtención de RUC	S/.	100.00
Libros contables + legalización	S/.	112.00
Trámites y Registros	S/.	589.00
Millar de Impresión de facturas y boletas	S/.	167.00
Búsqueda de marca Xafiro en Indecopi	S/.	26.00
Registro Marca Xafiro en Indecopi	S/.	176.00
Publicación en diario El Peruano	S/.	220.00
Marketing y Publicidad	S/.	2,480.00
Página web de Xafiro	S/.	1,700.00
Diseño de Logo	S/.	500.00
1 millar Díptico para Lanzamiento	S/.	200.00
Tarjeta de presentación	S/.	80.00
TOTAL DE INVERSION INICIAL	S/.	22,315.00

* Cada programador uso su propia laptop para el proyecto. A cambio se le pagó 100 soles al mes por concepto de alquiler

** La oficina fue un espacio ubicado dentro la vivienda de unos de los programadores y el alquiler de 100 soles incluía el pago de servicios como agua, luz, teléfono e internet.

Ventas

El PMS Xafiro se vendió en 2 planes. Un plan básico y un plan premium. La diferencia entre ambos planes fue la cantidad de funcionalidades que podía tener el cliente.

Tabla 21

Planes del sistema hotelero Xafiro

Plan	Precio sin IGV
Básico	S/ 250
Premium	S/ 350

Tabla 22

Ingresos de ventas del año 1 al año 5

	Q 2016		Q 2017		Q 2018		Q 2019		Q 2020	
	ANUAL		ANUAL		ANUAL		ANUAL		ANUAL	
Nro clientes	10		10		10		20		30	
Aplicación base (incluye renovaciones)	120	S/ 30,000	120	S/ 30,000	120	S/ 30,000	165	S/ 41,250	230	S/ 57,500
Aplicación Premium (incluye renovaciones)	-	-	-	-	-	-	42	S/.14,700	100	S/.35,000
INGRESOS	120	S/ 30,000	120	S/ 30,000	120	S/ 30,000	207	S/ 55,950	330	S/ 92,500

Tabla 23

Participación de ventas por servicios ofrecidos

	2016	2017	2018	2019	2020
Aplicación base	100%	100%	100%	80%	70%
Aplicación Premium	0%	0%	0%	20%	30%
TOTAL	100%	100%	100%	100%	100%

Estado de Ganancias y pérdidas:

En los estados de ganancias y pérdidas se debe aclarar que entre el periodo 2016 – 2018 se presentaron nuevos clientes que requerían soluciones a la medida. Por tanto, como nueva oportunidad de negocio, Abbost decidió enfocarse a ello y postergar la comercialización de Xafiro, este motivo hizo que no haya crecimiento en esos años.

Aun así, se puede observar que el retorno de la inversión fue de 3 años. Ya que la inversión de S/ 22,315 fue recuperada al tercer año con S/ 23,450 soles

Tabla 24

Estado de ganancias y pérdidas del 2016 al 2020

	2016		2017		2018		2019		2020	
VENTAS	S/.	30,000	S/.	30,000	S/.	30,000	S/.	55,950	S/.	92,500
Costo Variable (Hosting, y Dominio)	S/.	900	S/.	900	S/.	900	S/.	900	S/.	1,800
MG DE CONTRIBUCION	S/.	29,100	S/.	29,100	S/.	29,100	S/.	55,050	S/.	90,700
Sueldo programador	S/.	16,800								
Sueldo vendedor		-	S/.	3,400		-		-	S/.	13,020
Gasto de ventas 10% (comisiones)		-		-		-		-	S/.	10,200
UTILIDAD ANTES DE IMPUESTOS	S/.	12,300	S/.	8,900	S/.	12,300	S/.	38,250	S/.	50,680
Impuesto a la Renta 30%	S/.	3,690	S/.	2,670	S/.	3,690	S/.	11,475	S/.	15,204
UTILIDAD NETA	S/.	8,610	S/.	6,230	S/.	8,610	S/.	26,775	S/.	35,476
UTILIDAD ACUMULADA	S/.	8,610	S/.	14,840	S/.	23,450	S/.	50,225	S/.	85,701

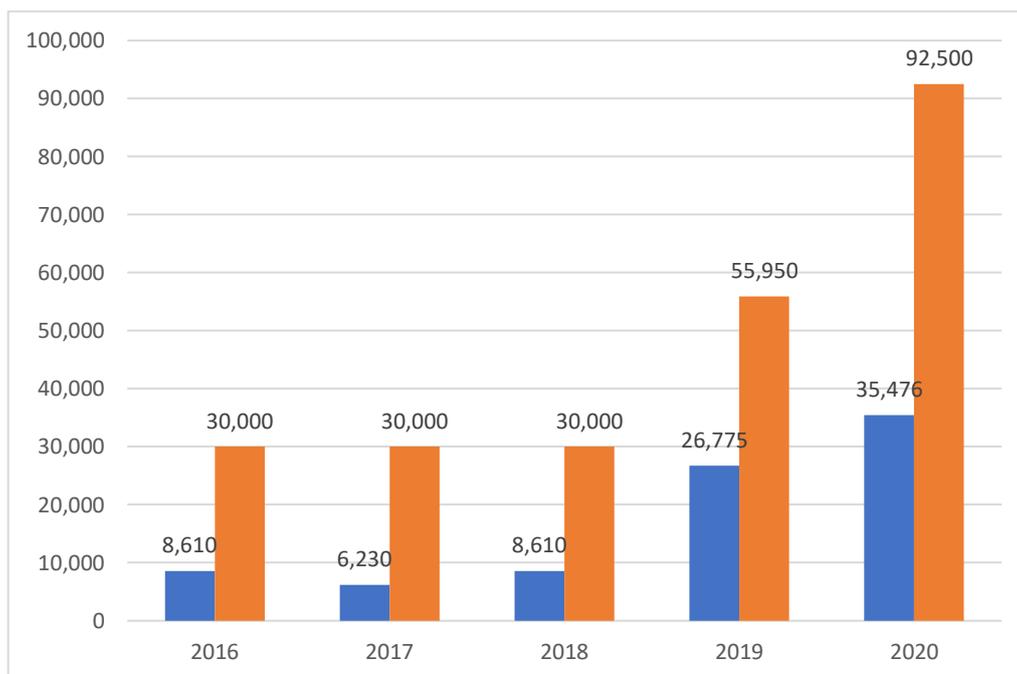


Figura 53. Gráfico de evolución de ventas y utilidades en soles. En color naranja las ventas y en azul las utilidades. Fuente: Elaboración propia.

Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR).

Para el cálculo del valor actual neto se usó una tasa de interés del 10%. Si el VAN resulta mayor a cero es un indicador positivo para realizar la inversión. Asimismo, si la tasa interna de retorno indica un valor positivo también es un buen indicador para realizar el proyecto. Para el cálculo de los flujos se tomo en cuenta la utilidad después de impuesto proyectada.

Los resultados mostraron un escenario favorable.

Tabla 25

Cálculo del VAN y TIR

	2015	2016	2017	2018	2019	2020
FLUJOS	-S/. 22,315	S/. 8,610	S/. 6,230	S/. 8,610	S/. 26,775	S/. 35,476
VAN	S/. 37,445					
TIR	47%					

Análisis y resultados

A mediados del 2015, luego de varios meses de trabajo, se logró el objetivo de construir un sistema informático de tipo PMS que cumpla con los requerimientos establecidos por el cliente.

Para poder comprobar el impacto de este sistema informático sobre la gestión del hotel fue necesario recolectar datos por los siguientes tres años. No se pudo contrastar los resultados de ventas obtenidos con los resultados del sistema manual debido a que la mayoría de los cuadernos y libros tenían contenido ilegible, estaban extraviados o deteriorados por el tiempo. A raíz de esto, se decidió tomar el año 2016 como año base para comparar los resultados obtenidos en el 2017 y 2018.

Se eligieron 4 indicadores clave en la gestión del hotel: el incremento de ventas, el nivel de ocupación del hotel, el ahorro de horas hombre y la rotación del personal de recepción.

Era evidente que el software en sí mismo no iba a generar más ventas, sin embargo, sirvió como una herramienta de control para confirmar que las decisiones tomadas por el administrador eran las correctas o no.

El nivel de ocupación del hotel fue otro indicador para motivar al administrador a tomar decisiones con el fin de reducir la capacidad ociosa del hotel. Algunos administradores al ver los resultados impulsaron campañas de tarifas reducidas.

El ahorro de horas hombre fue una realidad que el sistema evidenció. La cantidad de horas hombre ahorradas fue usado para hacer actividades de mayor valor para el negocio. De ese modo los procesos manuales fueron reemplazados por procesos automatizados (ver apéndice A)

La tasa de rotación del personal fue decreciendo notablemente. De un 61% en el 2015 se llegó a un 14% en el 2017.

A continuación, se muestran los indicadores mencionados por cada hotel.

Incremento de ventas.

Se puede apreciar que la mitad de los hoteles tuvo una caída de ventas en el segundo año. Sin embargo, a raíz de los indicadores obtenidos por el sistema los administradores pudieron advertir el problema y tomar las medidas correctivas para revertir los resultados en el siguiente año.

La fórmula empleada fue: sumatoria de monto total de la tabla “reserva_detalle”.

Tabla 26

Reporte de ventas anual entre el 2016 y 2018

	2016	2017	2018
SUNSET	350948	274215	349209 ↓
ALBAREDA	426288	485220	501440 ↑
QUEEN	348867	384587	409458 ↑
WASHINGTON	764917	910411	1017924 ↑
NORTE	713517	596670	532104 ↓
VICTORIA	612400	615763	973994 ↑
MOLINA	313110	296430	337640 ↑
MISTI	368000	334370	397865 ↑
FRONTERA	361080	358200	375463 ↑
GOCTA	447300	513800	559550 ↓

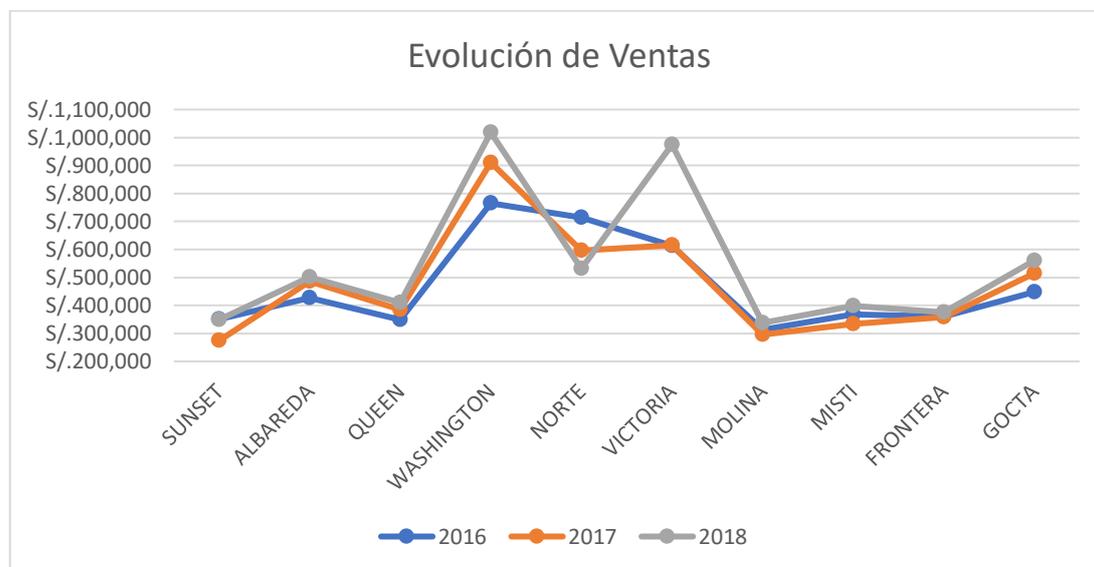


Figura 54. Gráfico de evolución de ventas en soles. Fuente: Xafiro.

Nivel de ocupación promedio.

El indicador de ocupabilidad fue importante porque permitió medir la capacidad ociosa que tenía el hotel. A medida que el índice se acercaba a 100% menor capacidad ociosa se tenía.

Según los resultados obtenidos la mitad de los hoteles elevó su nivel de ocupación en los siguientes años. La razón lo atribuyen, los administradores, a los reportes de gestión que pudieron advertir el problema.

La fórmula usada fue: $\# \text{Noches ocupadas} / (\# \text{Habitaciones} * \# \text{Noches disponibles})$

Tabla 27

Nivel de ocupabilidad anual entre el 2016 y el 2018

	2016	2017	2018	2018 vs 2016
SUNSET	65%	50%	59%	-9% ↓
ALBAREDA	50%	59%	62%	24% ↑
QUEEN	35%	38%	41%	17% ↑
WASHINGTON	48%	51%	51%	6% ↑
NORTE	72%	61%	59%	-18% ↓
VICTORIA	38%	37%	49%	29% ↑
MOLINA	59%	48%	62%	5% ↑
MISTI	46%	41%	43%	-7% ↓
FRONTERA	46%	44%	45%	-2% ↓
GOCTA	51%	59%	64%	25% ↑

Nota: El crecimiento promedio entre 2018 y 2016 fue de 7%

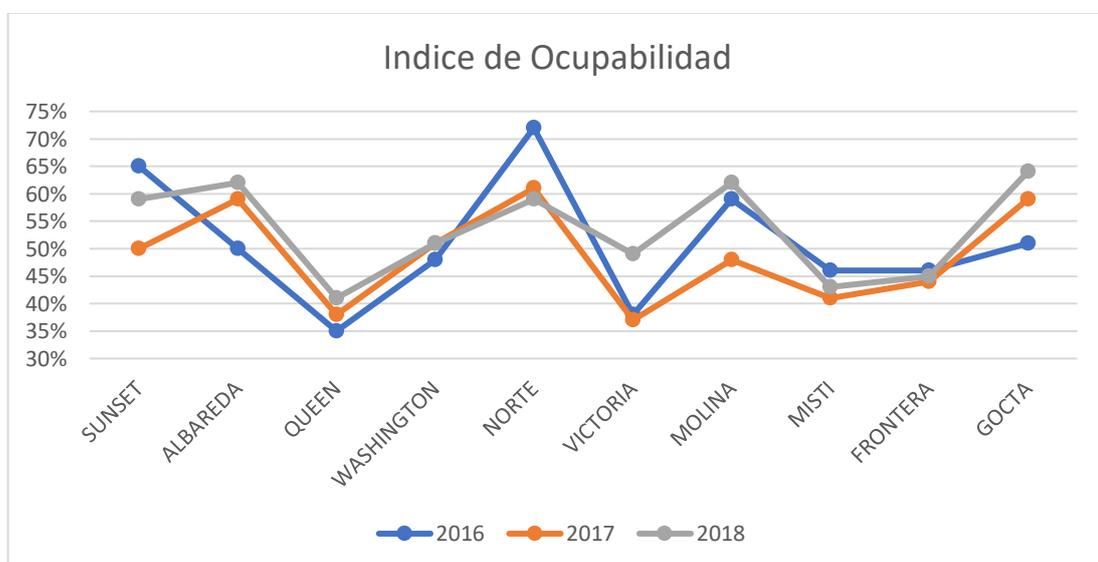


Figura 55. Índice de ocupabilidad por año. Fuente: Xafiro

Nivel de rotación de personal.

La tasa de rotación mostró una caída considerable en el último año. Mientras que en el 2015 el promedio ascendió a 61%, en el 2017 el promedio descendió a 14%. Mayormente los trabajadores renunciaban por los descuentos que tenían cuando la caja no cuadraba o cuando había pérdida de productos.

La mayoría de los hoteles mantuvo casi todo su personal en el último año, lo que permitió capacitarlos y especializarlo en sus funciones.

Tabla 28

Rotación del personal entre el 2015 y 2017

	Empleados actual	Reemplazos actual	TR actual	Reemplazos 2016	TR	Reemplazos 2017	TR
SUNSET	6	7	116%	4	67%	1	17%↓
ALBAREDA	7	8	114%	5	71%	2	29%↓
QUEEN	7	7	100%	6	86%	2	29%↓
WASHINGTON	10	5	50%	3	30%	1	10%↓
NORTE	8	2	25%	0	0%	0	0%↓
VICTORIA	10	1	10%	2	20%	0	0%↓
MOLINA	7	3	43%	4	57%	3	43%↓
MISTI	7	3	43%	2	29%	0	0%↓
FRONTERA	6	3	50%	2	33%	0	0%↓
GOCTA	6	4	67%	1	17%	1	17%↓

Nota: en el 2015 el promedio de la TR fue de 61% mientras que en 2017 fue de 14.5%

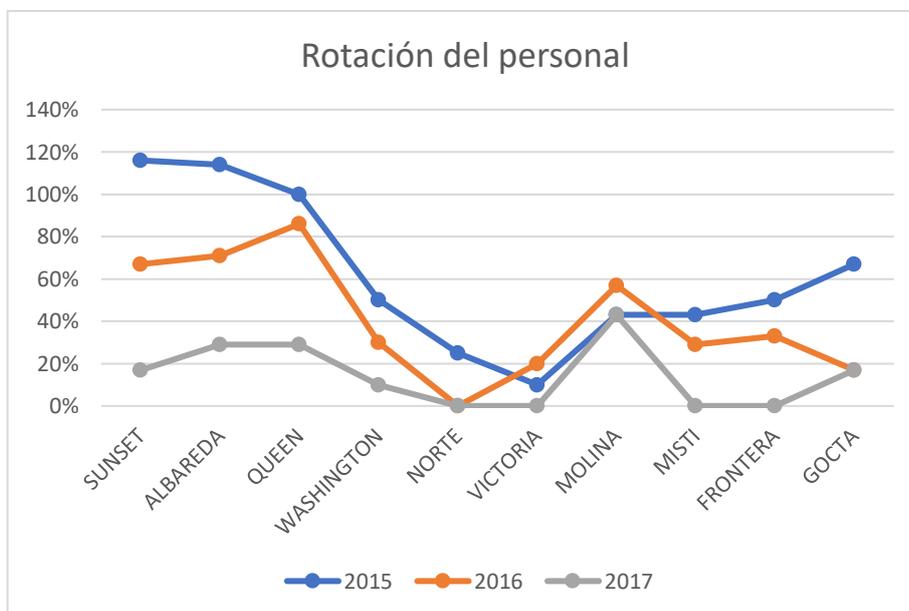


Figura 56. Tasa de rotación del personal. Fuente: Xafiro

Ahorro de tiempo en los registros de operaciones.

Durante las operaciones diarias del hotel se invertían varios minutos en las transacciones de reservas, control de inventarios y cierres de caja. En algunos casos se invertían 10, 20, 30 hasta 40 minutos mientras que usando XAFIRO se redujo ese tiempo hasta en un 80%.

Tabla 29

Ahorro de tiempo en registro de reservas por año.

	TIEMPO ACTUAL	TIEMPO XAFIRO	DIFERENCIA (MIN)	DIFERENCIA (HRS)	RESERVAS ANUAL PROMEDIO	AHORRO ANUAL(HRS)
SUNSET	11	2 (↓81%)	9	0.15	3386	508
ALBAREDA	9	3 (↓67%)	6	0.10	13092	1309
QUEEN	10	1 (↓90%)	9	0.15	8287	1243
WASHINGTON	8	2 (↓75%)	6	0.10	15306	1531
NORTE	8	2 (↓75%)	6	0.10	2235	224
VICTORIA	11	2 (↓81%)	9	0.15	6304	946
MOLINA	10	1 (↓90%)	9	0.15	7104	1066
MISTI	9	3 (↓67%)	6	0.10	2321	232
FRONTERA	8	2 (↓75%)	6	0.10	4123	412
GOCTA	6	3 (↓50%)	3	0.05	2666	133

Notas: El ahorro promedio de tiempo asciende a 75%

Tabla 30

Ahorro de tiempo en cierres de caja. Se cierre la caja 3 veces por día los 365 días

	TIEMPO ACTUAL	TIEMPO XAFIRO	DIFERENCIA (MIN)	DIFERENCIA (HRS)	AHORRO ANUAL (HRS)
SUNSET	23	2 (↓91%)	21	0.35	383
ALBAREDA	35	3 (↓91%)	32	0.53	584
QUEEN	26	2 (↓92%)	24	0.40	438
WASHINGTON	18	3 (↓83%)	15	0.25	274
NORTE	20	4 (↓80%)	16	0.27	292
VICTORIA	30	4 (↓87%)	26	0.43	475
MOLINA	40	5 (↓88%)	35	0.58	639
MISTI	16	2 (↓88%)	14	0.23	256
FRONTERA	20	3 (↓85%)	17	0.28	310
GOCTA	19	3 (↓84%)	16	0.27	292

Notas: El ahorro promedio de tiempo asciende a 87%

Tabla 31

Ahorro de tiempo en control de inventarios. Se controla el inventario 1 vez por día los 365 días

	TIEMPO ACTUAL	TIEMPO XAFIRO	DIFERENCIA (MIN)	DIFERENCIA (HRS)	AHORRO ANUAL(HRS)
SUNSET	30	5 (↓ 83%)	25	0.42	152
ALBAREDA	25	7 (↓ 72%)	18	0.30	110
QUEEN	35	8 (↓ 77%)	27	0.45	164
WASHINGTON	41	6 (↓ 85%)	35	0.58	213
NORTE	32	9 (↓ 72%)	23	0.38	140
VICTORIA	30	10 (↓ 67%)	20	0.33	122
MOLINA	45	7 (↓ 84%)	38	0.63	231
MISTI	34	8 (↓ 76%)	26	0.43	158
FRONTERA	31	10 (↓ 68%)	21	0.35	128
GOCTA	40	9 (↓ 78%)	31	0.52	189

Notas: El ahorro promedio de tiempo asciende a 76%

Tabla 32

Ahorro de tiempo total por año en horas

	TIEMPO RESERVAS	TIEMPO INVENTARIO	TIEMPO CIERRE CAJA	TIEMPO AHORRADO TOTAL (HRS)
SUNSET	508	383	152	1043
ALBAREDA	1309	584	110	2003
QUEEN	1243	438	164	1845
WASHINGTON	1531	274	213	2017
NORTE	224	292	140	655
VICTORIA	946	475	122	1542
MOLINA	1066	639	231	1936
MISTI	232	256	158	646
FRONTERA	412	310	128	850
GOCTA	133	292	189	614



Figura 57. Tiempo ahorrado por año. Fuente: Xafiro

Análisis Económico del cliente

Luego de comprobarse los ahorros de tiempo en los principales procesos del cliente se realizó una proyección económica. Para ello se usaron 3 herramientas financieras como el Valor Actual Neto (VAN), la Tasa Interna de Retorno (TIR) y el retorno de Inversión (ROI).

VAN y TIR

En el año 2016 el sueldo mínimo vital fue de 850 soles con una jornada laboral de 48 horas semanales. El costo/hora de mano de obra era de S/ 4.43

Para el cálculo del ingreso se multiplicaron las horas ahorradas anualmente (ver tabla 24) por el costo/hora de mano de obra. Mientras que para el cálculo del gasto se proyectó la tarifa mensual de Xafiro (ver tabla 14) a una tarifa anual. La diferencia de ingresos menos gastos dieron los flujos anuales por los próximos 5 años.

Los resultados mostraron que en 7 establecimientos el VAN y el TIR resultó positivo. Esto indicó que el proyecto era rentable y no tenía riesgos financieros. En los hoteles restantes, los indicadores salieron negativos, pero es necesario mencionar que dentro de los flujos anuales no se consideró el ahorro de costos de capacitación por reducción de la tasa de rotación.

Tabla 33

Cálculo del VAN y TIR para los establecimientos hoteleros

	AHORRO ANUAL	GASTO ANUAL	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5	VAN	TIR
SUNSET	S/. 4,617	S/. 3,000	-	1617	1617	1617	1617	1617	S/.6,130	161745%
ALBAREDA	S/. 8,867	S/. 3,000	-	5867	5867	5867	5867	5867	S/.22,241	586745%
QUEEN	S/. 8,168	S/. 3,000	-	5168	5168	5168	5168	5168	S/.19,590	516797%
WASHINGTON	S/. 8,929	S/. 3,000	-	5929	5929	5929	5929	5929	S/.22,476	592943%
NORTE	S/. 2,900	S/. 3,000	-	-100	-100	-100	-100	-100	-S/.381	-
VICTORIA	S/. 6,827	S/. 3,000	-	3827	3827	3827	3827	3827	S/.14,505	382656%
MOLINA	S/. 8,571	S/. 3,000	-	5571	5571	5571	5571	5571	S/.21,117	557083%
MISTI	S/. 2,860	S/. 3,000	-	-140	-140	-140	-140	-140	-S/.532	-
FRONTERA	S/. 3,763	S/. 3,000	-	763	763	763	763	763	S/.2,891	76302%
GOCTA	S/. 2,718	S/. 3,000	-	-282	-282	-282	-282	-282	-S/.1,069	-

Nota: solo se está considerando el ahorro de tiempo más no el ahorro en capacitación

ROI

El retorno de la inversión fue de un mes debido a dos razones. El PMS no tuvo costos de activación y si ben es cierto el pago del sistema era al inicio del mes, el administrador del hotel recibiría los beneficios(ahorro) al término del mes. Con relación al hardware todos los hoteles contaban con un computador que cumplía los requerimientos mínimos.

Conclusiones

El diagnóstico de necesidades fue clave para desarrollar este proyecto informático. Lo que a primera vista parecían problemas terminaron siendo síntomas del verdadero problema. Una actividad que favoreció este análisis fue realizar entrevistas con cada uno de los stakeholders.

El PMS demostró que se pudo reducir el tiempo en el proceso de reservas hasta una media del 75%. Esa ganancia de tiempo se debe a que con el sistema, consultar la disponibilidad de una habitación podría hacerse en contados segundos.

El PMS demostró que se pudo reducir el tiempo en el cierre de caja hasta una media del 87%. La razón principal fue que las sumatorias eran automáticas y toda la información estaba centralizada.

El PMS demostró que se pudo reducir el tiempo en el control de inventarios hasta una media del 76%. La razón principal fue que el cardex se generaba automáticamente, por lo que el usuario ya no tendría que hacer sumatorias de fechas pasadas.

El PMS demostró una reducción considerable de la tasa de rotación. De un índice del 61% en el 2015 se redujo a un 14.5% en el 2017.

En términos generales se pudo concluir que un PMS mejora la gestión de recursos y genera eficiencias en los procesos. Sin embargo, la explotación de sus indicadores depende de la capacidad de interpretación de los administradores. Es ahí donde interviene la otra parte de la solución que es la capacitación en temas de gestión.

Recomendaciones

Un sistema por si mismo no garantiza el éxito de la empresa, sino la interpretación de la información y la toma de decisiones que conlleva. Es recomendable que los administradores tengan una capacitación en marketing y gestión para perfeccionar el planeamiento estratégico de sus negocios.

Es necesario tener conocimiento de lo que el cliente final opina. Para poder confirmar que los ahorros de tiempo y dinero se están trasladando a una mejor experiencia del cliente, se podría desarrollar un cuestionario virtual que se integre con el sistema. De esta forma se podría tener retroalimentación del servicio prestado al cliente en tiempo real.

Se diseñó el proyecto para que pueda ser escalable y se le incorporen componentes como la gestión de relación con los clientes (CRM por sus siglas en inglés) o un Channel Manager para manejar todos los canales de venta desde Xafiro. Estos módulos generarían más valor agregado para los hoteles.

El futuro del PMS Xafiro está en la inteligencia de negocios. El modelo de negocio debería apuntar a proveer el sistema con precios mínimos a la mayor cantidad de hoteles, de ese modo se podría tener una gran base de datos para analizar patrones de consumo, perfiles de clientes y segmentación. En lugar de vender un software se vendería información enriquecida.

Referencias

- ÁLVAREZ, C. (2017). Software de Gestión Hotelera – Un adecuado funcionamiento [versión electrónica]. Revista Hostelpro, 24, 28. Balsa, L. &
- GARCÍA, I. (2016). Software de Gestión Hotelera – Gestionar de manera eficiente [versión electrónica]. Revista Hostelpro, 18, 26 – 27.
- BALSA, L. (2017). Software de Gestión Hotelera – Un adecuado funcionamiento [versión electrónica]. Revista Hostelpro, 24, 26
- AXELOS (2019). ITIL® Foundation, ITIL 4 edition. The Stationery Office
- AMAZON WEB SERVICES (2016). AWS nombrado líder en el informe de Magic Quadrant en referencia a la infraestructura como servicio (IaaS) por sexto año consecutivo*. Extraído de <https://aws.amazon.com/es/resources/gartner-2016-mq-learn-more/>
- AMAZON WEB SERVICES (2019). Amazon Compute Service Legal Agreement. Extraído de <https://aws.amazon.com/compute/sla/>
- AMAZON WEB SERVICES (2019). Tipos de Instancia EC2 Amazon. Extraído de <https://aws.amazon.com/ec2/instance-types/>
- CIOPERU (17.04.2018) Gartner: Los ingresos de nube pública crecerán 21,4% en el 2018. Perú. Recuperado de <https://cioperu.pe/articulo/25644/gartner-los-ingresos-de-nube-publica-creceran-214-en-el-2018/>
- CLOUD SECURITY GUIDANCE (2017). Security Guidance v4.0. Extraído de <https://cloudsecurityalliance.org/artifacts/security-guidance-v4/>
- DIARIO GESTION (13.12.2016). ¿Por qué no logra despegar la creación de mypes vía online? Perú. Recuperado de <https://gestion.pe/economia/empresas/logra-despegar-creacion-mypes-via-online-123483>
- DIARIO GESTION (12.03.2017) Pymes cierran más contratos por software de gestión de procesos. Perú. Recuperado de <https://gestion.pe/economia/pymes-cierran-contratos-software-gestion-procesos-130479>
- EL PERUANO (07.01.2016) Modifican el Reglamento General de Calidad de los Servicios Públicos de Telecomunicaciones. Extraído de <https://busquedas.elperuano.pe/normaslegales/modifican-el-reglamento-general-de-calidad-de-los-servicios-resolucion-no-005-2016-cdosiptel-1335063-1/>
- EL PERUANO (29.08.2018) Nueva Era de la Fibra Óptica. Extraído de <https://elperuano.pe/suplementosflipping/economika/279/web/pagina02.html>
- GRANT THORNTON (2018) Emerging clouds in hotel technology. Reino Unido. pp.2-3
- IBM (s.f). Arquitecturas de tres niveles. Extraído de https://www.ibm.com/support/knowledgecenter/es/SS7K4U_9.0.5/com.ibm.websphere.zseries.doc/ae/covr_3-tier.html
- INTELLIPAAT (s.f) AWS vs Azure vs Google – Detailed Cloud Comparison. Extraído de

<https://intellipaat.com/blog/aws-vs-azure-vs-google-cloud/#establishment>

INTERNET EXPERTS (s.f). Mbps vs. MBps: What's the Difference? Extraído de <https://www.attinternetservice.com/resources/mbps-vs-mbps/>

KIYAMU A. (2018) La calidad de diseño de un Property Management System y la utilidad en hoteles tres estrellas de Miraflores. USIL

LAZARTE RUBEN (s.f.) ¿Cuándo Utilizar Metodología Ágil y Cuándo Cascada? PMI. Extraído de <https://www.pmi.org.pe/cuando-utilizar-metodologia-agil-cuando-cascada>

LA REPLUBLICA (21.09.2017), El 17% de las Pymes en el Perú tiene una página web. Perú. Recuperado de <https://larepublica.pe/economia/1101430-el-17-de-las-pymes-en-el-peru-tiene-una-pagina-web>

MARTÍNEZ J, MAJÓ J Y CASADESÚS (2006) M. El uso de las tecnologías de la información en el sector hotelero. ResearchGate.

MICROSFOT (s.f). Regiones de Azure. Extraído de <https://azure.microsoft.com/es-es/global-infrastructure/regions/>

MINCETUR (2016). Medición económica del Turismo. Perú. Recuperado de https://www.mincetur.gob.pe/wpcontent/uploads/documentos/turismo/publicaciones/medicion_economica_turismo_alta.pdf

MINCETUR (2018). Arribo, pernотaciones y oferta hotelera en establecimientos de hospedaje. Recuperado de <http://datosturismo.mincetur.gob.pe/appdatosTurismo/Content3.html>

OOKLA (2016) Ookla reconoció a Claro como el operador que ofrece el internet fijo más rápido del Perú. Extraído de https://www.claro.com.pe/recursos_contenido/institucional/a_c28ed3c-48c8-40d3_-8d36-df4161f_d4592-Claro-recibe-reconocimiento-por-ofrecer-el+internet-fijo-m as-rapido-del-Peru.pdf

ORACLE (s.f) Entrega de experiencias inolvidables para los clientes con Hotel PMS. Extraído de <https://www.oracle.com/cl/industries/hospitality/what-is-hotel-pms.html>

PALAO DAVID (2018) PMS: pieza clave en la gestión de un hotel. Tecnohotel, España

PONCE FÁTIMA & ZEVALLOS EMILIO (2017) La innovación en la micro y la pequeña empresa: no solo factible, sino accesible. PUCP. Recuperado de <http://revistas.pucp.edu.pe/index.php/360gestion/article/view/19050>

PRODUCE (2017). Más de 1,9 millones de Mipyme formales operan en el mercado peruano al 2017. Recuperado de <http://ogeiee.produce.gob.pe/index.php/shortcode/estadistica-oe/estadisticas-mipyme>

RENUEVO MARCOS (2016) Por qué tener un pms en tu hotel. Quoinet. España

RIVERA RODRIGO (2014). Perspectivas sobre tecnología y crecimiento económico en PYMEs. Boston Consulting Group. Estados Unidos. Recuperado de <https://www.cepal.org/sites/default/files/news/files/52658-01-Rodrigo-Rivera-BCG-.pdf>

SOFTWARE ENGINEERING INSTITUTE (2019). Architecting Software the SEI Way – Software Architecture Fundamentals: Technical, Business, and Social Influences. Extraído de <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=21534>

WINSTON W. ROYCE (1970). Modelo Waterfall o en Cascada. Gestionando el desarrollo de grandes sistemas de software. Digital Talent Agency.

Bibliografía

- ADSLZONE (05.07.2018) Esta es la velocidad mínima de Internet recomendada para una Smart TV. Extraído de <https://www.adslzone.net/2018/07/05/velocidad-minima-smart-tv/>
- BENTO M. ALBERTO & AGGARWAL K. ANIL (2013) Cloud computing Service and Deployment models. Estados Unidos: Universidad de Baltimore
- CANATUR (2019). Hoteles de Perú se recuperan este año en niveles de ocupación y tarifas. Perú. Recuperado de <http://www.portaldeturismo.pe/noticia/hoteles-de-peru-se-recuperan-este-ano-en-niveles-de-ocupacion-y-tarifas>
- DIARIO GESTION (12.06.2014) Un repunte económico en el Perú haría prevalecer las inversiones en software tecnológico. Extraído de <https://gestion.pe/tecnologia/repunte-economico-peru-haria-prevalecer-inversiones-software-tecnologico-62747-noticia/>
- DZONE (09.01.2018). EC2 Network Performance Cheat Sheet . Extraído de <https://dzone.com/articles/ec2-network-performance-cheat-sheet>
- EAE BUSINESS SCHOOL (2019). ¿Qué es un sistema CRM?. España. Recuperado de <https://www.eaeprogramas.es/empresa-familiar/que-es-un-sistema-crm>
- EVALUANDOCLOUD.COM (07.07.2015) Cloud Computing en América Latina. <https://evaluandocloud.com/cloud-computing-en-america-latina/>
- GENTILE A. JOSÉ (2016) Competitividad Hotelera 1a ed. Mar de Plata - Argentina. Amazon Kindle
- GÓMEZ Q. ROBERTO (2014). ¿Qué es un "full-stack developer"?. BBVA API_Market. España. Recuperado de <https://bbvaopen4u.com/es/actualidad/que-es-un-full-stack-developer>
- IBM (s.f). Integración de Software RUP. Extraído de https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/guidances/conceptos/software_integration_2F85C9B0.html#
- INEI (2013). Análisis de la estructura empresarial de Lima Metropolitana. Recuperado de https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1178/cap04.pdf
- INEI (2018). Demografía Empresarial en el Perú. Recuperado de https://www.inei.gob.pe/media/MenuRecursivo/boletines/02-informe-tecnico-n-02-demografia-empresarial-i-trim2018_may2018.pdf
- JEFFREY R. YOST (2017). *Making It Work: A History of the computer services industry*. Massachusetts – Estados Unidos. The MIT Press. Recuperado de <http://shorturl.at/DNP9>
- KATZ RAÚL (2015). El ecosistema y la economía digital en América Latina. Madrid – España. Fundación Telefónica

LOUIE ANDRE (2018). 2018 SaaS Industry Market Report: Key Global Trends & Growth Forecasts. FinancesOnline. Estados Unidos. Recuperado de <https://financesonline.com/2018-saas-industry-market-report-key-global-trends-growth-forecasts/>

MILLAN GARCIA, GOMEZ DIAZ (2018). Factores e indicadores de competitividad hotelera. Venezuela. Recuperado de <https://www.redalyc.org/jatsRepo/880/88055200021/html/index.html>

MARIA LUISA SAAVEDRA GARCIA (2012). Una propuesta para la determinación de la competitividad en la pyme latinoamericana. Barranquilla – Colombia. Recuperado de http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1657-62762012000200005

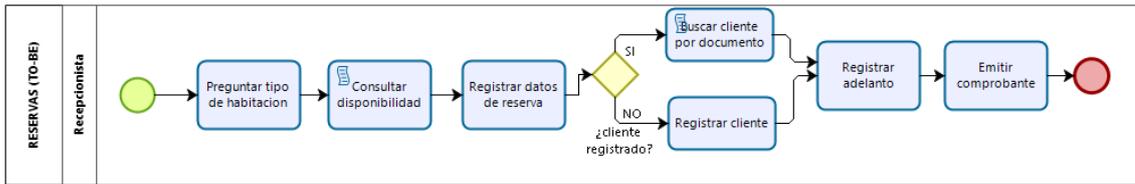
PQS (28.03.2016) Facturación electrónica: conoce sus beneficios y características. Perú. Recuperado de <https://www.pqs.pe/economia/facturacion-electronica-beneficios>

SALESFORCE (2018). ¿Qué es Cloud Computing? Estados Unidos. Recuperado de <https://www.salesforce.com/mx/cloud-computing/>

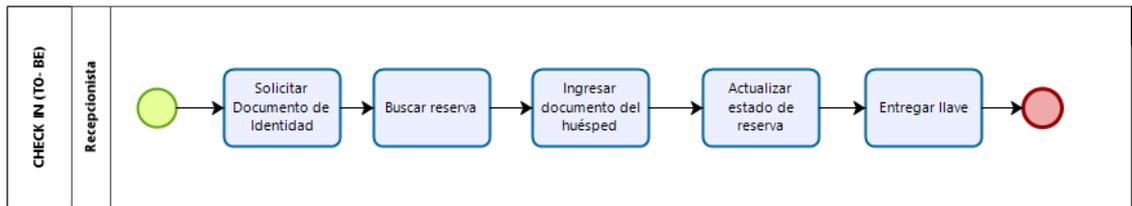
SUNAT (2018). Características de la micro y pequeñas empresas. Perú. Recuperado de <http://www.sunat.gob.pe/orientacion/mypes/caracteristicas-microPequenaEmpresa.html>

Apéndice A

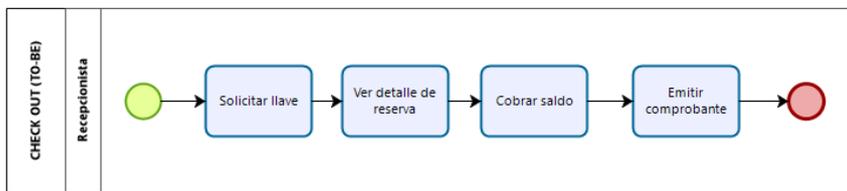
Modelamiento de procesos de negocio (To-be)



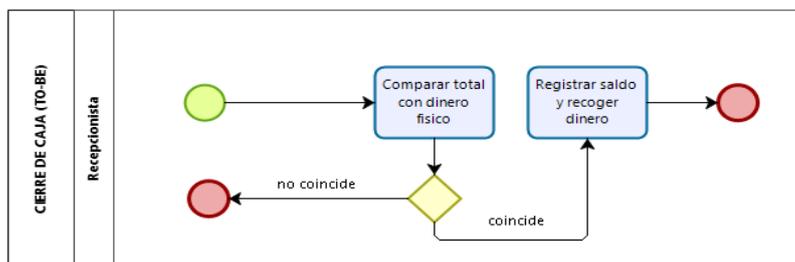
Proceso de reservas usando el PMS. Elaboración propia



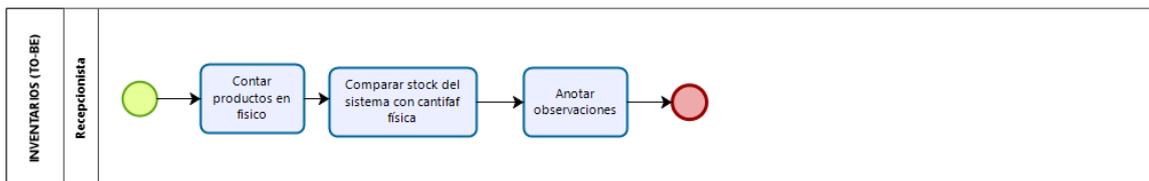
Proceso de checkin usando el PMS. Elaboración propia



Proceso de checkout usando el PMS. Elaboración propia



Proceso de cierre de caja usando el PMS. Elaboración propia



Proceso de gestión de inventarios usando el PMS. Elaboración propia