PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

DESARROLLO DE UN SISTEMA PARA LA AUTOMATIZACIÓN DE PROCESOS EN HOSTALES DEL CENTRO HISTÓRICO DE QUITO.

Autor: MARCELO ALEXANDER ESPARZA MONTES

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

Director: ING. DAMIÁN NICOLALDE

QUITO, 2018

Dedicatoria

A mi madre

Por creer en mí siempre, por darme un apoyo incondicional durante toda mi carrera profesional y por siempre darme consejos sabios que fueron de gran ayuda para nunca rendirme y alcanzar el éxito.

A mi padre

Por haberme criado con un carácter fuerte pero noble, para nunca dejarme caer ante cualquier situación difícil. Gracias a todo el apoyo económico que recibí de su parte gracias a su arduo trabajo siempre fue una inspiración a ser cada día un mejor profesional.

A mi hermano

Gracias al éxito que vi en su vida profesional fue una gran motivación para superarme. Fue de gran ayuda en muchos aspectos, y en otras ocasiones se opuso a brindarme las facilidades, que hicieron que me esforzara aún más por conseguir las metas que me propuse.

A mi primo

Desde pequeño tuve gran admiración hacia mi primo mayor por su gran personalidad y sobre todo el conocimiento del área de sistema que poseía, por lo que fue de gran motivación para mí al seleccionar este camino profesional.

A mi mentor, Andrés Alban

Además de mentor, profesor, y jefe, se convirtió en un gran amigo de quien aprendí mucho. La mayor parte del conocimiento que él me enseño fueron aplicadas en el presente trabajo de disertación. Además de haber inculcado muchos valores en mí, siempre fue un gran apoyo en estos últimos 2 años de vida profesional y laboral.

Al presidente de la empresa Yes Consultancy, Frank Patiño

Por darme las facilidades para poder completar mi carrera y acabar con el presente trabajo de disertación. Todo el conocimiento, motivación, dedicación y responsabilidad que he aprendido de él fue de gran ayuda en el desarrollo del presente trabajo de disertación.

Agradecimientos

Agradezco a la vida por permitirme alcanzar todas las metas que me he propuesto y por haber puesto en mi camino personas con tan buen corazón y conocimiento como mi círculo de amistades en la universidad, mis profesores, mi primer jefe en la primera empresa que trabajé y sobre todo al gran apoyo de mis padres, en todo aspecto que nunca permitieron que me rindiera por más difícil que fuera la situación estuvieron ahí para ayudarme a seguir adelante y cumplir con todos mis objetivos.

A el Ing. Damián Nicolalde, director del presente trabajo de disertación, por toda la colaboración y apoyo brindado durante todo el desarrollo del trabajo de disertación.

Índice de Contenidos

CAPÍTULO 1 ..... **1**

1. INTRODUCCIÓN 1

1.1 Justificación *1*

1.2 Planteamiento del problema *2*

1.3 Objetivos *3*

1.3.1 Objetivo General 3

1.3.2 Objetivos Específicos 3

1.4 Antecedentes *4*

CAPÍTULO 2 ..... **6**

2. FUNDAMENTOS TEÓRICOS 6

2.1 Hostales del Centro Histórico de Quito *6*

2.1.1 Gestión de los hostales 6

2.1.2 Tipos de hostales 7

2.1.3 Servicios 7

2.1.4 Publicidad y marketing 8

2.1.5 Segmento de mercado 10

2.2 Procesos internos de los hostales del Centro Histórico de Quito *11*

2.2.1 Procesos del área de recepción 11

2.2.2 Procesos del área de limpieza y mantenimiento 12

2.2.3 Procesos del área de bar 13

2.2.4 Procesos de inventariado 13

2.2.5 Procesos del área de contabilidad 14

2.3 Metodología de desarrollo del sistema *15*

2.3.1 Metodologías Ágiles 15

2.3.2 Definición de Extreme Programming 16
2.3.3 Aplicación de la metodología en el sistema 20
2.3.4 Desarrollo por capas y consumo de datos 22
2.4 Herramientas para el desarrollo del sistema 23
2.4.1 PowerDesigner 23
2.4.2 Mockup Builder 24
2.4.3 Base de datos MySQL y MySQL WorkBench 24
2.4.4 Servidor Apache XAMPP 25
2.4.5 Lenguajes de programación 26
2.4.5.1 HTML5 26
2.4.5.2 CSS3 27
2.4.5.3 JavaScript 28
2.4.5.4 PHP 30
2.4.5 Sublime Text 32
2.4.6 Framework 32
2.4.6.1 Laravel 32
CAPÍTULO 3 35
3. DESARROLLO DEL SISTEMA 35
3.1 Primera Iteración 36
3.1.1 Planificación 36
3.1.2 Diseño 40
3.1.3 Codificación 44
3.1.4 Pruebas 56
3.2 Segunda Iteración 58
3.2.1 Planificación 58
3.2.2 Diseño 61
3.2.3 Codificación 67

3.2.4 Pruebas 78

3.3 Tercera Iteración *82*

3.3.1 Planificación 82

3.3.2 Diseño 85

3.3.3 Codificación 90

3.3.4 Pruebas 98

CONCLUSIONES Y RECOMENDACIONES 103

CONCLUSIONES 103

RECOMENDACIONES 104

BIBLIOGRAFÍA 106

Índice de Figuras

Ilustración 1 Arquitectura de Laravel (Esparza, 2018) 36

Ilustración 2 Historia de Usuario 1 (Esparza, 2018) 37

Ilustración 3 Historia de Usuario 2 (Esparza, 2018) 38

Ilustración 4 Historia de Usuario 3 (Esparza, 2018) 39

Ilustración 5 Prototipo de Autenticación para el Sistema (Esparza, 2018) 40

Ilustración 6 Tarjeta CRC de un Empleado del hostal (Esparza, 2018) 41

Ilustración 7 Prototipo Opciones Generales del Sistema (Esparza, 2018) 42

Ilustración 8 Prototipo Listado de Habitaciones (Esparza, 2018) 43

Ilustración 9 Prototipo CRUD Insert y Edit de Habitaciones (Esparza, 2018) 43

Ilustración 10 Prototipo Delete del CRUD de Habitaciones (Esparza, 2018) 44

Ilustración 11 Comando de creación de proyecto con Laravel Framework (Esparza, 2018) 45
Ilustración 12 Modelo Físico del Diagrama de Clases del Sistema (Esparza, 2018) 46
Ilustración 13 Conexión a la base de datos en archivo database.php (Esparza, 2018).....47
Ilustración 14 Conexión a la base de datos en archivo .env (Esparza, 2018).....47
Ilustración 15 Creación de un Modelo en Laravel (Esparza, 2018).....48
Ilustración 16 Creación de un Controlador en Laravel (Esparza, 2018).....48
Ilustración 17 Creación de Autenticación de Laravel por CMD (Esparza, 2018) 49
Ilustración 18 Creación de Tablas de Usuarios de Laravel por CMD (Esparza, 2018) 49
Ilustración 19 Interfaz de Usuario Página de Autenticación (Esparza, 2018) 49
Ilustración 20 Vista de la Pantalla principal del Sistema (Esparza, 2018) 51
Ilustración 21 Función “Listado” de HabitacionController (Esparza, 2018).....52
Ilustración 22 Vista de la Pantalla de Listado de Habitaciones (Esparza, 2018) 52
Ilustración 23 Vista de la Pantalla de Nueva Habitación (Esparza, 2018).....54
Ilustración 24 Vista de la Pantalla de Editar Habitación (Esparza, 2018).....54
Ilustración 25 Vista de la Pantalla de Papelera de Habitaciones (Esparza, 2018).....56
Ilustración 26 Pruebas Caja Blanca Sin errores (Esparza, 2018).....57
Ilustración 27 Pruebas Caja Negra Primer Prueba (Esparza, 2018) 58
Ilustración 28 Historia de Usuario 4 (Esparza, 2018).....59
Ilustración 29 Historia de Usuario 5 (Esparza, 2018).....60
Ilustración 30 Prototipo Check-in / Check-out (Esparza, 2018).....62
Ilustración 31 Prototipo Proceso Check-in paso 1 (Esparza, 2018) 62
Ilustración 32 Prototipo Proceso Check-in, Sin Reserva (Esparza, 2018).....63
Ilustración 33 Prototipo Proceso Check-in, Sin Reserva, Cliente Frecuente (Esparza, 2018).....63

Ilustración 34 Prototipo Proceso Check-in, Sin Reserva, Cliente Nuevo (Esparza, 2018) 64

Ilustración 35 Prototipo Proceso Check-in, Acompañantes (Esparza, 2018) 64

Ilustración 36 Prototipo Proceso Check-in, Con Reserva (Esparza, 2018) 65

Ilustración 37 Prototipo Proceso Check-in, Selección de Habitaciones (Esparza, 2018).....65

Ilustración 38 Prototipo Proceso Check-out (Esparza, 2018) 66

Ilustración 39 Prototipo Proceso Check-in / Check-out Listado de Huéspedes (Esparza, 2018)..66

Ilustración 40 Código Controlador Pantalla principal Check-in / Check-out (Esparza, 2018) 67

Ilustración 41 Pantalla principal Check-in / Check-out (Esparza, 2018).....68

Ilustración 42 Pantalla Check-in Con o Sin Reserva (Esparza, 2018).....71

Ilustración 43 Pantalla Check-in Listado de Reservas (Esparza, 2018) 71

Ilustración 44 Pantalla Check-in Cliente Nuevo o Frecuente (Esparza, 2018) 72

Ilustración 45 Pantalla Check-in Listado de Clientes Frecuentes (Esparza, 2018) 72

Ilustración 46 Pantalla Check-in Acompañantes (Esparza, 2018).....73

Ilustración 47 Pantalla Check-in Cliente Nuevo (Esparza, 2018) 73

Ilustración 48 Pantalla Check-in Selección de Habitaciones (Esparza, 2018) 74

Ilustración 49 Pantalla Check-in Fechas de estadía (Esparza, 2018).....74

Ilustración 50 Pantalla Check-in Realizado Correctamente (Esparza, 2018) 75

Ilustración 51 Pantalla Check-out Realizado Correctamente (Esparza, 2018).....75

Ilustración 52 Pantalla principal Reservas (Esparza, 2018).....78

Ilustración 53 Pantalla Listado de Reservas (Esparza, 2018).....78

Ilustración 54 Pruebas Check-out Caja Blanca Sin errores (Esparza, 2018).....79

Ilustración 55 Pruebas Check-in Caja Blanca Sin errores (Esparza, 2018).....79

Ilustración 56 Pruebas Listado de Huéspedes Caja Blanca Sin errores (Esparza, 2018) 80

Ilustración 57 Pruebas Listado de Reservas Caja Blanca Sin errores (Esparza, 2018) 80
Ilustración 58 Pruebas Caja Negra Segunda Prueba (Esparza, 2018) 81
Ilustración 59 Historia de Usuario 6 (Esparza, 2018) 83
Ilustración 60 Historia de Usuario 7 (Esparza, 2018) 84
Ilustración 61 Prototipo pantalla principal Modulo Bar (Esparza, 2018) 85
Ilustración 62 Prototipo Pantalla Compra de Productos (Esparza, 2018) 86
Ilustración 63 Prototipo Pantalla Compra de Nuevo Producto (Esparza, 2018) 86
Ilustración 64 Prototipo Pantalla Venta de Productos (Esparza, 2018) 87
Ilustración 65 Prototipo Pantalla Stock de Productos (Esparza, 2018) 87
Ilustración 66 Prototipo Pantalla Inventario (Esparza, 2018) 88
Ilustración 67 Prototipo Pantalla Selección de Cliente a Facturar (Esparza, 2018) 89
Ilustración 68 Prototipo Pantalla Factura (Esparza, 2018) 89
Ilustración 69 Pantalla Stock (Esparza, 2018) 92
Ilustración 70 Pantalla Inventario (Esparza, 2018) 93
Ilustración 71 Pantalla de Compra de Productos (Esparza, 2018) 94
Ilustración 72 Pantalla Compra de Nuevo Producto (Esparza, 2018) 94
Ilustración 73 Pantalla Venta de Productos (Esparza, 2018) 95
Ilustración 74 Pantalla Facturación - Clientes hospedados (Esparza, 2018) 96
Ilustración 75 Pantalla Factura (Esparza, 2018) 97
Ilustración 76 Impresión de Factura (Esparza, 2018) 98
Ilustración 77 Pruebas Stock Caja Blanca Sin errores (Esparza, 2018) 99
Ilustración 78 Pruebas Inventario Caja Blanca Sin errores (Esparza, 2018) 99
Ilustración 79 Pruebas Compra de Productos Caja Blanca Sin errores (Esparza, 2018) 100

Ilustración 80 Pruebas Venta de productos Caja Blanca Sin errores (Esparza, 2018).....100

Ilustración 81 Pruebas Factura Caja Blanca Sin errores (Esparza, 2018) 101

Ilustración 82 Pruebas Caja Negra Tercera Prueba (Esparza, 2018).....102

Capítulo 1

1. Introducción

En este capítulo se va a describir la justificación, el planteamiento del problema, los objetivos generales y específicos, los antecedentes que existen del tema y el alcance del presente trabajo de disertación de grado.

1.1 Justificación

El funcionamiento de los hostales a nivel general es similar a hoteles, con la diferencia que los hoteles manejan un flujo de información considerable con procesos de mayor complejidad debido a todos los servicios que ofrecen, mientras que un hostal maneja de manera más simple los procesos por el tipo de servicios que estos ofrecen, pese a que los servicios de los hostales son similares a los de un hotel. Sin embargo, los servicios que los hostales ofrecen a los huéspedes como cafetería, Check-in, Check-out, entre otros, los gestionan de forma manual, registrando información en agendas, documentos de Word, Excel, etc. o utilizan sistemas básicos desarrollados en Excel que de alguna manera ayudan a manejar la información pero no es una forma segura tenerla archivada ya que es relevante para el establecimiento; a diferencia de los hoteles que si cuentan con un sistema que automatiza los procesos de su establecimiento.

Aunque, como se mencionó anteriormente, los servicios que un hostal ofrece son similares a los de un hotel, los sistemas que los hoteles utilizan tienen muchas funcionalidades que quedarían inútiles para un hostal, y a más de esto, aquellos sistemas hoteleros son costosos y de mayor complejidad, por esta razón se desarrollará un sistema que se ajuste a las necesidades de los hostales.

1.2 Planteamiento del problema

Se plantea el tema de disertación de grado puesto que, los hostales del centro histórico de Quito no disponen de un sistema que ayude a gestionar los procesos de los servicios que ofrecen. Por lo que suelen tener inconvenientes con llevar la contabilidad, el estado de las habitaciones y a nivel general los servicios que ofrecen con un orden adecuado. Esto ocasiona que al final de cada mes el balance, la cantidad de veces que las habitaciones fueron rentadas y los productos o servicios que se vendieron no tengan concordancia con el valor real en la caja. Por las razones mencionadas se busca desarrollar un sistema que se acople a las necesidades de los hostales del centro de Quito para que puedan tener automatizados los procesos de dirección con el fin de llevar un orden de toda la administración o funcionamiento del hostal.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema para la automatización de procesos de los servicios que brindan los hostales del Centro Histórico de Quito.

1.3.2 Objetivos Específicos

1. Definir los requerimientos del sistema en base a los servicios más importantes que ofrecen los hostales del centro histórico de Quito para su automatización.

2. Diseñar el sistema propuesto que cumpla las necesidades para automatizar los procesos de un hostal del centro histórico de Quito.

3. Codificar el sistema propuesto para la automatización de procesos viables seleccionados para un hostal del centro histórico de Quito para que pueda mantener seguro y en orden su establecimiento.

4. Realizar pruebas de caja blanca del sistema codificado para verificar que el flujo de información sea el correcto y el más optimo posible.

5. Realizar pruebas de caja negra del sistema codificado para verificar que las salidas que el sistema entrega son las que el usuario necesita.

1.4 Antecedentes

Las casas en el centro histórico de Quito son generalmente populares y tienen una estructura y apariencia de antigüedad peculiar. Esto se debe a que las casas antiguas de este sector de quito que fueron conservadas con esa apariencia durante una gran cantidad de años a lo largo de la historia por esta razón en el año 1978 fue declarado patrimonio cultural.

En la actualidad la gran mayoría de las casas dejaron de ser utilizadas para vivienda como tal. Los dueños o arrendatarios aprovecharon la infraestructura de estas para adecuar un negocio comúnmente conocido como hostales.

Weissinger (2001) afirma: “Nada ha hecho parar la creciente espiral del negocio hotelero. Hoy día, el sector es sumamente diverso y ofrece alojamiento para satisfacer a casi cualquier tipo de viajero”

Debido a la gran cantidad de turistas que se tiene en esta zona es un negocio muy rentable y además de eso ofrecen un ambiente colonial a los viajeros para que sientan la esencia de Quito desde la antigüedad.

Ahora bien, hablando de antiguo es totalmente distinto a anticuado, y allí es donde radica el problema que los hostales tienen hoy en día al no poseer un sistema de automatización de procesos. Muchas veces los dueños creen que por ser patrimonio cultural no pueden

usar este tipo de sistemas en sus negocios, pero están interpretando mal el concepto de tecnología ya que el sistema que se propone implementar en uno de los hostales del centro histórico de Quito no cambia en lo absoluto a la infraestructura del establecimiento como tal.

Capítulo 2

2. Fundamentos Teóricos

En el presente capítulo se va a describir todos los conceptos que se requieren entender para el presente trabajo de disertación de grado. Se detallarán desde los conceptos de la lógica del negocio a la cual está orientado el sistema que se va a desarrollar hasta los conceptos fundamentales sobre la programación web y herramientas que se usaran para el desarrollo del sistema.

2.1 Hostales del Centro Histórico de Quito

2.1.1 Gestión de los hostales

El ministerio de turismo de la república del Ecuador categoriza en 6 a todos los negocios que ofrezcan servicios de alojamiento, los cuales son los Hoteles, Hostales, Resorts, Hosterías y Hacienda turística. Cada uno de ellos tiene un rango de estrellas que pueden poseer según su ranking en base a los servicios que ofrecen. El ministerio de turismo es el encargado de definir en qué categoría entra cada hostal del país.

La gestión o administración de un hostal se asemeja a la de los hoteles a menor escala por la diferencia de dimensión de los dos tipos establecimientos. Sin embargo, siguen siendo los mismos procesos y funciones que cumplen con el objetivo de mantener el negocio funcionando armónicamente en todas sus áreas de trabajo. Para alcanzar este objetivo todo negocio de este tipo sigue cuatro fases esenciales para procesos administrativos que son: planificación, organización, dirección y control.

2.1.2 Tipos de hostales

Según el reglamento de alojamiento turístico emitido por el Ministerio de Turismo; Los hostales se dividen en tres categorías según el número de estrellas que tenga cada establecimiento pudiendo tener desde una hasta tres estrellas como máximo. Para alcanzar a ser considerado un hostal privilegiado y reconocido por el ministerio de turismo, dependiendo a que categoría pertenece, debe cumplir ciertos parámetros detallados en el anexo B del acuerdo ministerial 24.

2.1.3 Servicios

Los servicios que ofrecen este tipo de negocios en el centro histórico de Quito pueden variar, sin embargo, existen algunos que no son opcionales, es decir, es una obligación ofrecerlos para que el ministerio de turismo apruebe que el negocio puede operar sin problema. Entre los servicios que un hostal debe ofrecer a sus huéspedes están poseer

una ruta de evacuación en todas las habitaciones, ofrecer servicio de limpieza, tener los servicios básicos en correcto funcionamiento, etc.

Ahora bien, para estar en una de las categorías antes mencionadas muchas veces depende de los servicios que el negocio ofrezca, sin embargo, estamos hablando del centro histórico de Quito, lo que implica que hay normas y leyes con ciertas restricciones respecto a hacer una modificación o adecuación de arquitectura y/o infraestructura de las construcciones a nivel general del centro ya que este es considerado patrimonio cultural.

2.1.4 Publicidad y marketing

Es necesario que todo hostal del centro sea nuevo o viejo en el mercado, se dé a conocer por que es la clave de todo negocio para promover la demanda y los ingresos de este. Además, con el estudio realizado a los huéspedes extranjeros en el hostal Latitud ECUEM se concluye que la gran cantidad de turistas que arriban al Ecuador, desean visitar uno de los sectores más conocidos en nuestra ciudad de Quito a nivel internacional, El Centro Colonial. Para la publicidad y marketing de este tipo de negocios existen dos maneras principales de cómo hacerlo, las cuales se mencionan a continuación.

Marketing tradicional

La publicidad tradicional básicamente se refiere a todas las técnicas utilizadas por todo negocio para darse a conocer antes de la aparición del internet. De entre las técnicas más conocidas están el volanteo, las pancartas, comerciales en televisión y la radio. Por estos y otros medios sin incluir el internet, son los utilizados para hacer publicidad de un negocio de forma tradicional, llegando a una gran cantidad de personas.

E-Marketing o Marketing digital

El marketing digital se refiere a la misma publicidad que se hace en la tradicional, pero mediante el internet únicamente. La gran ventaja que conlleva usar este innovador medio es que gracias a las herramientas que contiene podemos segmentar el mercado de manera inteligente y llegar a las personas que realmente pueden llegar a ser clientes efectivos del producto o servicio que el negocio ofrezca, es decir, enfocarse en el mercado objetivo.

Gracias a la gran cantidad de facilidades que el mundo digital ofrece, se puede implementar un sin número de estrategias de marketing para conseguir el objetivo de todo negocio, en este caso vender una estadía en un hostal. Dentro de las estrategias que se pueden implementar se puede escoger en que plataforma hacerlo, ya sea un sitio web, redes sociales, un blog, etc. dependiendo de la necesidad del negocio que esté deseando implementar el marketing digital para atraer a los consumidores a su negocio.

2.1.5 Segmento de mercado

Segmento de mercado se denomina a la acción de un establecimiento o negocio a dividir y definir, de entre un universo de posibles consumidores de su producto o servicio, quienes son los clientes potenciales, para desarrollar una planificación estratégica que produzca una ventaja competitiva.

La segmentación del mercado va de la mano junto con la categoría en la que se encuentre el negocio. Ya que si el hostal es de una estrella significa que carecerá de un conjunto de servicios y facilidad que un hostal de dos o tres estrellas posee.

Sin embargo, dos hostales de una estrella no necesariamente ofrecen exactamente los mismos servicios en el mercado y lo mismo ocurre con los de dos y tres estrellas, por lo que, algunos requerirán de una segmentación de mercado más especifica que se acople a las necesidades de los huéspedes que el negocio desea atraer. Por ejemplo, si un hostal de una estrella ofrece servicios básicos, limpieza sin costo adicional y cobra por habitación sin importar la cantidad de huéspedes que hagan uso de ella a un precio económico, se puede deducir que la segmentación de mercado del negocio detallado está orientada hacia grupos de personas con bajos recursos económicos.

Dentro de las tres categorías los hostales pueden llegar a tener la misma segmentación de mercado dependiendo bajo que parámetros se esté basando cada negocio, obligando a cada uno a desarrollar una ventaja competitiva de negocio para permanecer en el mercado.

La segmentación de mercado en el centro histórico de Quito son los turistas por ser patrimonio cultural, por esta razón es uno de los sectores de la ciudad de Quito más demandada por turistas durante todo el año.

No obstante, es recomendable que los negocios de los hostales del centro histórico vayan más allá de enfocarse únicamente en los clientes potenciales sino en todo posible consumidor con el objetivo de mantener al negocio en constante producción.

2.2 Procesos internos de los hostales del Centro Histórico de Quito

2.2.1 Procesos del área de recepción

La recepción de un hostal es aquella manejada por lo general por un empleado del negocio, el cual, recibe a sus clientes ayudándoles con el pago de su estadía, hacer el Check in y Check out o simplemente una reservación, además, de permanecer a las órdenes de sus huéspedes en la recepción en todo momento que ellos lo requieran. El empleado o mejor conocido en este caso como recepcionista realiza el proceso registrando todos los datos en una agenda o cuaderno, lo cual es altamente riesgoso en caso de pérdida de este documento.

El proceso inicia cuando el cliente llega a la recepción y solicita hacer el Check in o una reservación. Seguido de una serie de preguntas por parte de recepcionista de los datos

personales y el servicio que desea el futuro huésped. Después de haber recolectado la información necesaria para el Check-in o reservación el recepcionista cotiza un valor de los servicios solicitados por el cliente. Seguido de esto, se pregunta al cliente si está de acuerdo con lo mencionado y se procede hacer el pago o la reserva de todos los servicios incluidos en la cotización. Se emite la factura correspondiente y se entrega un documento de constancia de todos los servicios pagados por el cliente.

En caso de ser una reservación el proceso finaliza después de emitir la factura con el documento antes mencionado. Por otro lado, si se trata del Check-in se procede a entregar la o las habitaciones que el cliente solicitó y el proceso finaliza.

2.2.2 Procesos del área de limpieza y mantenimiento

La limpieza y mantenimiento es un proceso indispensable manejado por todos los negocios, y más tratándose de hospedaje. Para ello los hostales suelen tener un empleado encargado de todas las tareas de limpieza y mantenimiento.

El proceso inicia cuando los huéspedes o la habitación se encuentra desocupada o en mantenimiento, entonces el empleado puede ingresar y hacer la limpieza adecuada a la habitación. Además, de hacer limpieza a las áreas externas a las habitaciones. El empleado lleva una hoja de control en la cual se registra cada una de las tareas que va completando por habitación, por ejemplo, si tiene que tender la cama de una de las habitaciones, una vez finalizada la tarea pondrá un Check en la casilla Cama tendida de la habitación N.

El proceso finaliza cuando todas las casillas de la hoja de control que se le asignó al empleado de la limpieza hayan sido marcadas como listas para ser ocupadas.

2.2.3 Procesos del área de bar

El bar es un servicio que no todos los hostales del centro histórico de Quito ofrecen, sin embargo, hay un porcentaje considerable que sí lo hace, por esta razón es importante tomarlo en cuenta. El servicio es igual al de una cafetería fuera del establecimiento; ofrece bebidas calientes y frías, frutas, sándwiches, snacks, etc.

El proceso inicia cuando el huésped se acerca al área del bar a solicitar ya sea un producto que haya o no haya estado incluido en su reservación. El recepcionista procede a registrar la venta de los productos junto con los datos del cliente y su habitación. Al final del día es importante actualizar el stock y revisar el inventario de los productos vendidos para notificar cualquier novedad al administrador del hostal.

2.2.4 Procesos de inventariado

El inventariado es un proceso que hasta en los hogares es aplicado. Es muy importante ya que nos permite conocer el estado de los productos o servicios que ofrece un negocio para tomar las medidas correspondientes, pues, si el stock es muy bajo, se necesita adquirir nueva mercadería.

El proceso inicia en contabilizar toda la mercadería que se adquiere para el stock en primera instancia, a partir de allí, se debe hacer un control periódico junto con los registros de los productos vendidos. Por lo general para un hostal se sugiere hacer el control general de toda la mercadería que el negocio ofrece mensualmente, y en caso de ser necesario, en periodos más cortos.

Se añade al proceso un reporte de las estadísticas de venta de cada uno de los productos vendidos para que el supervisor o dueño puedan tomar las decisiones correspondientes. Además, de incluir una lista de los productos y/o servicios mayormente solicitados por los huéspedes y que no se encontraban dentro de la oferta del negocio.

El proceso finaliza cuando la persona encarga de contabilizar los productos que el hostal dispone define un stock para cada uno de ellos. En caso de haber un stock bajo o de cero, se notifica al supervisor o dueño para tomar acciones rápidamente.

2.2.5 Procesos del área de contabilidad

La contabilidad es probablemente el proceso más importante de todo negocio, ya que es donde se ven los resultados en términos monetarios mostrando si el negocio realmente está dando el retorno de la inversión que se tenía planificado.

El proceso se hace mensualmente con la popularmente conocida hoja de balance, se definen los ingresos y egresos, para determinar la utilidad bruta y la utilidad neta mensual del negocio. Es importante saber que este proceso también se debe hacer anual para que el negocio pueda hacer las declaraciones correspondientes, además, de proveer

datos anuales para hacer un análisis estadístico del estado de la empresa durante cada año.

Para tener un control correcto de dicho proceso es importante entregar siempre al cliente final una factura o recibo de todo lo que haya sido consumido dentro del negocio, por lo que el hostal tendrá siempre un respaldo de los números que tenga en su hoja de balances.

El proceso finaliza cuando se determina la utilidad neta del periodo que se está analizando.

2.3 Metodología de desarrollo del sistema

2.3.1 Metodologías Ágiles

Laínez (2015) afirma: “Las metodologías ágiles varían en sus prácticas y en sus fases, sin embargo, comparten algunas características tales como: desarrollo iterativo e incremental, comunicación y reducción de productos intermediarios y de la documentación extensiva.”

Para lo cual en el presente desarrollo la práctica de mayor importancia que se va a poner en marca es el desarrollo iterativo e incremental lo que significa que el desarrollo tendrá varios ciclos aplicando una mejora continua en cada uno de ellos.

“Individuos e interacciones en vez de procesos y herramientas, software ejecutable en vez de documentación… el Manifiesto Ágil no rechaza los procesos y herramientas, la documentación o la planificación, pero simplemente muestra que estos tienen una

importancia secundaria cuando son comparados con los elementos humanos del proyecto (desarrolladores y clientes) y la rápida disponibilización de un software ejecutable, de acuerdo con la necesidad del cliente.” (Laínez Fuentes, 2016).

Se concluye que las metodologías ágiles no subestiman u omiten las prácticas, herramientas y documentación que usan las metodologías tradicionales, sino que las optimiza y las considera como una importancia secundaria.

2.3.2 Definición de Extreme Programming

Extreme Programming o popularmente conocida como XP por sus siglas, es una metodología ágil de desarrollo de software. Fue creada por Kent Beck en 1997 en el proyecto Chrysler. XP tiene una serie de principios y prácticas que abarcan la eficiencia, el costo y la calidad del software que se esté desarrollando, por esta razón es tan útil para los desarrolladores. Esta metodología ha ido evolucionando a lo largo del tiempo, logrando resolver uno de los problemas más comunes en proyectos de desarrollo de software, que es la entrega de este con una alta calidad de forma rápida, además de alcanzar las necesidades de negocio que están en constante cambio en la actualidad.

La palabra Extreme se debe al hecho de que hace uso al extremo las mejores prácticas de la ingeniería de software. XP considera importante una serie de valores que el equipo de desarrollo debe tener y son: Feedback, Comunicación, Simplicidad, Coraje y Respeto. Todos y cada uno de ellos están orientados a la relación con el cliente.

A pesar de, que los valores son importantes en la metodología, Beck definió los principios que un equipo que haga uso de esta metodología debe seguir para su correcta aplicación, y son:

- Feedback rápido
- Presunta simplicidad
- Cambios incrementales
- Aceptación de los cambios
- Alta calidad

“La metodología XP define cuatro variables para cualquier proyecto de software: costo, tiempo, calidad y alcance.” (Joskowicz, 2008).

Por esta razón es que la metodología es muy completa a la hora de realizar un proyecto de desarrollo, sin embargo, cada una de las 4 variables depende la una de la otra, por lo que, si el costo aumenta en un proyecto, probablemente fue porque el tiempo disminuyo, el alcance aumento y la calidad se mantiene. Así ocurre con cualquiera de las 4 variables.

Extreme Programming como ya se mencionó es una metodología que se aplica en varios ciclos de desarrollo y cada ciclo tiene un flujo importante que se debe considerar el cual está conformado por cuatro fases que son:

Planificación

En la cual básicamente se define como se va a trabajar durante todo el ciclo, se asignan roles, el cliente define las historias de usuario y en conjunto con el equipo de desarrollo dan prioridades y definen que historia se va a desarrollar en cada ciclo.

“Las “Historias de usuarios” (“User stories”) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente.” (Joskowicz, 2008).

Una vez definidas las historias de usuario que se desarrollarán en cada ciclo, el equipo de desarrollo asigna tareas en base a cada historia de usuario y definen tiempos estimados para cumplir con dichas tareas.

Diseño

Dentro de la fase de diseño se desarrollan prototipos que den solución a cada una de las historias de usuario asignadas para cada ciclo de desarrollo. Además, esta etapa permite al proyecto ser escalable, ya que, al diseñar prototipos, los diseños finales no necesariamente deben ser estrictamente idénticos, pero si ayudan en un alto porcentaje a su desarrollo ya que los prototipos son funcionales, es decir, son pensados en el problema que se expone en la historia de usuario y proponen la solución plasmada en un diseño.

Codificación (Rediseño)

En la codificación o también llamada fase de rediseño, se codifica el sistema. Conocida como fase de rediseño, porque cuando se codifica el sistema, muchas veces los prototipos desarrollados en la fase de diseño cambian. A pesar de que fueron pensados en ofrecer la solución al problema definido en las historias de usuarios, el programador suele detectar nuevas funcionalidades importantes para el sistema. El diseño propuesto

en los prototipos a nivel estructural por lo general se sugiere no cambiar, ya que fueron pensados junto con el cliente para dar la mejor experiencia de usuario a la hora de manejar el sistema. Por esta razón, los prototipos siempre son bastante acercados a la realidad.

Además de exponer la funcionalidad final que va a tener el desarrollo de cada módulo del sistema, también se exponen las funciones y/o fragmentos de código más relevantes para hacer posible dicha solución. Por esta razón la metodología es ágil, ya que se enfoca solamente en las partes relevantes del sistema y como se desarrollaron para que sean factibles.

Pruebas

Dentro de la fase de pruebas, tenemos las pruebas unitarias y los criterios de aceptación. Ambas habrán sido previamente definidas en las historias de usuario, es decir, además de haber detallado el problema que se quiere solucionar, también se detalló que es lo que el sistema debe hacer para cumplir con la solución a dicho problema. Por ende, las pruebas unitarias son aplicadas a cada historia de usuario y los criterios de aceptación son aquellos definidos en cada historia por cada funcionalidad que el sistema debe cumplir para ser aceptado, por esta razón, se conoce a dicho proceso verificación de los criterios de aceptación.

2.3.3 Aplicación de la metodología en el sistema

"El uso de la XP debe ser realizado en empresas que priman los procesos dinámicos de desarrollo, basados en la colaboración del cliente…" (Laínez Fuentes, 2016).

Para el presente trabajo de disertación de grado se define que la metodología XP es la más adecuada, puesto que, uno de los hostales del centro histórico de Quito está dispuesto escribir las historias de usuario y a dar el feedback necesario para el desarrollador, además, de poner a disposición sus instalaciones para que el desarrollo sea lo más cercano posible a la realidad. Haciendo posible cumplir el primer principio del XP, que es el feedback rápido y, en consecuencia, el cumplimiento de los cuatro principios restantes.

Se debe considerar que en el presente trabajo el equipo de desarrollo está conformado por un solo programador, el cual estará en constante interacción con el cliente en el proceso de desarrollo siguiendo las pautas de la metodología, como si el equipo de desarrollo estuviera conformado por varias personas.

En cuanto a las fases del ciclo de vida del software se desarrollarán de la siguiente manera:

Planificación

Dentro de la fase de planificación se definirá las historias de usuario, las tareas a realizar y el tiempo estimado de culminación.

Diseño

Para el diseño se utilizará Mockup Builder para el prototipo de la solución a cada historia de usuario. Además, se hace una breve explicación de cómo debería funcionar dicho prototipo.

Codificación (Rediseño)

Dentro de la codificación se detallan los fragmentos de código relevantes para la solución desarrollada, además de presentar capturas de pantalla de la solución y del código escrito en el sistema.

Pruebas

En la fase de pruebas se tienen dos tipos de pruebas:

- **Pruebas de caja blanca**

  Aquellas que verifican que cada solución o módulo desarrollado no contengan ningún error a nivel de visualización y/o rendimiento.

- **Pruebas de caja negra:**

  Aquellas pruebas que se encargan de verificar que cada módulo del sistema cumpla con las necesidades que el sistema requiere.

2.3.4 Desarrollo por capas y consumo de datos

El modelo vista controlador o MVC es el desarrollo por capas más popular y usado por muchos marcos de trabajo desarrollados en distintos lenguajes de programación. El desarrollo por capas es un modelo de desarrollo software que se basa en la división en partes un sistema software o también una arquitectura cliente-servidor: lógica de negocios, capa de presentación y capa de datos. Este modelo facilita al desarrollador crear diferentes interfaces de usuario sobre un mismo sistema sin requerirse cambio alguno en la capa de datos o lógica. La lógica de desarrollo por capas brinda una solución que durante años fue un gran problema a los desarrolladores, el cual consistía en que, si se requiere de una modificación, solo afectará al nivel requerido sin tener que revisar entre el código fuente de otros módulos.

Por otro lado, brinda la fácil distribución del trabajo de una solución de software por niveles, ya que cada grupo de trabajo con cierta tarea asignada debe enfocarse a su tarea sin preocuparse por las demás sino hasta la fase de integración. Sin embargo, esta metodología a pesar de que cada grupo desarrolla su tarea de manera independiente, no dificulta la integración de los módulos.

El consumo de datos básicamente se refiere a la capa encargada de la interacción con la base de datos. Si bien es cierto que todas las capas tienen datos en algún punto del proceso, la capa que realmente está encargada de obtener esos datos es la capa de datos para posterior a eso las otras capas se encarguen de darle un sentido o una lógica a dichos datos y presentarlos al usuario final.

2.4 Herramientas para el desarrollo del sistema

2.4.1 PowerDesigner

PowerDesigner es una herramienta que facilita el modelamiento de la arquitectura y de los datos de cualquier modelo de negocio que requiera de almacenamiento de data en un motor de base de datos o simplemente ayuda en el modelamiento de la arquitectura empresarial en una organización.

Esta herramienta soporta una serie considerable de sistemas de base de datos relacionales las cuales son mayormente utilizadas en el mercado. Brinda una gran facilidad en diseño de arquitectura de datos y modelamiento de datos, es la herramienta líder para la solución de estas prácticas.

Para el presente trabajo de disertación se hará uso de la herramienta para modelar de forma relacional la lógica de los procesos internos de los hostales del centro histórico de Quito como son las Ventas y la Disponibilidad de las habitaciones en términos generales considerando las entidades que sean necesarias para abarcar completamente los procesos.

El programa posee también una variedad de herramientas para modelar procesos y todo tipo de diagramas de flujo que ayudan al entendimiento de cualquier modelo de negocio. En el presente trabajo de disertación de grado de hará uso del diagrama de procesos

para representar de manera gráfica propiamente cómo funcionan los procesos de los hostales del centro histórico de Quito.

2.4.2 Mockup Builder

Mockup Builder es una herramienta que permite a los usuarios plasmar la idea de una solución que tengan en su cabeza en un prototipo visual para detectar la funcionalidad real que pueda tener dicha solución y determinar si esa es viable o no. Mockup Builder entre otras herramientas de Prototipado es una de las más utilizadas gracias a su facilidad de uso, tiene una distribución para Windows y Mac. En el presente trabajo de disertación se ha utilizado para presentar las pantallas principales y ver en términos muy generales la funcionalidad que va a tener el sistema una vez que sea desarrollado.

2.4.3 Base de datos MySQL y MySQL WorkBench

MySQL es un motor de gestión de base de datos que a pesar de haber sido comprada por Oracle posee un licenciamiento Publica General o por sus siglas en inglés GPL lo que hace que sea de código abierto. El sistema de bases de datos MySQL se puede obtener de manera gratuita de su página oficial junto con MySQL Workbench la herramienta visual para la gestión de las bases de datos.

MySQL Workbench es la herramienta para manejo visual gráfico del diseño de base de datos para el motor MySQL. Esta herramienta facilita la gestión de las bases de datos de manera visual de manera fácil y brindando una experiencia de usuario agradable. La herramienta es compatible con varios lenguajes de programación entre otras ventajas competitivas la cual hace de la herramienta una de las más utilizadas para el desarrollo web en la actualidad.

2.4.4 Servidor Apache XAMPP

Servidor de aplicaciones es aquel software que permite a un ordenador comunicarse con otro. Un ordenador que trabaje como servidor su función principal es recibir solicitudes de información para recopilarla y enviarla de regreso hacia el lado del cliente.

"Existen muchas opciones de software de servidor, pero las dos más populares son Apache (software de código abierto) y Microsoft Internet Information Services (IIS). … También hay una versión de Windows. Microsoft IIS es parte de la familia de soluciones de servidor de Microsoft." (Robbins, 2012).

(BitRock Inc., 2018) "XAMPP es el entorno más popular de desarrollo con PHP. XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl."

Conociendo esto, se puede decir que XAMPP es el servidor de aplicaciones de código abierto que permite visualizar todos los desarrollos hechos con el lenguaje de programación PHP, mediante una distribución de base de datos llamada MariaDB de MySQL.

2.4.5 Lenguajes de programación

2.4.5.1 HTML5

HTML 5 es un lenguaje de etiquetas que nace a partir del lenguaje de etiquetas html 4.0 y xhtml, aunque HTML5 nació como una versión completa y sin las limitaciones que tenía html 4.0 y xhtml, no se hace llamar como la versión mejorada, sino más bien como un lenguaje de etiquetas totalmente independiente de los antes mencionados. HTML significa lenguaje de marcado de hipertexto por sus siglas en inglés y hace referencia al lenguaje de etiquetas que dan una estructura a las páginas web. HTML5 por otro lado tiene el mismo objetivo, de hecho, se usan las mismas etiquetas que en el html convencional, con la diferencia que es un lenguaje con estándares establecidos de estructura, nuevas etiquetas para dicha estructura y otras funcionalidades que limitaban bastante al lenguaje de etiquetas html convencional.

(Van Lancker, 2012) afirma: “Las novedades de Html5 y sobre todo del trinomio conceptual ‘Html5 + CSS3 y aplicaciones JavaScript’ son numerosas

Html5

- Un nuevo doctype simplificado y unificado
- Supresión de etiquetas y atributos de presentación
- Nuevas etiquetas semánticas o de organización
- Nuevas etiquetas de audio y de video que ya no necesitan más plugin dedicados
- Diseño 2D y pronto 3D gracias a la nueva etiqueta <canvas>
- Profusión de formularios innovadores como por ejemplo cursores o calendarios, y la integración de forma nativa en los navegadores de la validación de datos.”

Al afirmar que se usan las etiquetas del html convencional más nuevas etiquetas, entre otras novedades, cualquiera podría deducir que es una versión mejorada, pero lo que

Html5 quiere demostrar es la gran compatibilidad que tendrá con los diferentes lenguajes y navegadores, algo que xhtml fracaso en su versión 2.0 que fue la última que saco después de anunciar que el grupo encargado de dicho desarrollo estaría disuelto.

Html5 junto con CSS3 y nuevas novedades de JavaScript dan avances muy prometedores en la evolución del desarrollo web.

2.4.5.2 CSS3

CSS significa Cascade Style Sheets, su primera versión la 1 nace en la década de los 90, seguida de la versión 2 y 2.1 en los próximos años. A pesar de que el desarrollo de estas primeras versiones fue complejo y tomo años hacerlo, para la versión 3 se había adoptado una nueva metodología de desarrollo llamada modular, agrupando el trabajo grande en módulos funcionales independientes que van evolucionando lo que permitió a los navegadores ir evolucionando junto con el lenguaje de hojas de estilo. A diferencia de HTML5, CSS3 si se hace llamar una nueva versión después de la 2.1 con un gran cambio y evolución.

(Van Lancker, 2012) afirma: “Dentro de las novedades que nos presenta CSS3 están:

- Nuevos selectores
- Bordes redondeados
- Bordes decorados
- Tipos de letra personalizados que permitirán a la tipografía encontrar por fin su sitio en las interfaces web.

- Presentación de un texto en varias columnas.
- Sombras aplicadas al texto y al resto de elementos
- Degradados de color.
- Múltiples fondos.
- Opacidad o transparencia
- Transformaciones
- Transiciones (sin JavaScript)

Estas nuevas especificaciones CSS3 ya están disponibles en las versiones más recientes de los navegadores …"

Todas estas nuevas novedades que presenta CSS3, dan a los diseñadores y desarrolladores web una amplia gama de posibilidades para innovar en sus nuevos desarrollos y prototipos.

2.4.5.3 JavaScript

El lenguaje de programación JavaScript es el más popular dentro de los lenguajes de aplicaciones cliente-servidor en la web. Mientras que, el lenguaje HTML permite estructurar las páginas web de una forma atractiva presentando la información al usuario, sigue siendo un lenguaje muy poco interactivo con el usuario por lo que, JavaScript entra como el lenguaje que complementa la interacción del aspecto visual atractivo que se puede desarrollar con HTML.

Ventajas

- Agrega funcionalidades interactivas a las páginas web.

- Gracias a la tecnología AJAX de JavaScript, varios procesos se pueden realizar en segundo plano mientras que el usuario continúa navegando en el sitio, sin tener que esperar a una respuesta del servidor.
- Es un lenguaje que trabaja con objetos puesto que, el nombre de JavaScript nació de Java y éste es un lenguaje orientado a objetos. Esto permite que sea fácil de aprender y manejar.
- Es un lenguaje que no requiere de un programa específico para su codificación, se puede escribir código JavaScript en un simple WordPad y funciona.

Desventajas

- A pesar de, tener la tecnología AJAX para realizar procesos en segundo plano que por lo general son llamados a la base de datos, JavaScript por sí solo no puede interactuar con una base de datos sin la ayuda de un lenguaje de programación externo como PHP o ASP.
- JavaScript puede presentar muchas cosas al usuario, sin embargo, dicha información no está escrita en el disco del cliente por lo que todo lo que sea desarrollado netamente con JavaScript no actualizará el disco si no es con la ayuda de otro lenguaje como PHP.

2.4.5.4 PHP

PHP es un lenguaje de programación de código abierto interpretado, de alto nivel, usado dentro de los desarrollos web. Este lenguaje se ejecuta en el lado del servidor, por lo que, el código escrito en PHP no es visible al cliente a diferencia del HTML que si lo es. Al afirmar que, el código escrito no es visible al cliente, se refiere a que el cliente no conoce las peticiones que el programador dentro del código PHP está haciendo al servidor, pero si va a conocer la respuesta del servidor, sin olvidar que el retorno ya fue interpretado. Una de las características esenciales del lenguaje es que es interpretado, pero no compilado, por esta razón a PHP se lo conoce como uno de los lenguajes de mejor rendimiento en la web. A pesar de, que la sintaxis del lenguaje es similar a C y C++, tiene ciertas características como, por ejemplo, sabiendo que C y C++ trabajan con variables, y dichas variables obligatoriamente deben tener un tipo de dato, en el lenguaje PHP no requiere de un tipo de dato para las variables, por lo que facilitan la interacción de estas.

Como se ha dicho anteriormente el código PHP es codificado dentro de la codificación del lenguaje de etiquetas HTML, por lo que, es necesario tener en cuenta que para que el intérprete sepa que sección del código es HTML y cual es PHP deben ser diferenciados de alguna manera. El diferenciador para PHP es el siguiente:

<?PHP Código PHP ?>

Entendiendo que esta sección puede estar en cualquier parte del código de etiquetas HTML, las herramientas de codificación para programadores suelen dar un color característico a la apertura de código PHP y al cierre de este. Como, por ejemplo:

```
<body>
    <div>
        <?PHP echo “Hola Mundo!” ?>
    </div>
</body>
```

(Capuñay, 2013) afirma: “Cuando un cliente WEB (navegador o browser) hace una solicitud a un servidor Web, éste ubica el archivo solicitado y verifica si contiene código PHP o no, en caso de que sólo sea HTML. El archivo es enviado directamente al cliente, pero en caso de que tenga código PHP es enviado a ser procesado el documento que ha sido solicitado para ser procesado y luego enviar el resultado del procesamiento al cliente.”

Sabiendo que si el archivo enviado al servidor no contiene código PHP es retornado inmediatamente hacia el cliente, se puede concluir que, una página web sin código PHP es más rápida, sin embargo, toda página que sea HTML neto corre un altísimo riesgo de hackeo e inseguridad en sus datos además que existe cierta funcionalidad que hoy en día es indispensable en toda página web y son los formularios, para tener un formulario a pesar de que es un grupo de etiquetas HTML, no tiene ninguna funcionalidad si no es integrado con código PHP, por lo que no es nada recomendable omitir tener este lenguaje en un desarrollo web.

2.4.5 Sublime Text

Sublime Text es un editor de texto y editor de código fuente desarrollado en C++ y Python para poder disponer de plugin. A pesar de, haber sido desarrollado en estos 2 lenguajes, soporta la edición de código fuente de los lenguajes de programación más populares como son: HTML, CSS, JavaScript, PHP, R, C#, Matlab, Perl y hasta SQL.

2.4.6 Framework

Framework o en español Marco de Trabajo, es un ambiente estandarizado de trabajo que ayuda al programador en el desarrollo de las soluciones de software. Por lo general los desarrollos sin la utilización de un framework son muy inestables o están expuestos a ataques de hackeo. Entendiendo que, las desventajas que conlleva un desarrollo sin un framework podemos concluir que en términos generales dependiendo del framework que el programador use, puede ayudar en la arquitectura, estructura, seguridad, funcionalidad, escalabilidad de la aplicación web.

2.4.6.1 Laravel

(Sierra, 2018) afirma: “Laravel es un framework PHP de código abierto que intenta aprovechar las ventajas de otros Frameworks y desarrollar con las últimas versiones de PHP.”

Podemos concluir que es uno de los frameworks más completos que existen ya que ha aprovechado la funcionalidad de otros para implementar un marco de trabajo completo para el desarrollador además de utilizar las versiones más actuales del lenguaje de programación PHP.

Conociendo ya que el lenguaje de programación PHP es uno de los más rápidos en términos de rendimiento y funcionalidad, laravel viene a ser un complemento bastante funcional para los programadores gracias a su filosofía y su desarrollo por capas.

"Su filosofía es desarrollar código PHP de forma elegante y simple basado en un modelo MVC(Modelo-Vista-Controlador)." (Sierra, 2018)

"La potencia de Laravel radica en su integración, escalabilidad y facilidad de mantenimiento respecto a otros desarrollos en lenguajes 100% nativos y por lo tanto es una opción más que a tener en cuenta a la hora de decidir usar este framework en nuestros desarrollos… Como comentamos antes, la clave son las necesidades, por lo que elegiremos Laravel para desarrollos a medida cuando:

- Los CMS no cubren nuestras necesidades como aplicación web.
- Necesitamos de un desarrollo a medida ágil, seguro y con una fácil escalabilidad y mantenimiento.
- Si además de necesitar una solución web, esta tiene que estar integrada con otras aplicaciones (Google, servicios de mailing, ERP's, etc.)." (Sierra, 2018)

Conociendo que laravel es ideal cuando se habla de integración, escalabilidad y facilidad de mantenimiento suena bastante atractivo para el programador a la hora de elegir una tecnología para un desarrollo web. Por otro lado, se tiene un desarrollo ágil, seguro y abierto a cualquier tipo de diseño lo que suele ser un dolor de cabeza cuando utilizamos un CMS, suena atractivo para el cliente, concluimos que laravel brinda una serie de factores positivos a la hora de seleccionar una tecnología para una solución de software web.

Capítulo 3

3. Desarrollo del Sistema

En el capítulo “Desarrollo del Sistema” se va a crear el sistema propuesto para el trabajo de disertación dividido en una serie de iteraciones para su elaboración. XP propone dividir dichas iteraciones en los siguientes pasos:

1. **Planificación:** Se define las Historias de Usuario y los Criterios de Aceptación.
2. **Diseño:** Se propone un prototipo y tarjetas CRC (Clase, Responsabilidad, Colaboración) consisten en exponer cómo el usuario va a manejar cada módulo propuesto para el sistema.
3. **Codificación:** Codificación del sistema y Diseño del sistema en producción.
4. **Pruebas:** Se realizan Pruebas Unitarias y Pruebas de Aceptación de cada módulo del sistema.

El sistema se desarrollará con el marco de trabajo laravel, ideal para aplicaciones web, con un nivel de seguridad alto gracias a su esquema por capas. La arquitectura de Laravel se basa en Modelo Vista Controlador (MVC). Conociendo la tecnología que se utilizará en el sistema y la arquitectura por capas que tiene Laravel se denota de la siguiente manera dicha arquitectura:

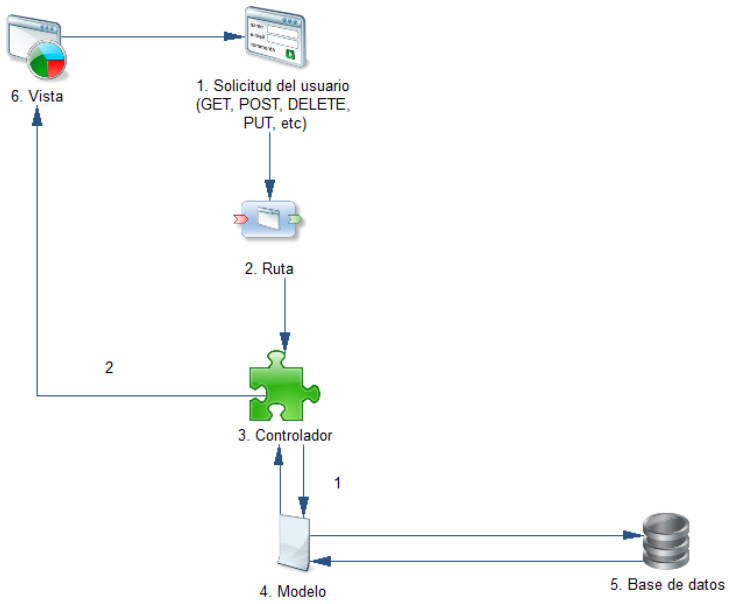


Ilustración 1 Arquitectura de Laravel (Esparza, 2018)

3.1 Primera Iteración

En el primer ciclo de desarrollo se va a codificar la autenticación de usuario, la página principal del sistema presentando los módulos del sistema macro y el desarrollo del módulo de Habitaciones.

3.1.1 Planificación

Historia de Usuario 1

Se desarrollará una página minimalista con las opciones para que el usuario con una cuenta pueda autenticarse en el sistema.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
1	Usuario del sistema	Requiere de autenticación de usuario para ingreso al sistema de administración	Módulo de seguridad para evitar que cualquiera pueda administrar el sistema	1	Autenticación	La página debe solicitar usuario y contraseña	Llenado del formulario de autenticación y envío de información	Autenticación exitosa o usuario y contraseña erróneos.

Ilustración 2 Historia de Usuario 1 (Esparza, 2018)

Tareas para realizar:

- Definición estructural de la página de autenticación
- Definir colores y logos globales para el sistema
- Creación de un usuario para la autenticación y de un token que identifique al usuario activo en el sistema de gestión.
- Desarrollo de la página de autenticación de usuario
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 1 a 2 días

Historia de Usuario 2

Se desarrollará una página principal del Sistema de manejo de procesos presentando los módulos del sistema que se pueden gestionar.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
2	Usuario del sistema	Requiere de una sección principal para conocer los módulos que el sistema podrá gestionar.	Visualizar todos los módulos que el sistema podrá gestionar.	1	Mostrar un boton de acceso a cada módulo del sistema, además de un atajo a cada funcionalidad en la sección lateral	Sistema posee boton de ingreso a todos los módulos y un link de atajo en la sección lateral	Pantalla visual de los módulos. "Click" en cualquiera de ellos y en los links de atajo	Visualización de la pantalla principal del sistema de gestión con los módulos del sistema

Ilustración 3 Historia de Usuario 2 (Esparza, 2018)

Tareas para realizar:

- Definición estructural de la página principal del sistema de gestión
- Definir iconografía para cada módulo
- Definir orden de los módulos en el sidebar
- Desarrollo de la página principal del sistema de gestión
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- 1 día

Historia de Usuario 3

Se desarrollará el módulo “Habitaciones” el cual mostrará un listado de habitaciones y se podrá gestionar dichas habitaciones, es decir, se desarrollará un CRUD para ese módulo, permitiendo al usuario Crear, Modificar y/o Eliminar una habitación. La eliminación de un registro será lógico, es decir, el registro nunca será eliminado de la base de datos, sino que solo cambiará de estado.

Esta práctica servirá para un control de cambios en el sistema por parte de gerencia del negocio que lo maneje.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	***Rol***	***Característica / Funcionalidad***	***Razón / Resultado***	***Escenario***	***Titulo***	***Contexto***	***Evento***	***Resultado***
3	Usuario del sistema	Requiere manejar caracteristicas y estado de las habitaciones de los hostales	Conocer rápidamente las características y estado de las habitaciones en un lugar centralizado además de poder administrarlas (ingresar, modificar, eliminar).	1	Ingresar nueva habitación con sus características y estado actual	Sistema posee boton de ingreso de habitación	Click en "Nuevo" y llenado de datos	Ingreso de habitación al sistema exitoso. Se muestra en Listado de Habitaciones
				2	Mostrar Listado de habitaciones	Una tabla con el listado actual de habitaciones	Click en "Ver Habitaciones"	Despliegue en una tabla de las habitaciones del hostal
				3	Modificar características y estado de una habitación	Sistema posee boton de modificacion de habitacion	Click en "Modificar" y cambio de los datos	Modificación exitoso. Se muestran los datos actualizados
				4	Eliminar habitación	Sistema posee boton de eliminación	Click en "Eliminar" y confirmación	Despues de confirmado el evento, mostrar tabla de habitaciones sin el registro eliminado

Ilustración 4 Historia de Usuario 3 (Esparza, 2018)

Tareas para realizar:

- Definición estructural del módulo de habitaciones, incluyendo una página para el listado de estas y una página para el CRUD del módulo.
- Definir fotos de habitaciones de un hostal
- Definir iconografía para el módulo
- Desarrollo del módulo habitaciones
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 2 a 3 días

3.1.2 Diseño

El prototipo de la página de autenticación va a contener un encabezado con el logo representativo del “Sistema de Administración de Hostales” y debajo de éste un formulario con las credenciales de autenticación (Login).

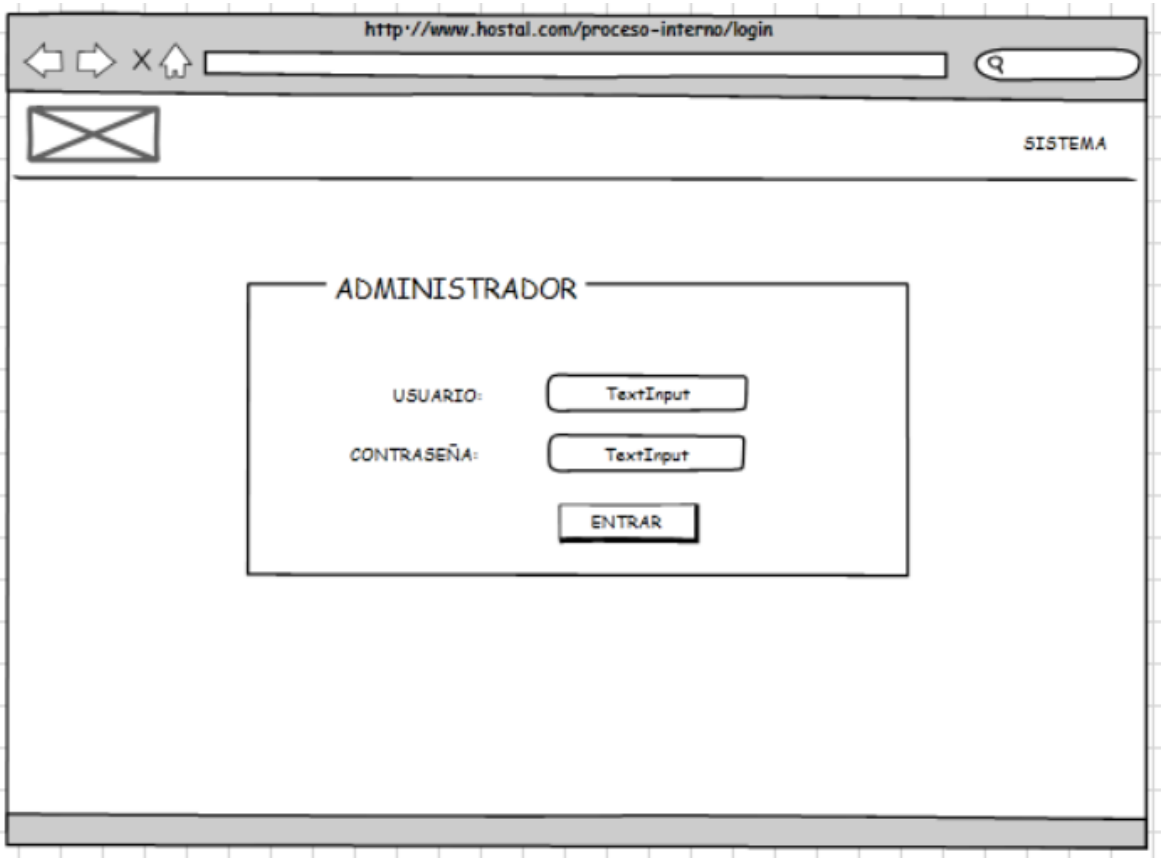


Ilustración 5 Prototipo de Autenticación para el Sistema (Esparza, 2018)

Tarjeta CRC

- Empleado del hostal
    - Se debe tomar en cuenta que el empleado, por lo general el recepcionista del hostal, tendrá el acceso al sistema a nivel general, por lo que la Tarjeta CRC será la misma para todos los módulos e historias de usuario

EMPLEADO DEL HOSTAL	
Tendrá un usuario y una contraseña para el ingreso al sistema	Podrá ingresar al sistema y gestionar los módulos que el sistema tendrá

Ilustración 6 Tarjeta CRC de un Empleado del hostal (Esparza, 2018)

La página tendrá un encabezado con el logo del sistema de administración, un sidebar con atajos a todos los módulos y submódulos del sistema y en el centro botones para ingreso a la gestión de cada uno de los módulos del sistema.

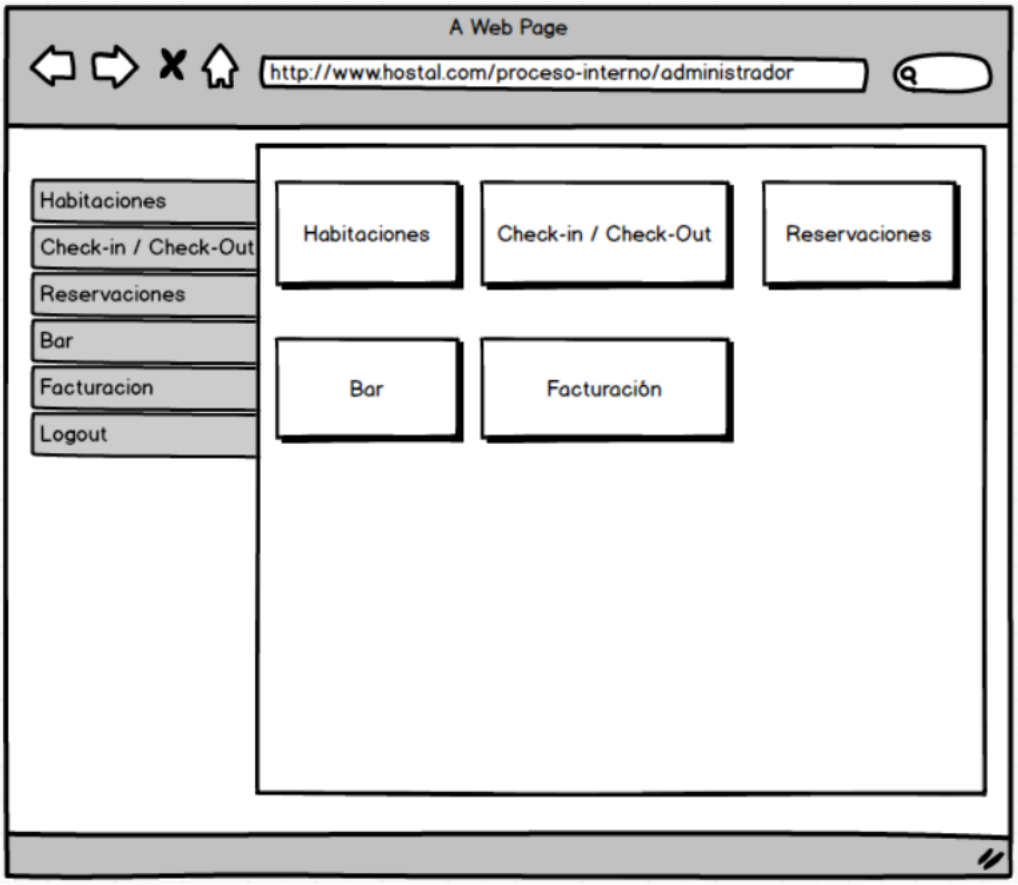


Ilustración 7 Prototipo Opciones Generales del Sistema (Esparza, 2018)

La pantalla “Habitaciones” mostrará un listado de habitaciones y se podrá gestionar dichas habitaciones, es decir, se desarrollará un CRUD para ese módulo, permitiendo al usuario Crear, Modificar y/o Eliminar una habitación. La eliminación de un registro será lógico, para un control de cambios en el sistema por parte de gerencia del negocio que lo maneje.

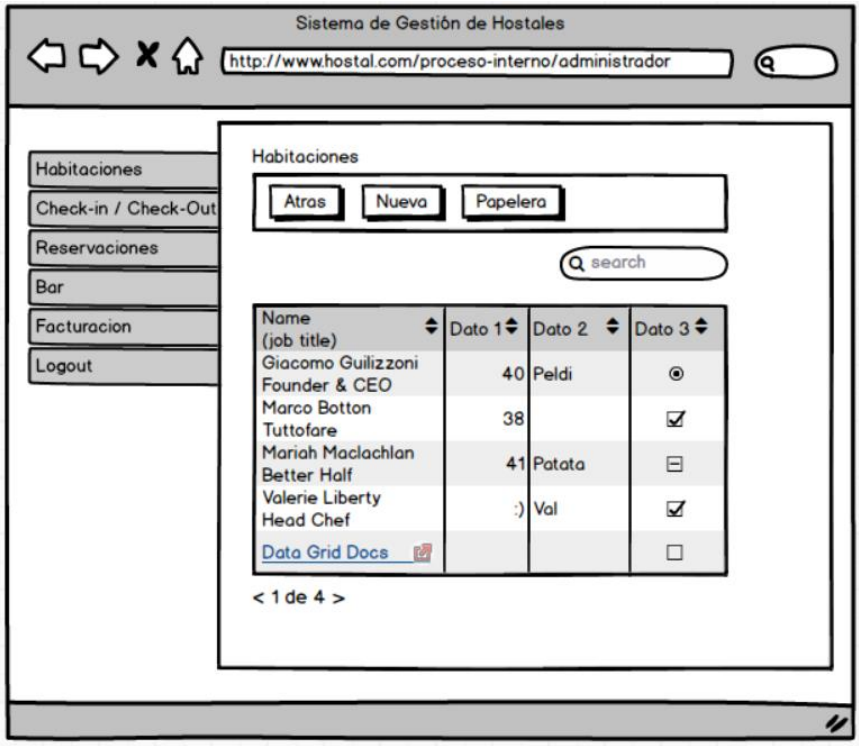


Ilustración 8 Prototipo Listado de Habitaciones (Esparza, 2018)

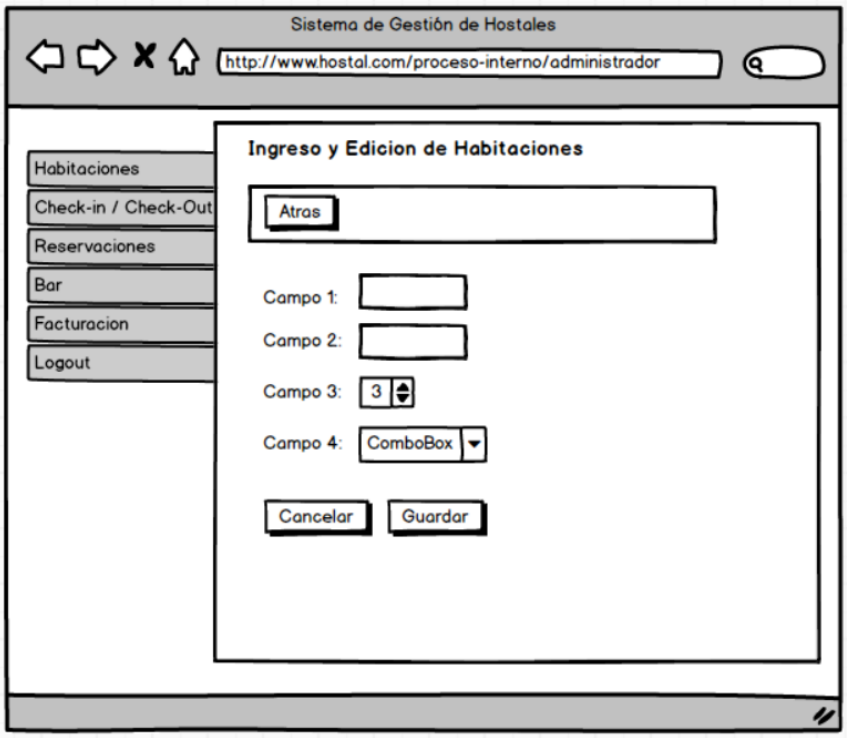


Ilustración 9 Prototipo CRUD Insert y Edit de Habitaciones (Esparza, 2018)

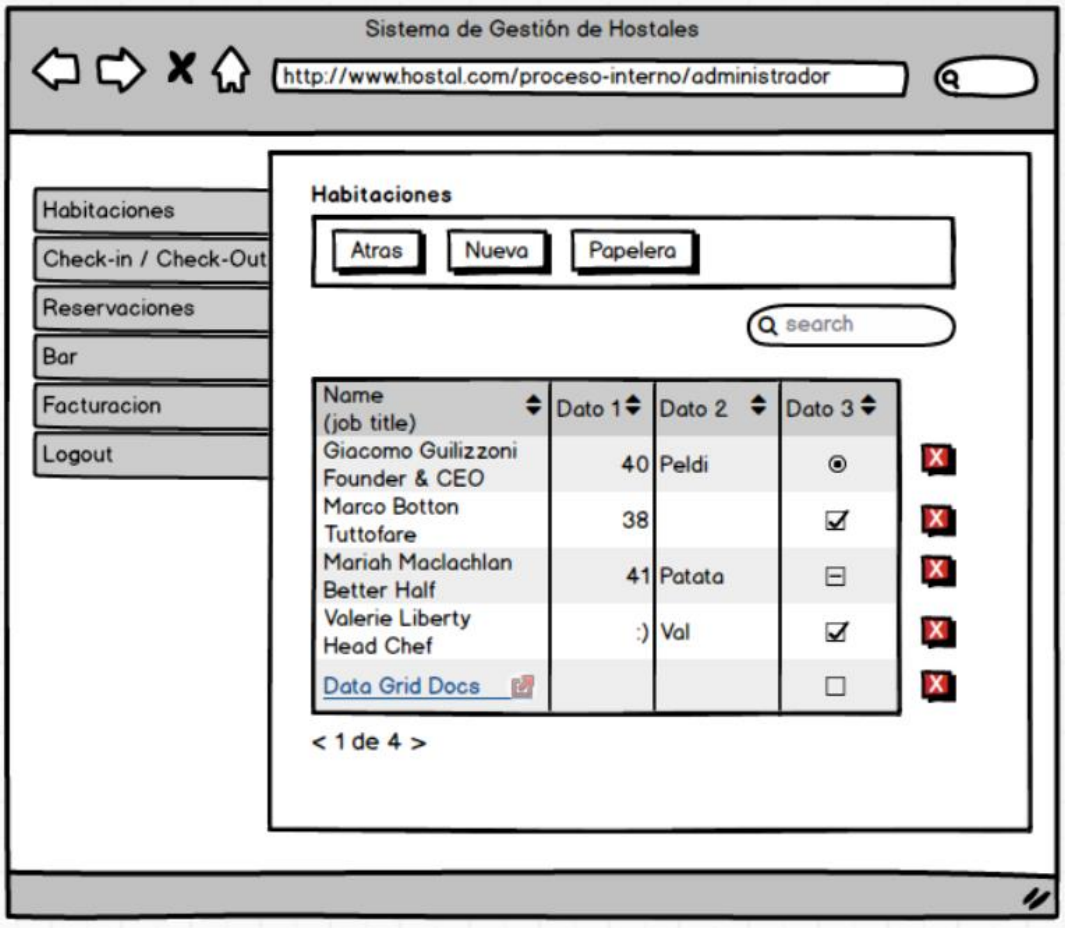


Ilustración 10 Prototipo Delete del CRUD de Habitaciones (Esparza, 2018)

3.1.3 Codificación

Para iniciar con el desarrollo del sistema, primero se verifica la instalación de los siguientes programas y/o frameworks que se va a usar:

- XAMPP
- MySQL Workbench

- Sublime Text 3
- Composer
- Framework Laravel

Una vez verificado que todo está instalado correctamente, se crea una carpeta para el proyecto en el directorio:

- C:\xampp\htdocs\SistemaHostales

Posterior a esto abrimos el intérprete de comandos CMD, nos posicionamos en la carpeta del proyecto creada y se corre el siguiente comando para crear un proyecto de laravel:

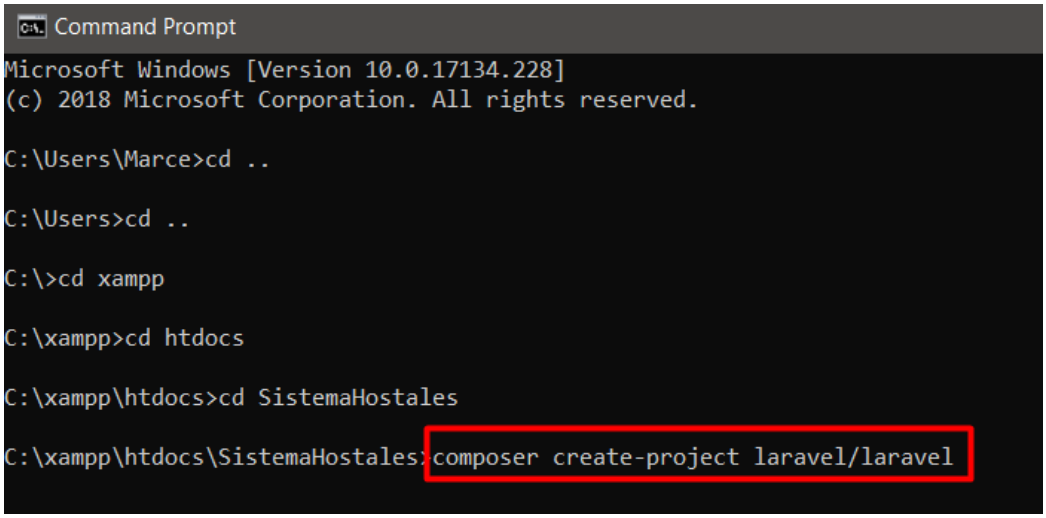


Ilustración 11 Comando de creación de proyecto con Laravel Framework (Esparza, 2018)

Una vez creado el proyecto se procede a crear una base de datos, configurar la conexión a la base de datos y creación de Modelos y Controladores los cuales serán encargados de la transferencia de datos entre la aplicación y el motor de base de datos. Para el módulo Autenticación, Página principal y Habitaciones se va a realizar los siguientes pasos:

Generación del script de creación de base de datos del modelo físico realizado en PowerDesigner:

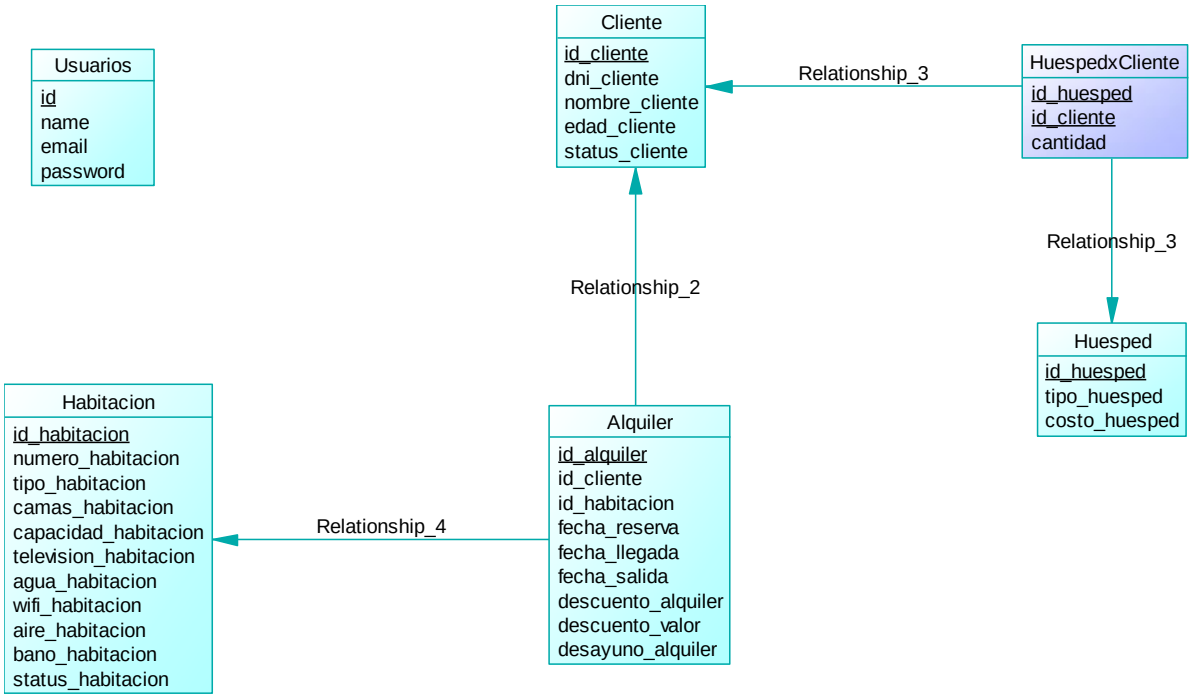


Ilustración 12 Modelo Físico del Diagrama de Clases del Sistema (Esparza, 2018)

Con el script se procede a la creación de la base de datos, y posterior se configura la conexión del proyecto a esta, modificando los archivos “.env” y “database.php”:



Ilustración 13 Conexión a la base de datos en archivo database.php (Esparza, 2018)

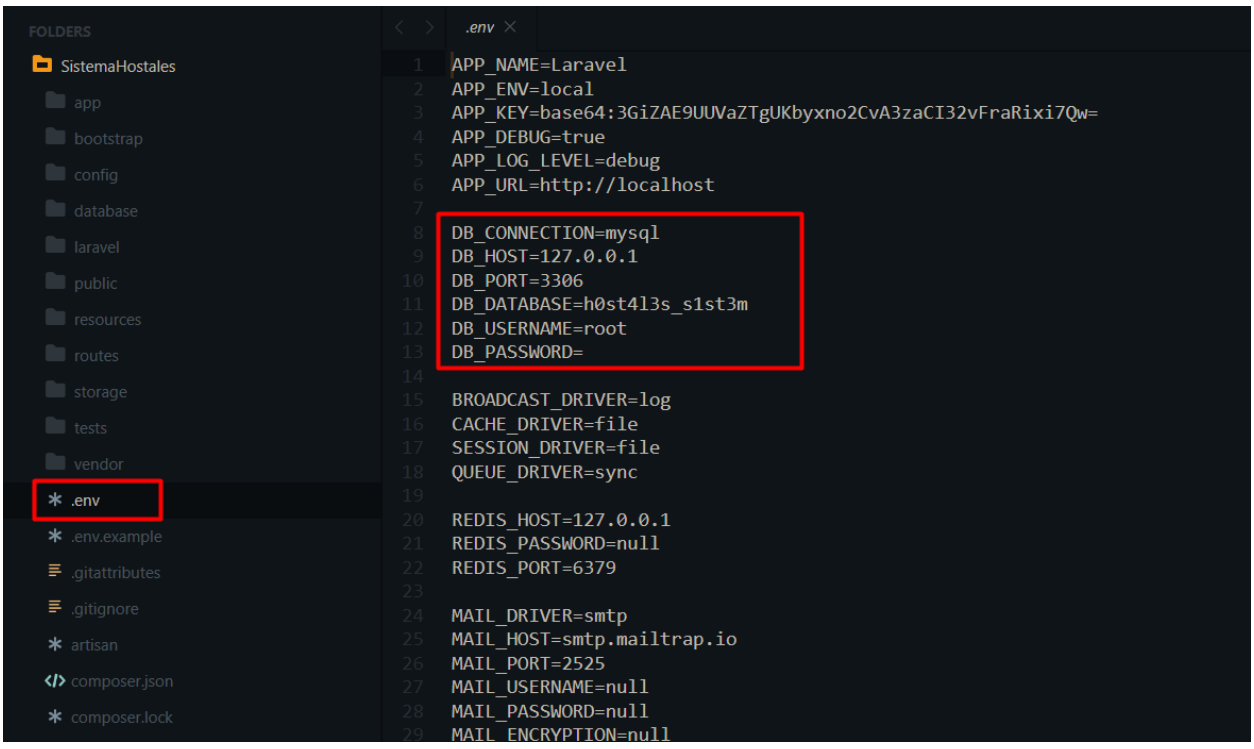


Ilustración 14 Conexión a la base de datos en archivo .env (Esparza, 2018)

Una vez conectado nuestro proyecto podemos crear los modelos y los controladores:

```
C:\xampp\htdocs\SistemaHostales>php artisan make:model Models/Habitacion
```

Ilustración 15 Creación de un Modelo en Laravel (Esparza, 2018)

```
C:\xampp\htdocs\SistemaHostales>php artisan make:controller HabitacionController
```

Ilustración 16 Creación de un Controlador en Laravel (Esparza, 2018)

Se debe tener en cuenta que, para interactuar con todas las tablas de la base de datos, el proyecto debe tener un modelo creado por cada tabla. En el caso de los controladores, se irán creando en base a la necesidad del programador, en el presente proyecto se creará un controlador por cada módulo del sistema.

Después de realizar lo antes mencionado se puede iniciar la codificación del sistema y de las historias de usuario propuestas para el presente ciclo.

Historia de usuario 1: Autenticación al Sistema

Laravel provee una facilidad para crear mediante un comando un Módulo de Autenticación normalmente usado para aplicaciones con social media. Este comando crea las rutas, las tablas en la base de datos, el modelo y el controlador. En el presente sistema el dueño del negocio podrá crear las cuentas de usuario para sus empleados y posterior a eso deshabilitar la ruta de registro, por lo que el sistema permitirá únicamente Autenticación de Usuarios previamente creados.

```
C:\xampp\htdocs\SistemaHostales>php artisan make:auth
```

Ilustración 17 Creación de Autenticación de Laravel por CMD (Esparza, 2018)

```
C:\xampp\htdocs\SistemaHostales>php artisan migrate
```

Ilustración 18 Creación de Tablas de Usuarios de Laravel por CMD (Esparza, 2018)

Una vez ejecutados los comandos mencionados Laravel crea por defecto una vista para el Login de Usuarios, por lo que, en el presente trabajo de disertación se va a ajustar para adaptar a las necesidades del sistema.

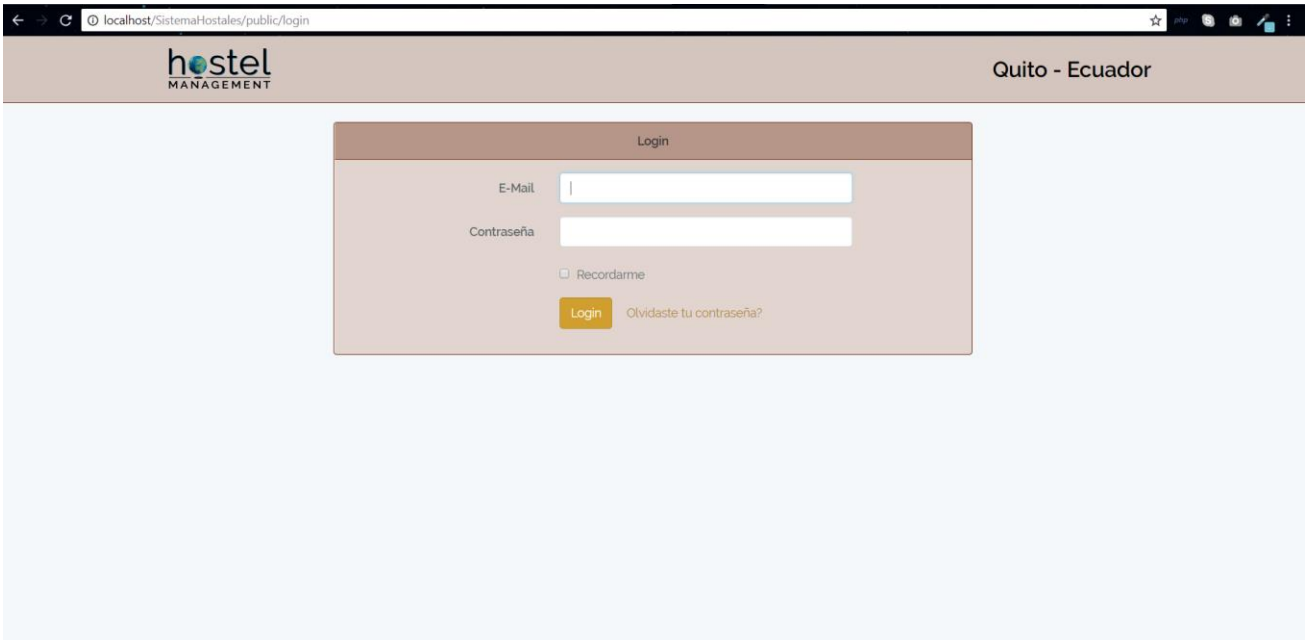


Ilustración 19 Interfaz de Usuario Página de Autenticación (Esparza, 2018)

Historia de usuario 2: Página Principal del Sistema

Para la página principal del sistema el controlador retornará la vista únicamente. Nos encargamos de crear la hoja de estilos CSS que estilice la página de una manera amigable al usuario y basándose en el prototipo propuesto.

Para la creación de la página principal fue necesario la creación de los siguientes ítems:

Paquetes

- Archivos “header”, “sidebar” y “plantilla” que serán heredados en prácticamente todo el sistema.
- Dentro de la carpeta “views” una carpeta llamada “hms” que contendrá el contenido de todas las vistas del sistema.
- Dentro de la carpeta “public”, creación de carpetas css, js e images las cuales contendrán las hojas de estilo, archivos javascript con librerías necesarias para el sistema y la carpeta images para alojar toda imagen que el sistema requiera.

Clases

- Al ser una vista sin datos, no requiere de ninguna clase

Métodos

- El método index dentro del controlador creado por defecto en laravel que únicamente retorne la vista.

Rutas

- Creación de dos posibles rutas para el despliegue de la pantalla principal:
    - /
    - /administrador

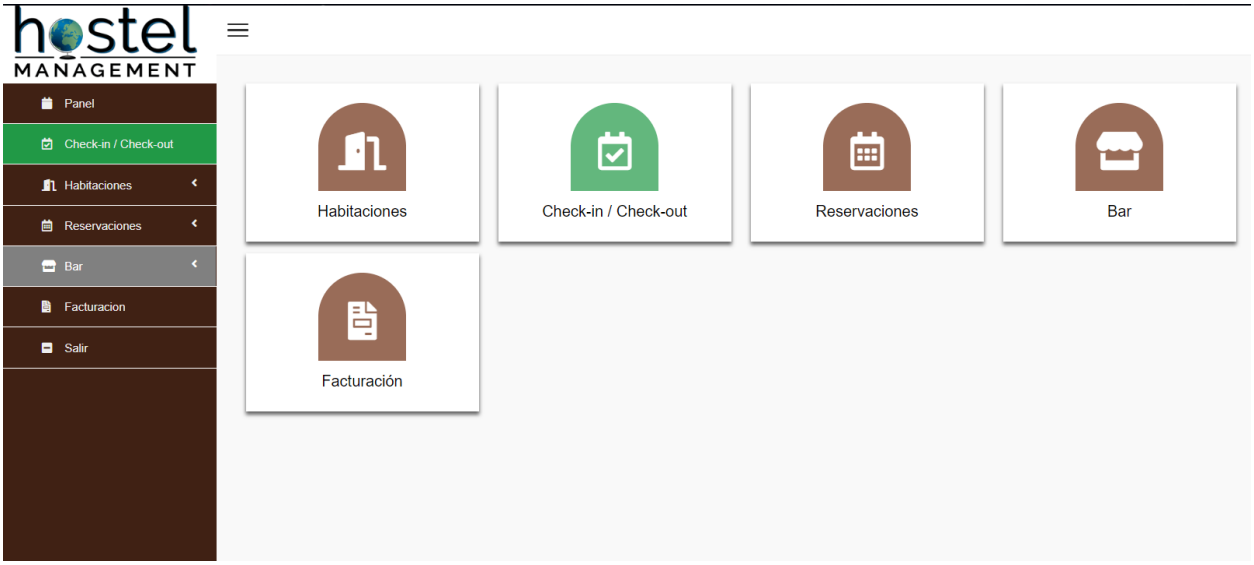


Ilustración 20 Vista de la Pantalla principal del Sistema (Esparza, 2018)

Historia de Usuario 3: Habitaciones

Para el presente módulo se van a codificar 3 pantallas:

1. Listado de Habitaciones

Para la creación del Listado de Habitaciones fue necesario la creación de los siguientes ítems:

Paquetes

- Dentro de la carpeta “hms” en las vistas, creamos una carpeta llamada “modulos” y dentro de esta una para el módulo de habitaciones
- Creación del archivo “listado” que será el contenido de la vista.
- Creación de un controlador llamado: “HabitacionController”

Clases

- También conocidas como Modelos en laravel fue necesario la creación de uno solo llamado: “Habitacion”.

Métodos

- Creación de un método “Listado”

```
class HabitacionController extends Controller
{
    public function Listado()
    {
        date_default_timezone_set('America/Guayaquil');
        $habitaciones = Habitacion::where('ESTADO_REGISTRO', '1')->get();
        return \View::make('hms.modulos.habitacion.listado')
        ->with('habitaciones', $habitaciones);
    }
```

Ilustración 21 Función “Listado” de HabitacionController (Esparza, 2018)

Rutas

- Creación de una ruta:
    - /administrador/habitación

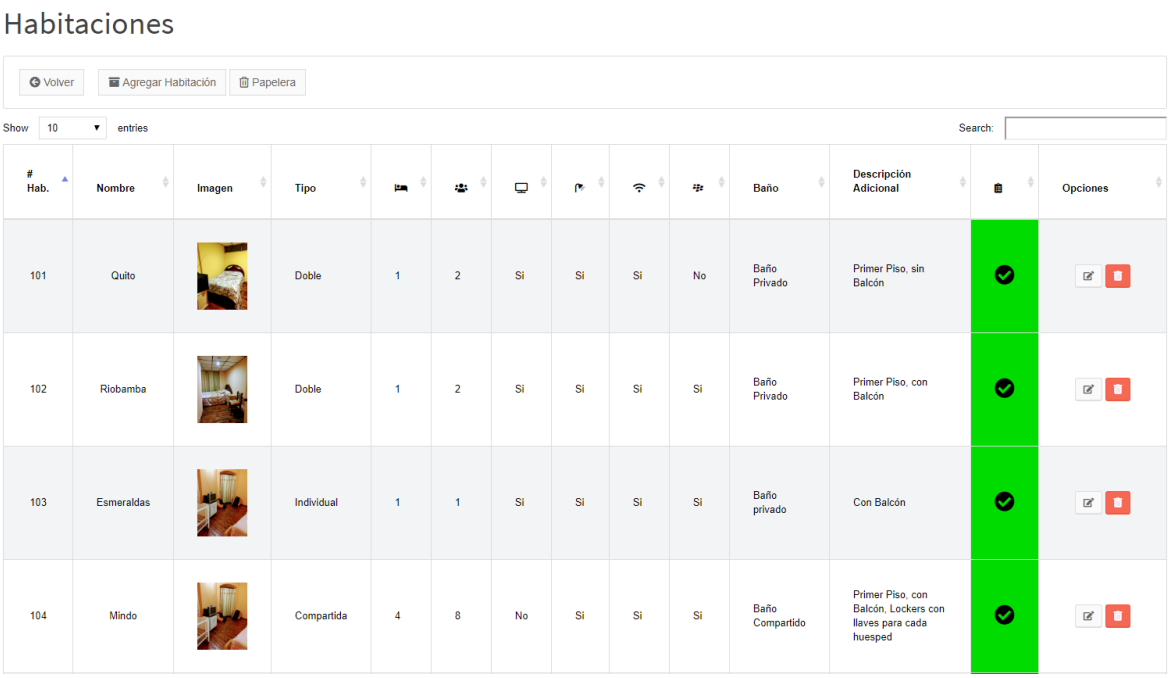


Ilustración 22 Vista de la Pantalla de Listado de Habitaciones (Esparza, 2018)

2. Agregar y Editar una Habitación

Para la creación de Inserción y Edición de Habitaciones fue necesario la creación de los siguientes ítems:

Paquetes

- Creación de dos archivos dentro de la carpeta “habitacion” de las vistas: “editar” y “nuevo”

Clases

- Ninguna clase adicional fue creada para estas vistas

Métodos

- Para el ingreso de habitación nueva:
    - Ingresar, que retornaba únicamente la vista
    - saveContent, que recolectaba los datos y los insertaba en la base de datos.
- Para la edición de una habitación:
    - Editar, que retornaba la vista con un objeto tipo “Habitacion”
    - updateContent, que realiza la actualización de los datos en la base de datos.

Rutas

- Se crearon cuatro rutas llamadas de la misma manera que los métodos del controlador:
    - /ingresar
    - /saveContent
    - /editar
    - /updateContent

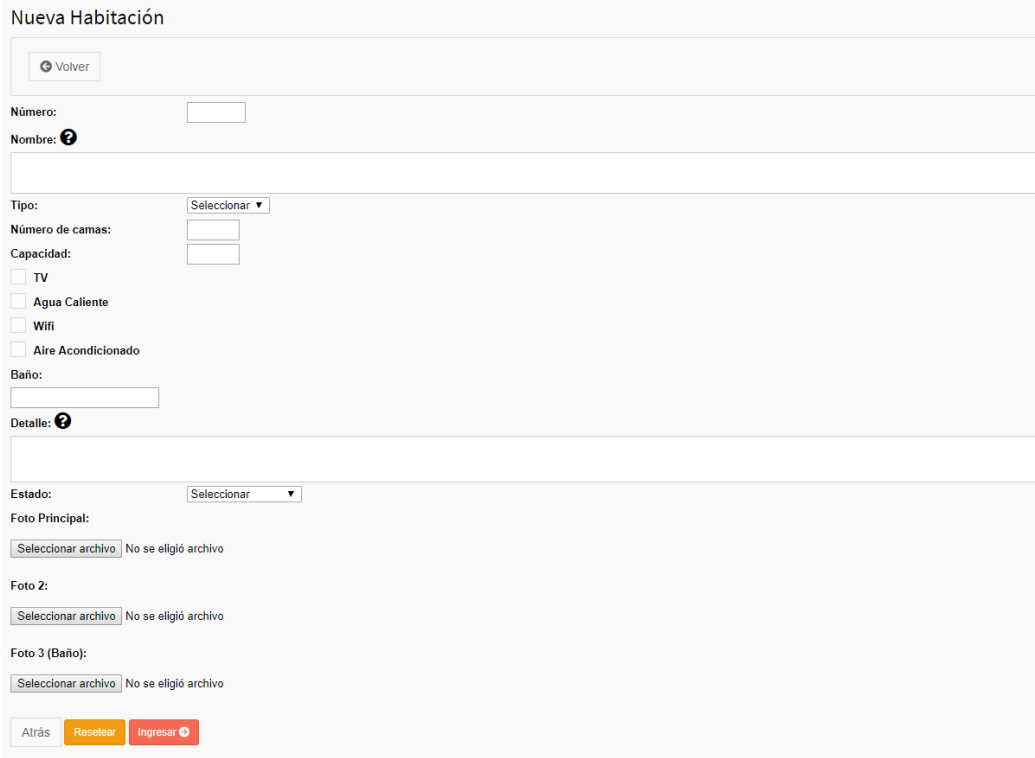


Ilustración 23 Vista de la Pantalla de Nueva Habitación (Esparza, 2018)

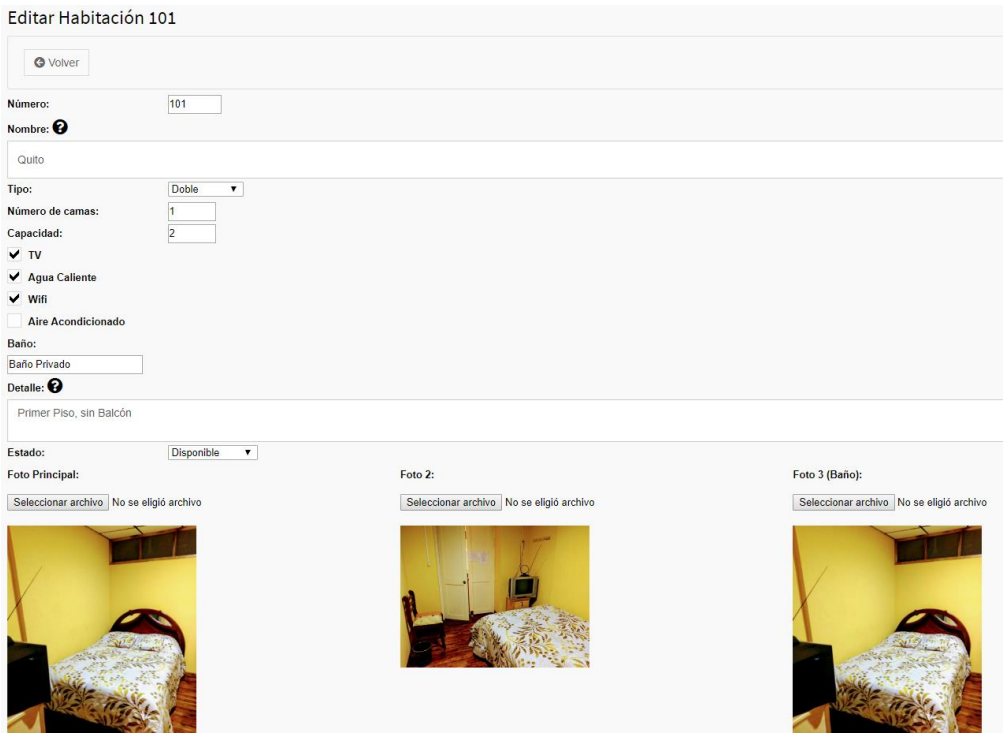


Ilustración 24 Vista de la Pantalla de Editar Habitación (Esparza, 2018)

3. Papelera de Habitaciones

Para la creación de la papelera de Habitaciones fue necesario la creación de los siguientes ítems:

Paquetes

- Creación de un archivo dentro de la carpeta “habitacion” de las vistas: “papelera”

Clases

- Ninguna clase adicional fué creada para esta vista.

Métodos

- Fueron creados tres métodos para la papelera de habitaciones:
    - Recycle, que retorna la vista con el listado de habitaciones en estado 0
    - deleteRow, que realiza una actualización de una habitación en la base de datos al estado 0.
    - recoverRow, que permite recuperar el registro de la papelera, actualizando el registro del estado 0 al estado 1.

Rutas

- Se crearon tres rutas llamadas:
    - /papelera, que llama al método Recycle
    - /eliminar, que llama al método deleteRow
    - /recover, que llama al método recoverRow

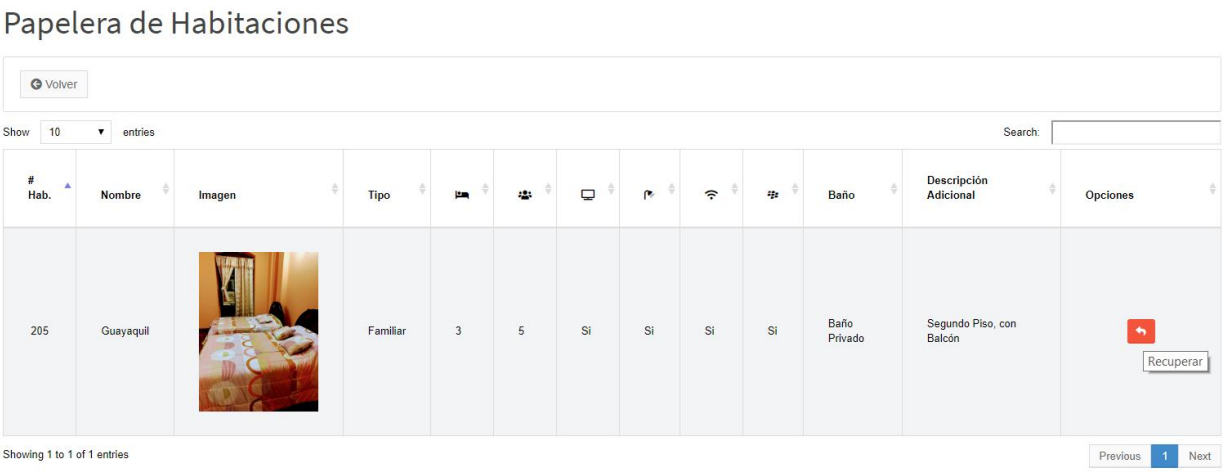


Ilustración 25 Vista de la Pantalla de Papelera de Habitaciones (Esparza, 2018)

3.1.4 Pruebas

Caja blanca: Prueba Unitaria

Se comprueba que el sistema no retorna ningún error, no posee ningún enlace roto y todos los links externos de librerías funcionan correctamente, la consola del sistema esta siempre limpia sin errores en todas las pantallas.

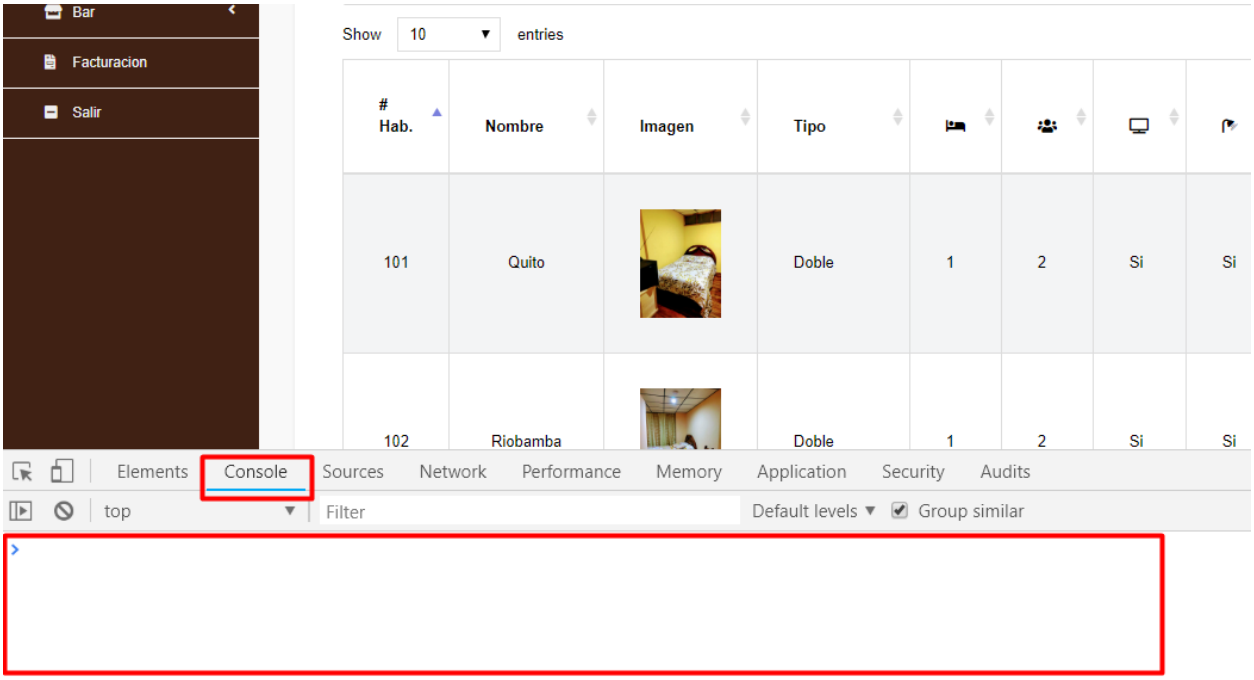


Ilustración 26 Pruebas Caja Blanca Sin errores (Esparza, 2018)

Caja Negra: Criterios de Aceptación

- Autenticación exitoso
- Página principal correcta y fácil de manejar
- Información de habitaciones clara y de fácil visualización
- Carga de datos en creación y edición de habitaciones
- Papelera de habitaciones funciona y se puede recuperar los registros sin problema

PRUEBA DE ACEPTACIÓN			
Número de prueba:	1	Número de historia de usuario:	1
			2
			3
Nombre de Caso de Prueba:	Usuario del Sistema		
Descripción			
Se evaluará la funcionalidad y la experiencia del usuario al manejar los 3 primeros modulos del sistema.			
Condiciones de ejecución			
Crear por lo menos un usuario y una contraseña. Conectarse a internet y Tener una habitación creada desde la base de datos.			
Entradas			
Formulario de autenticación: Usuario y Contraseña. Formulario para creación y edición de habitaciones: Numero de Habitación, Nombre, Tipo, Camas, Capacidad, Tv, Agua, Wifi, Baño, Detalle, Estad, Foto1, foto 2 y Foto 3.			
Resultado esperado:	Autenticación, Presentar y Cargar la información		
Evaluación	Colores más madera en las interfaces		
	Registrar en la base de datos el usuario que hizo algun movimiento		
	Agregar mensaje del significado de la iconografía usada		

Ilustración 27 Pruebas Caja Negra Primer Prueba (Esparza, 2018)

3.2 Segunda Iteración

En el segundo ciclo de desarrollo se va a codificar los módulos de Check-in, Check-out y Reservaciones, además, aplicar la evaluación obtenida del primer ciclo de desarrollo.

3.2.1 Planificación

Historia de Usuario 4

Se desarrollará el módulo “Check-in / Check-out” el cual mostrará un listado de huéspedes en el hostal y se podrá gestionar la entrada y salida de huéspedes denominado “Check-in” y “Check-out”, es decir, se desarrollará un CRUD para ese módulo, permitiendo al usuario Insertar un registro y Actualizar el estado de un huésped a Check-out.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
4	Usuario del sistema	Requiere de check-in y check-out de cada cliente o huesped	Conocer la llegada y salida de cada huesped al hostal junto con sus datos personales y la o las habitaciones solicitadas.	1	Check-in	Boton "check-in" para ingreso de nuevo huesped al hostal	Click en "Check-in" y llenado de datos	Ingreso exitoso de los datos validados del huesped y habitaciones
				2	Check-out	Boton "check-out" para salida de huesped del hostal	Buscar huesped mediante un search form y click en boton "Check-out"	Mensaje de confirmación "Huesped XXXX ha abandonado el hostal"
				3	Listado de huespedes en el hostal	Una tabla con el listado de huespedes que estan en el hostal	Click en "Ver huespedes"	Despliegue en una tabla todos los huespedes del hostal.

Ilustración 28 Historia de Usuario 4 (Esparza, 2018)

Tareas para realizar:

- Definición del proceso de Check in
- Definición de todos los flujos que puede tener el proceso
- Codificación de todo el proceso de Check-in
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 2 a 3 días

Historia de Usuario 5

Se desarrollará el módulo “Reservaciones” el cual mostrará un listado de las reservaciones que hay para el hostal y se podrá gestionar, es decir, se desarrollará un CRUD permitiendo al usuario tener un listado e ingresar nuevas reservaciones.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
5	Usuario del sistema	Requiere manejar las reservaciones que llegan al hostal y conocer los datos del cliente que reserva.	Conocer rápidamente las habitaciones que tienen una reservación y el cliente en un lugar centralizado además de poder administrarla (ingresar, modificar, eliminar).	1	Ingresar nueva reservación	Sistema posee boton de ingreso de reservación	Click en "Nuevo" y llenado de datos	Ingreso de reservación al sistema exitoso. Se muestra en Listado de Reservaciones
				2	Mostrar Listado de reservaciones	Una tabla con el listado actual de habitaciones	Click en "Ver Reservaciones"	Despliegue en una tabla las reservaciones
				3	Modificar reservación	Sistema posee boton de modificacion de reservación	Click en "Modificar" y cambio de los datos	Modificación exitoso. Se muestran los datos actualizados
				4	Eliminar reservación	Sistema posee boton de eliminación	Click en "Eliminar" y confirmación	Despues de confirmado el evento, mostrar tabla de reservaciones sin el registro eliminado

Ilustración 29 Historia de Usuario 5 (Esparza, 2018)

Tareas a realizar:

- Definición del proceso de Reservaciones
- Definición de todos los flujos que puede tener el proceso
- Codificación de todo el proceso de Reservaciones

- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 1 a 2 días

3.2.2 Diseño

El prototipo para el proceso de Check-in. El sistema brinda la opción de checkear una reserva o sin reserva. Si el usuario checkea una reserva, los datos están listos para confirmar el “Check-in” caso contrario el usuario podrá seleccionar si es un cliente nuevo, o un cliente frecuente. Si el cliente es nuevo se solicitarán todos sus datos, caso contrario, de una lista de clientes frecuentes se selecciona y la información se cargará automáticamente.

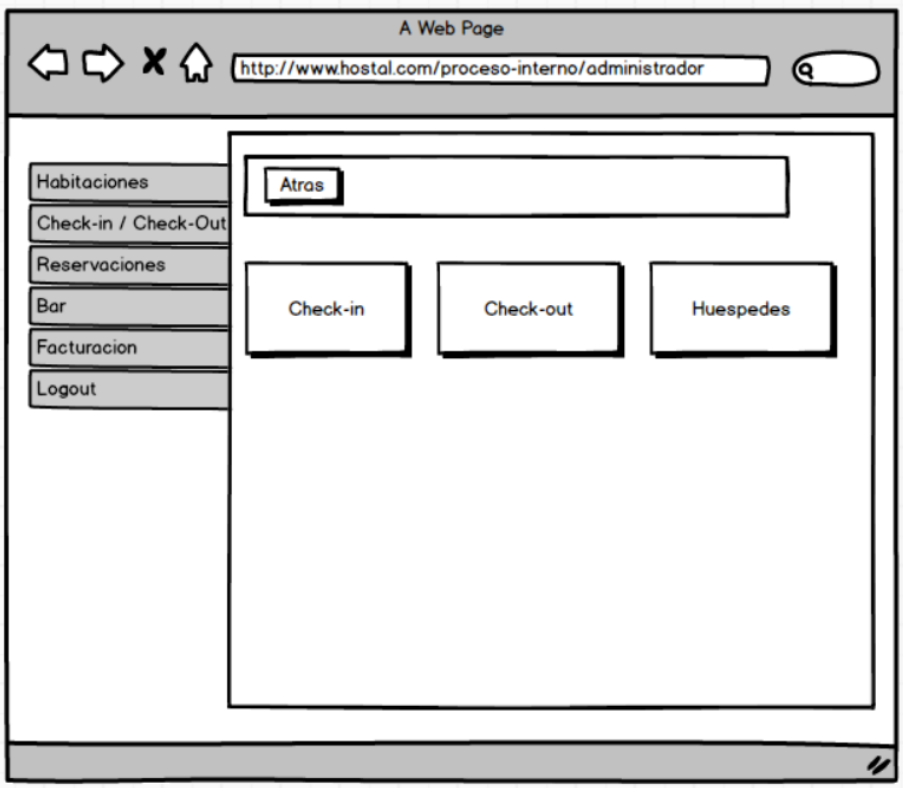


Ilustración 30 Prototipo Check-in / Check-out (Esparza, 2018)

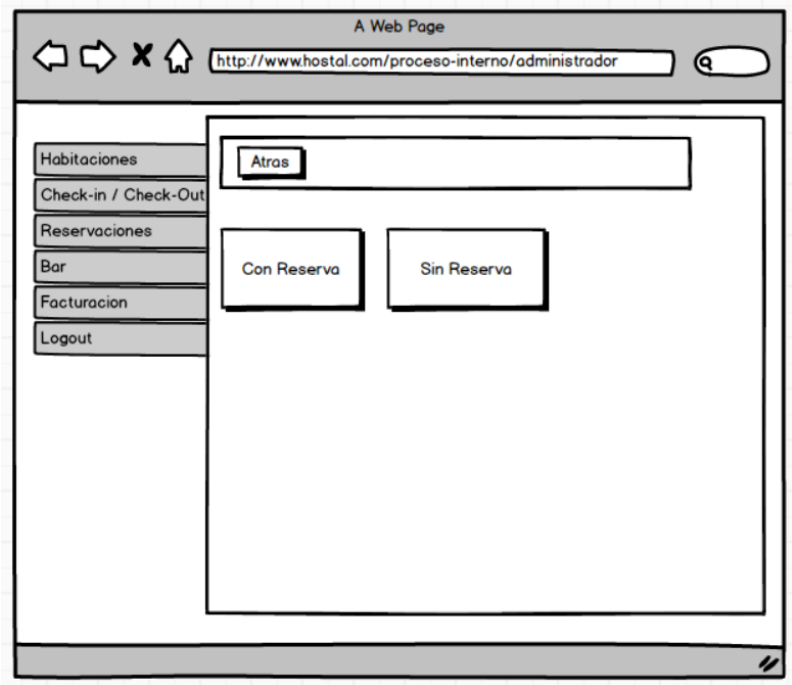


Ilustración 31 Prototipo Proceso Check-in paso 1 (Esparza, 2018)

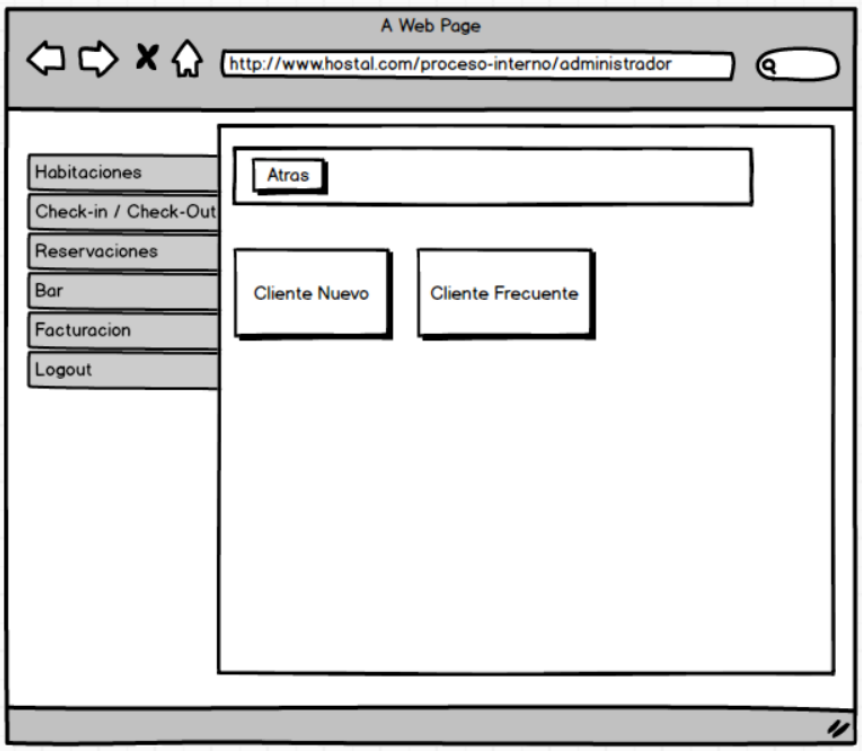


Ilustración 32 Prototipo Proceso Check-in, Sin Reserva (Esparza, 2018)

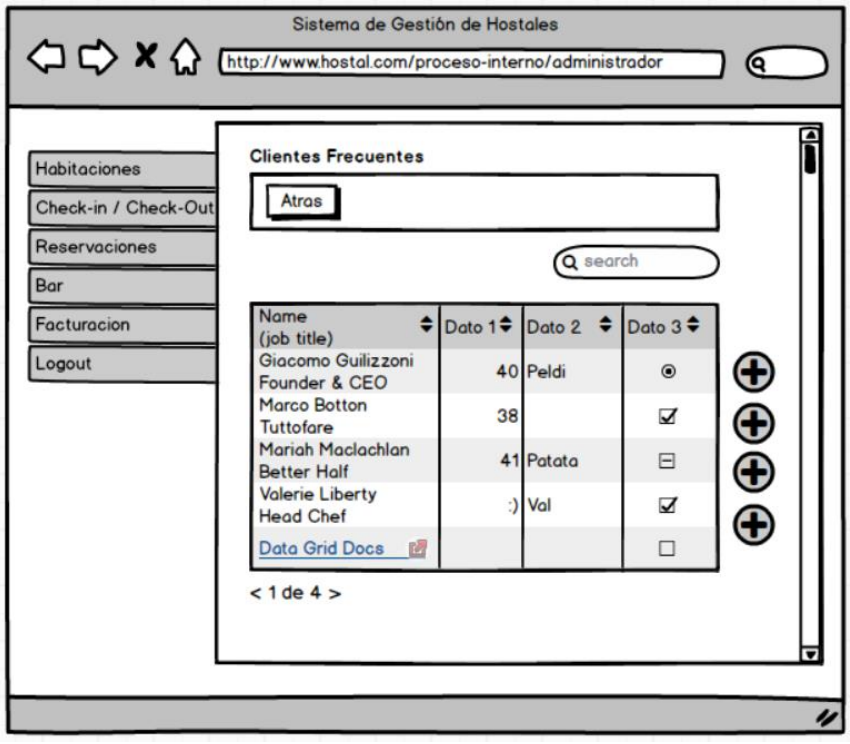


Ilustración 33 Prototipo Proceso Check-in, Sin Reserva, Cliente Frecuente (Esparza, 2018)

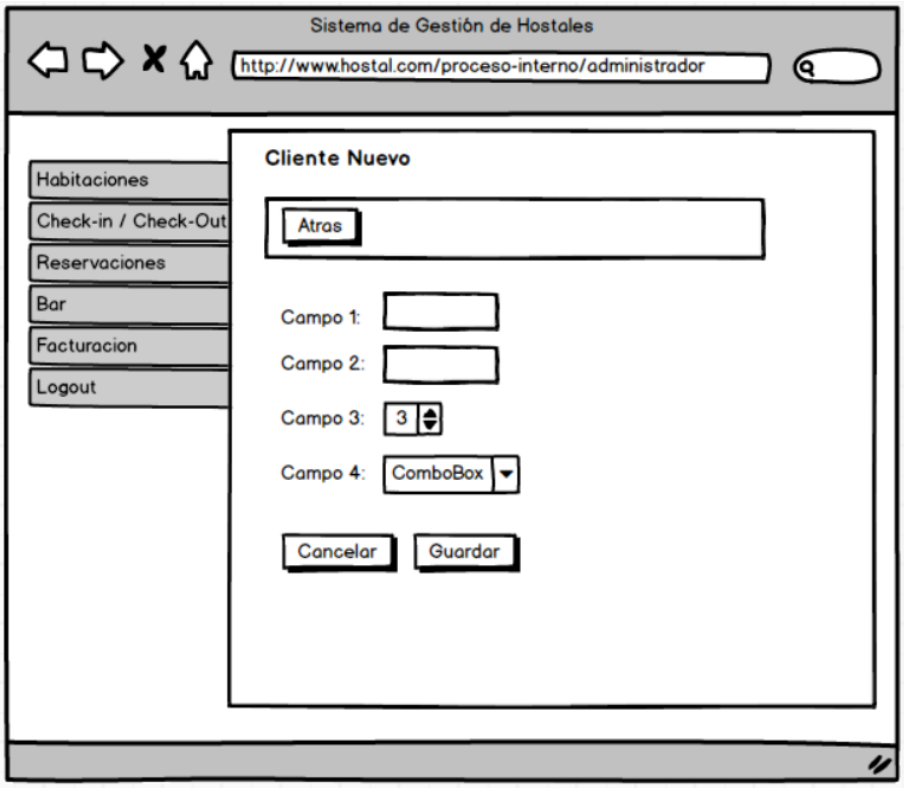


Ilustración 34 Prototipo Proceso Check-in, Sin Reserva, Cliente Nuevo (Esparza, 2018)

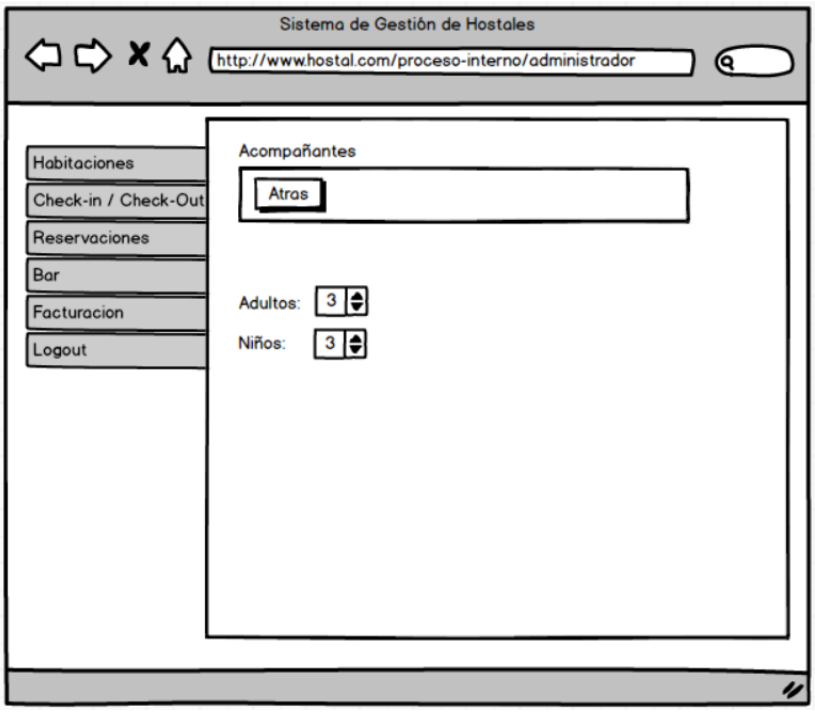


Ilustración 35 Prototipo Proceso Check-in, Acompañantes (Esparza, 2018)

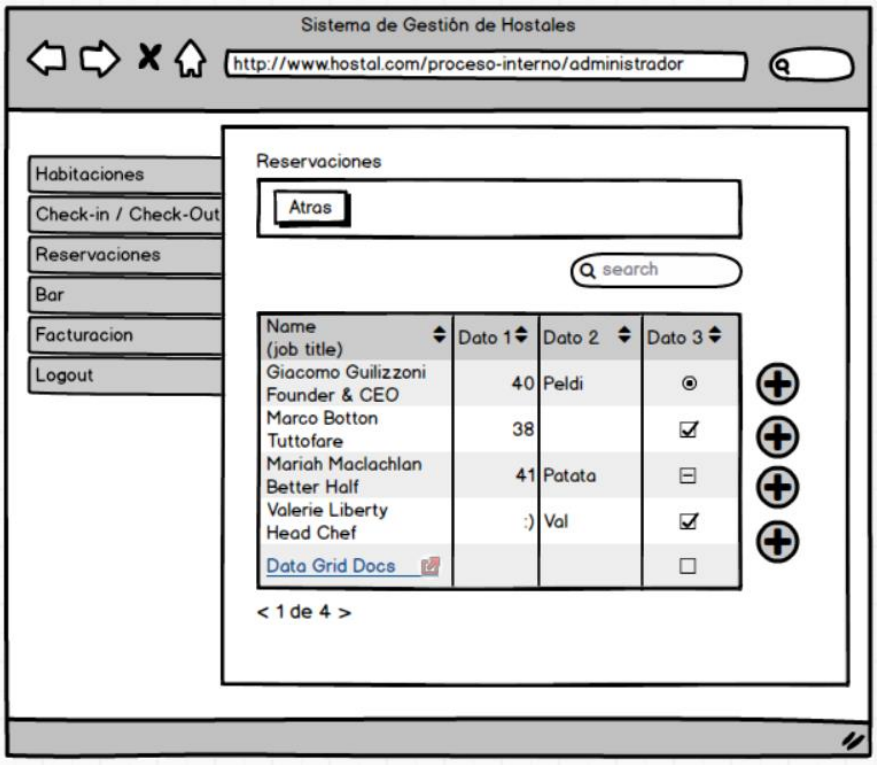


Ilustración 36 Prototipo Proceso Check-in, Con Reserva (Esparza, 2018)

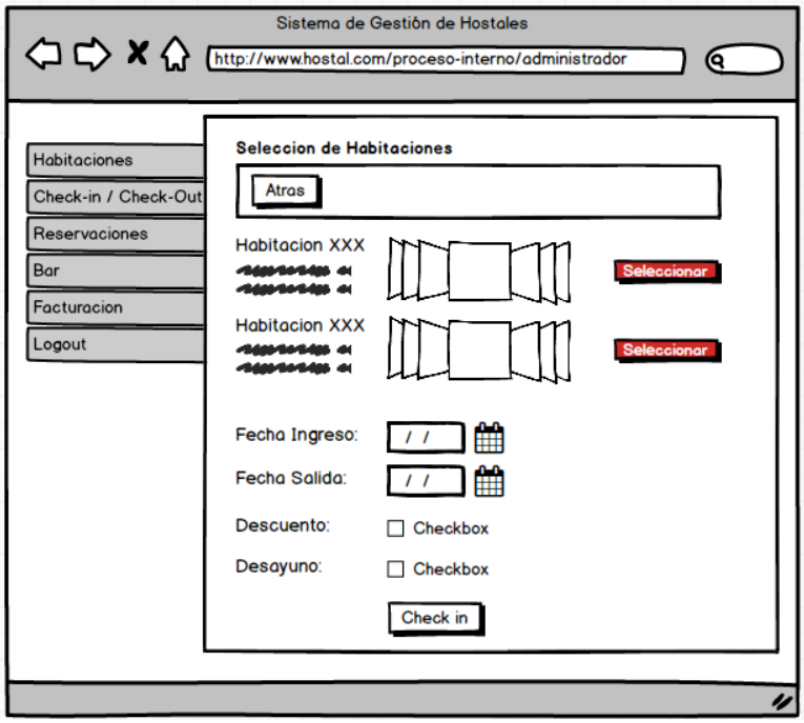


Ilustración 37 Prototipo Proceso Check-in, Selección de Habitaciones (Esparza, 2018)

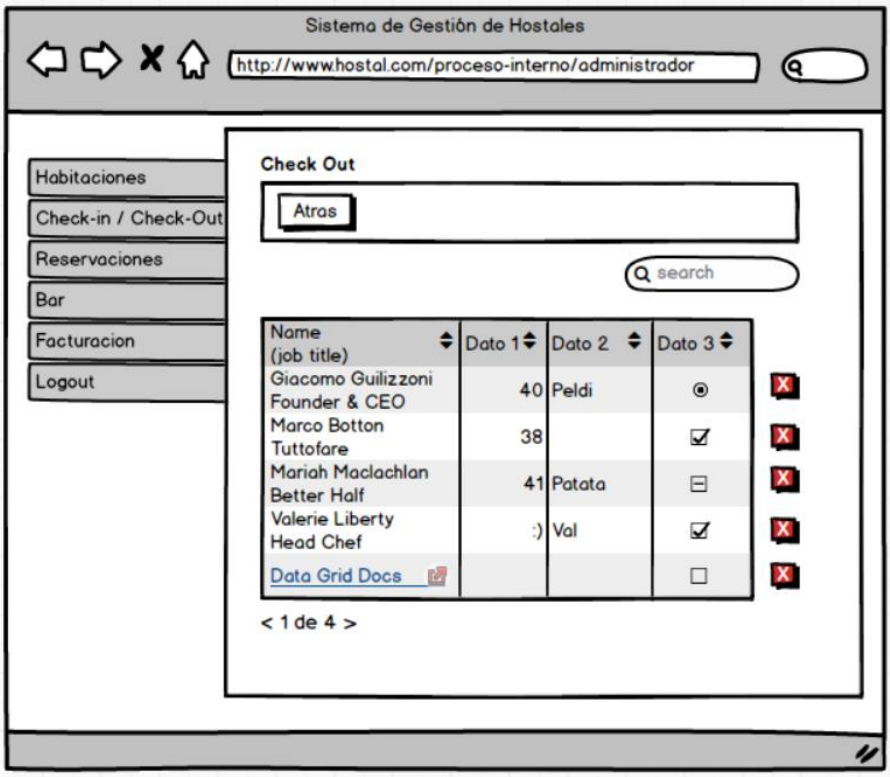


Ilustración 38 Prototipo Proceso Check-out (Esparza, 2018)

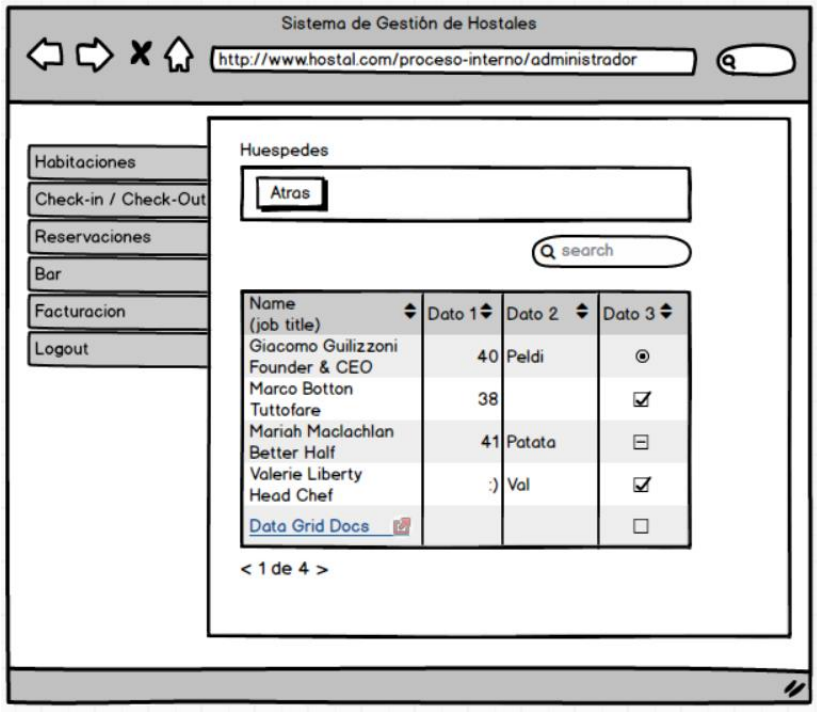


Ilustración 39 Prototipo Proceso Check-in / Check-out Listado de Huéspedes (Esparza, 2018)

Para el proceso de Reservaciones, se proponen los mismos prototipos que en las ilustraciones 38, 34, 35, 36, 37, 39 en ese orden respectivamente.

3.2.3 Codificación

Historia de Usuario 4

Para el presente módulo se van a codificar varias pantallas, algunas se reutilizarán, pero con un flujo de proceso alterno:

1. Pantalla principal Check-in / Check-out

Para la creación de la Pantalla principal Check-in / Check-out fue necesario la creación de los siguientes ítems:

Paquetes

- Creación de dos carpetas Check-in y Check-out para agrupar los diferentes procesos
- Creación de un archivo dentro de la carpeta “Check-in” de la vista: “index”

Clases

- Ninguna clase adicional fue creada para esta vista

Métodos

- Para la vista se necesita un método que retorne únicamente la vista.

```
public function Index()
{
    return \View::make('hms.modulos.check-in.index');
}
```

Ilustración 40 Código Controlador Pantalla principal Check-in / Check-out (Esparza, 2018)

Rutas

- Se creó una ruta:
    - /status-huespedes

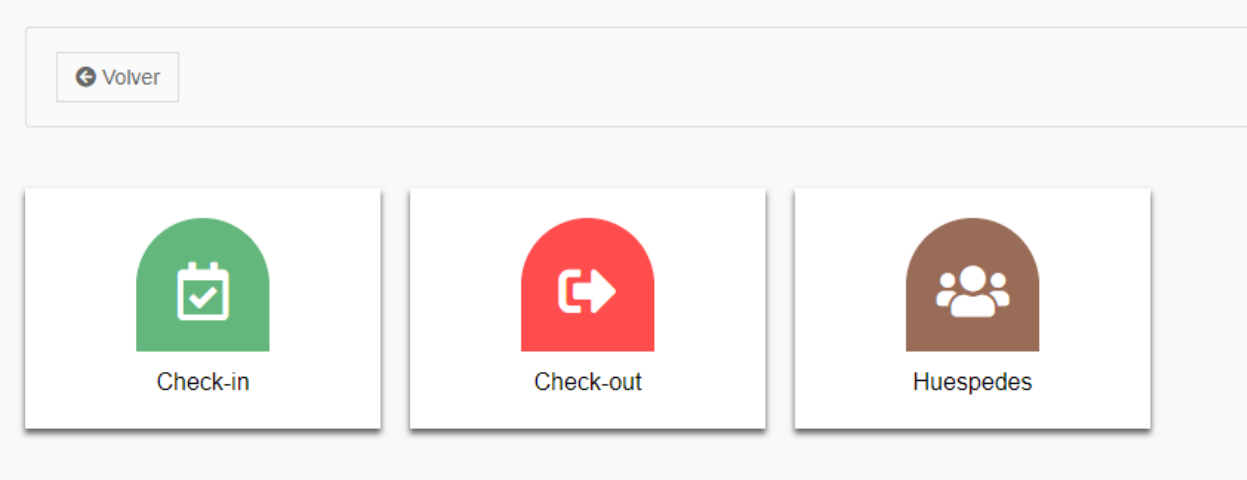


Ilustración 41 Pantalla principal Check-in / Check-out (Esparza, 2018)

2. Check-in / Check-out y Listado de Huéspedes

Para la creación de Check-in, Check-out y Listado de huéspedes fue necesario la creación de los siguientes ítems:

Paquetes

- Creación de los siguientes archivos dentro de la carpeta “Check-in”:
    - Check-in
    - Cliente-nuevo
    - Cliente-frecuente
    - Cliente
    - Huéspedes

    - Listado-reservas
    - Listado
    - Selección-habitación
- Creación del siguiente archivo dentro de la carpeta “Check-out”:
    - Check-out

Clases

- Las clases o Modelos creados para las funcionalidades mencionadas fueron:
    - Cliente, contiene la información de los clientes
    - Huesped, contiene la información de los tipos de huéspedes para el hostal
    - HuespedxCliente, contiene la información de los acompañantes de cada cliente
    - Alquiler, contiene la información de todos los alquileres que se realizaron en el negocio
    - Paises, todos los países del mundo para referenciar la nacionalidad de cada cliente.
    - Descuento, Contiene todos los descuentos que pueda ofrecer el negocio.

Métodos

Los métodos necesarios que se crearon fueron los siguientes:

- Listado()
- Checkin()
- ListadoReservas()
- SinReserva()
- ClienteF()

- HuespedesCF()
- saveHuespedes()
- ClienteNuevo()
- saveClient()
- saveRoom()
- CheckOut()
- saveCheckOut()

Rutas

- Se crearon las siguientes rutas necesarias para el proceso dentro del controlador “CheckController”:
    - /check-in
    - /check-in/saveCheckInReservation
    - /check-in/sin-reserva
    - /check-in/sin-reserva/cliente-nuevo
    - /check-in/sin-reserva/saveClient
    - /check-in/sin-reserva/seleccion-habitacion
    - /check-in/sin-reserva/saveRoom
    - /check-in/sin-reserva/cliente-frecuente
    - /check-in/sin-reserva/huespedes
    - /check-in/sin-reserva/saveHuespedes
    - /check-in/sin-reserva/cliente-frecuente/seleccion-habitacion
    - /check-in/sin-reserva/cliente-frecuente/saveRoom
    - /check-out
    - /saveCheckOut
    - /listado

En el proceso de Check-in el usuario podrá seleccionar si es de una reserva o si es sin reserva. En el caso de ser una reserva selecciona la reserva de una lista, asigna la cantidad de acompañantes del cliente y selecciona las habitaciones y fechas que se va a hospedar. En el caso de hacer Check-in de un cliente sin reserva, puede seleccionar si es un cliente frecuente o cliente nuevo, posterior a esto asigna los acompañantes y selecciona las habitaciones y fechas de hospedaje. Los datos requeridos que el controlador “CheckController” va a solicitar serán distintos dependiendo de la pantalla que se encuentre el usuario, como se muestra a continuación:

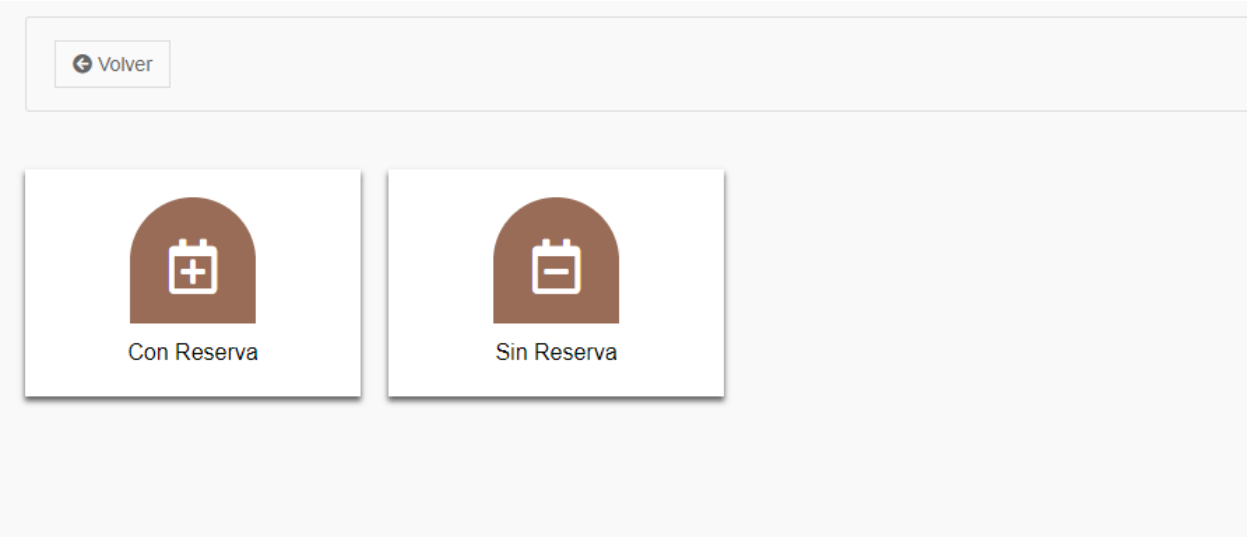


Ilustración 42 Pantalla Check-in Con o Sin Reserva (Esparza, 2018)

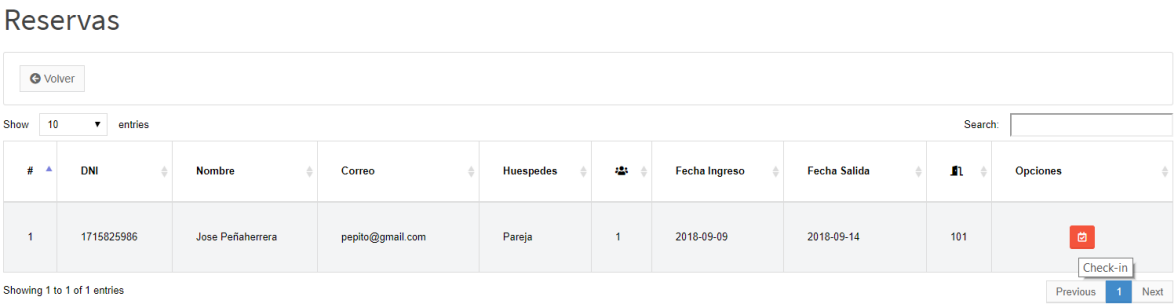


Ilustración 43 Pantalla Check-in Listado de Reservas (Esparza, 2018)

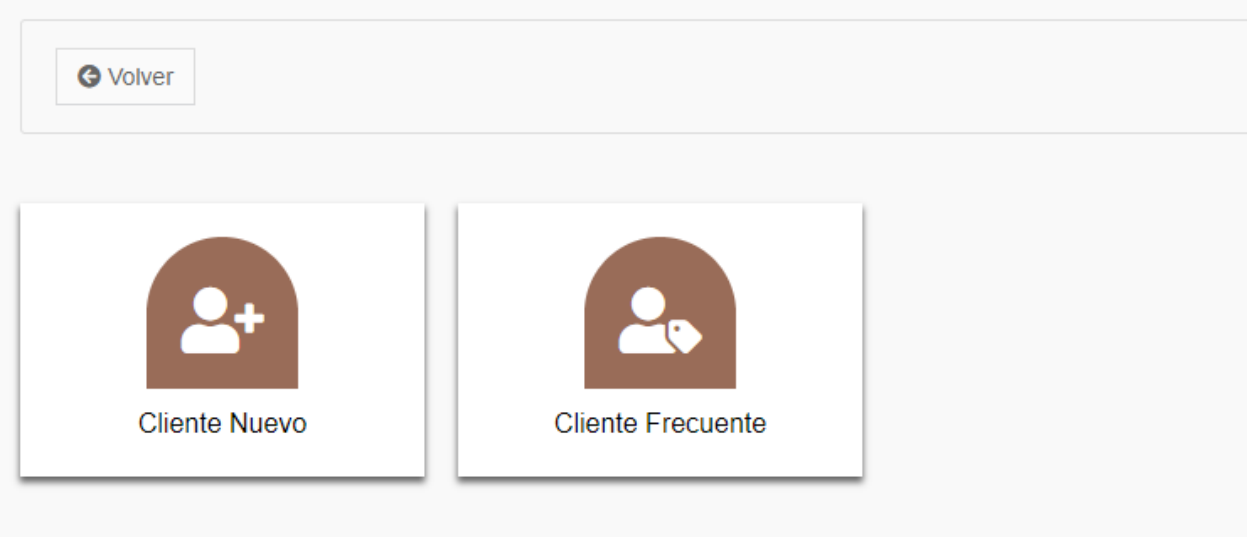


Ilustración 44 Pantalla Check-in Cliente Nuevo o Frecuente (Esparza, 2018)

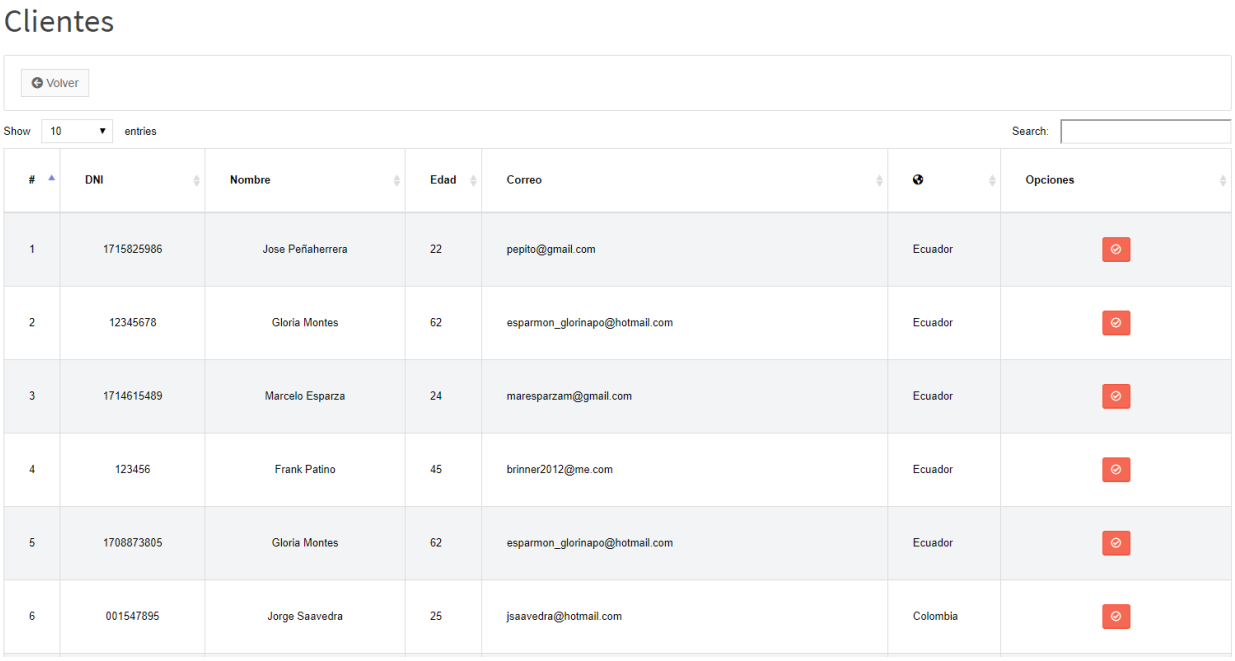


Ilustración 45 Pantalla Check-in Listado de Clientes Frecuentes (Esparza, 2018)

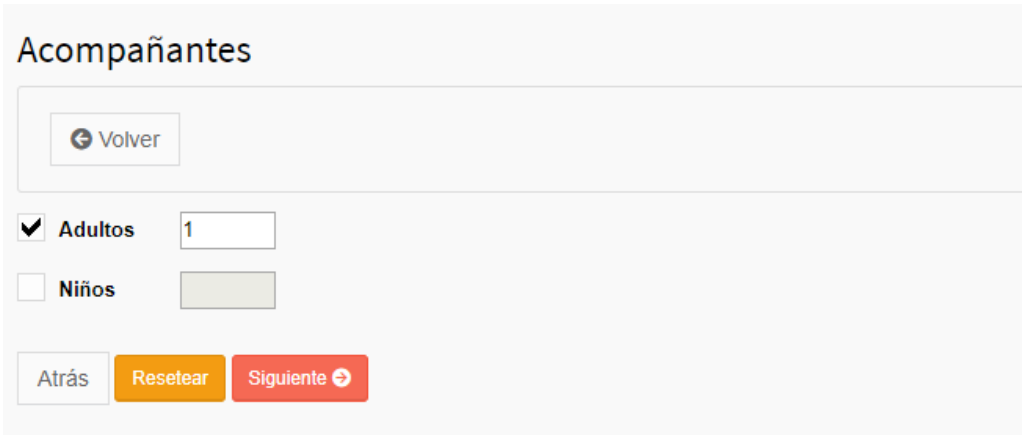


Ilustración 46 Pantalla Check-in Acompañantes (Esparza, 2018)

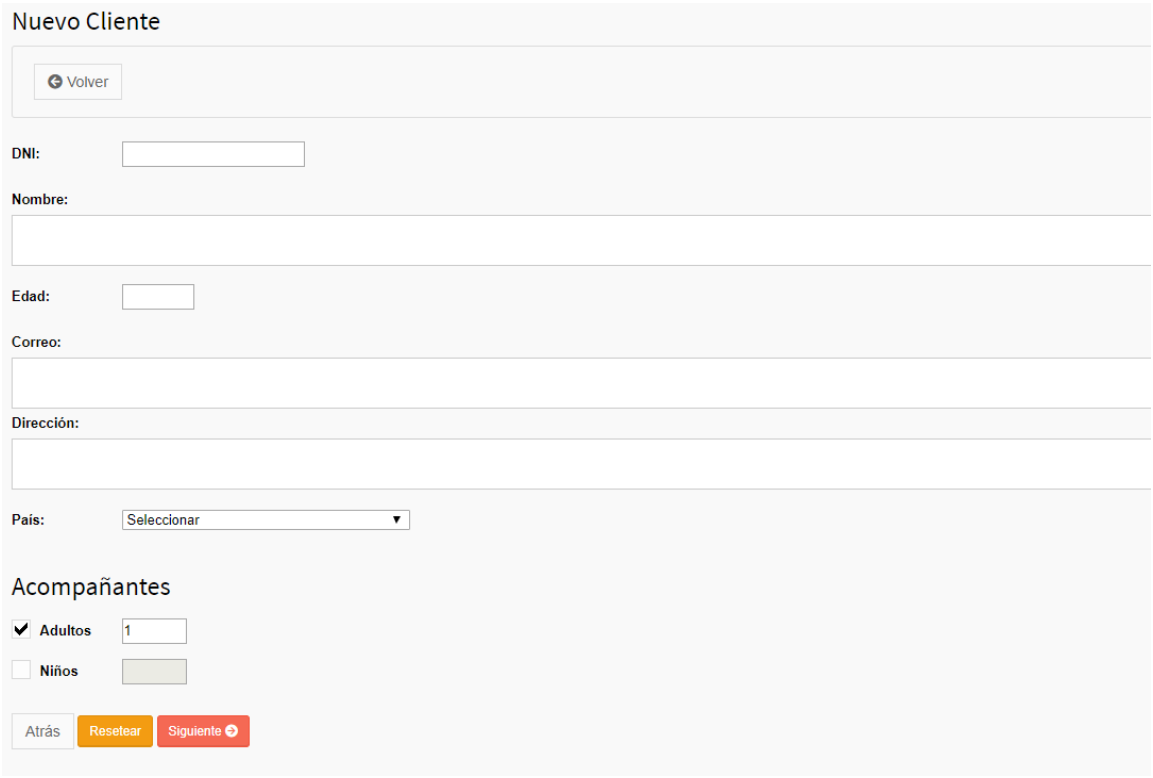


Ilustración 47 Pantalla Check-in Cliente Nuevo (Esparza, 2018)

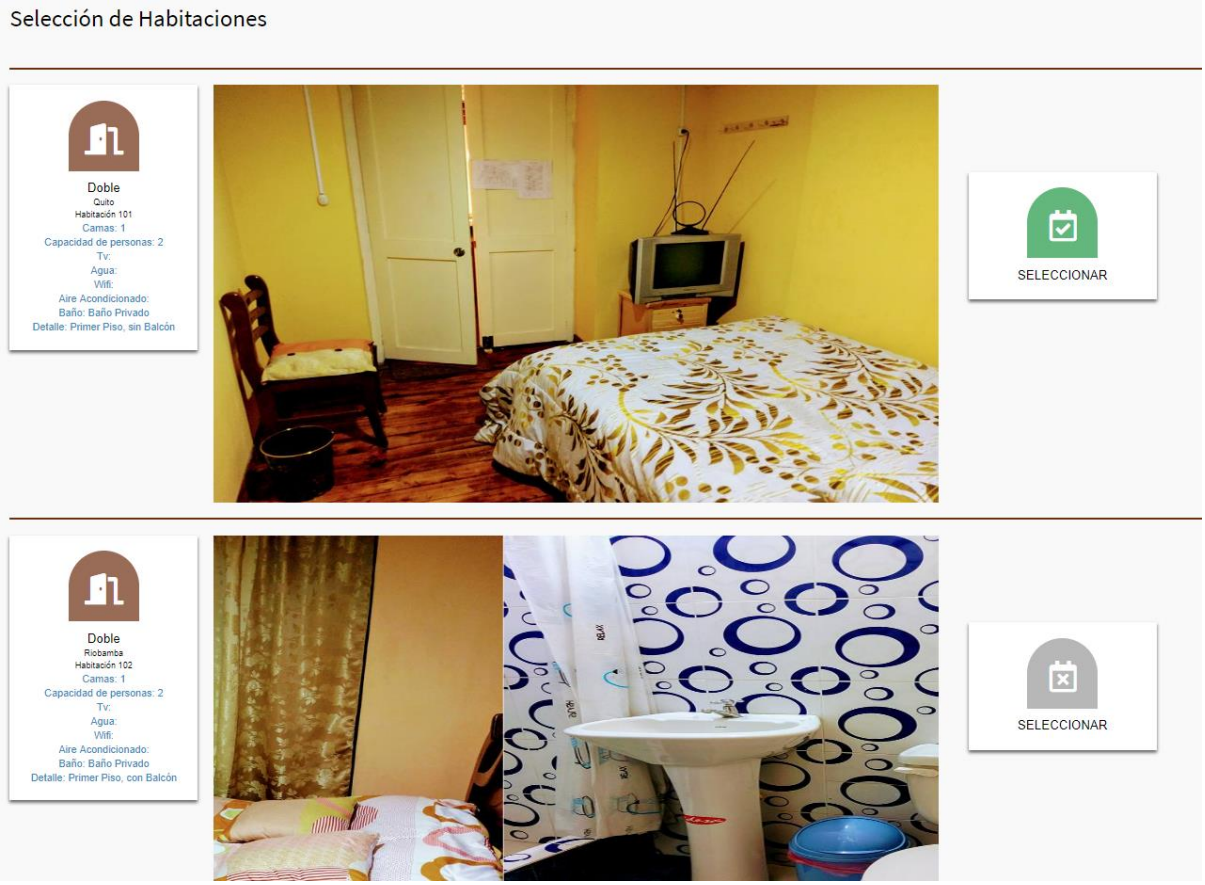


Ilustración 48 Pantalla Check-in Selección de Habitaciones (Esparza, 2018)



Ilustración 49 Pantalla Check-in Fechas de estadía (Esparza, 2018)

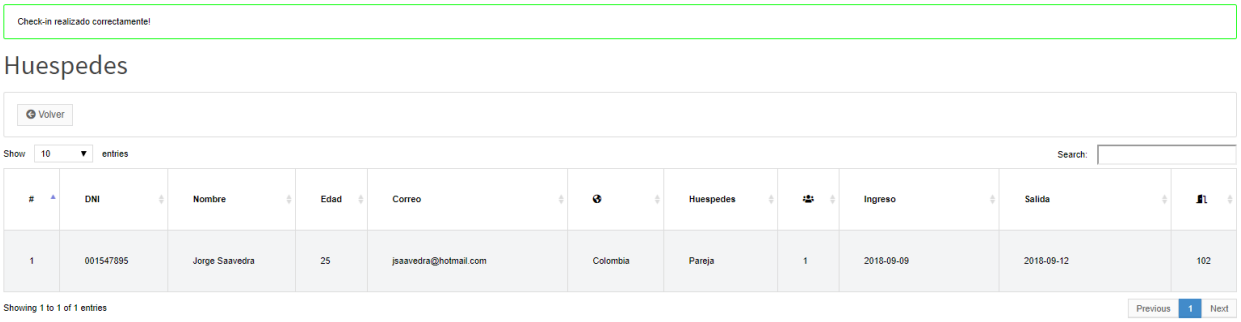


Ilustración 50 Pantalla Check-in Realizado Correctamente (Esparza, 2018)

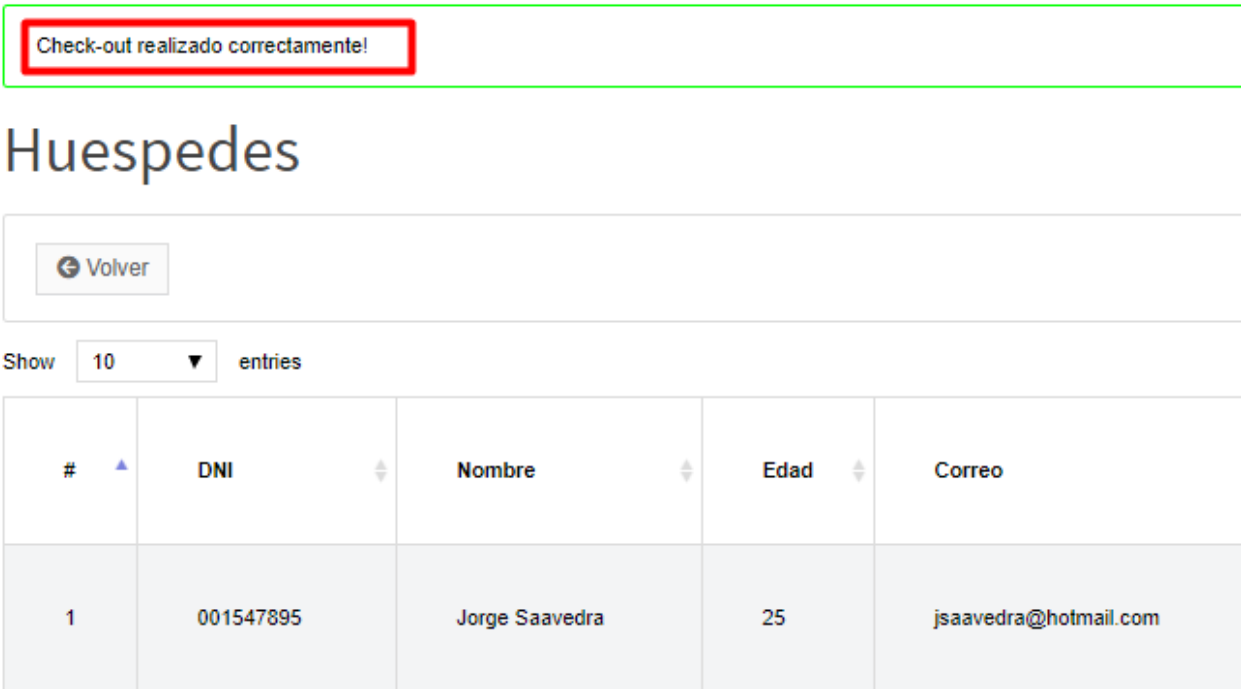


Ilustración 51 Pantalla Check-out Realizado Correctamente (Esparza, 2018)

Historia de Usuario 5

El proceso de reservaciones se equivalente al proceso de Check-in con algunos detalles variados, como, por ejemplo, el controlador no podrá hacer Check-in o Reservas de habitaciones reservadas. La fecha de reserva tiene que ser mínimo un día antes a la fecha de llegada al

negocio. Las funciones en el controlador de Reservas tienen los mismos nombres que las funciones del controlador Check, con las variaciones mencionadas. A Continuación, se detalla todo lo necesario para esta funcionalidad y las pantallas del proceso de reservación:

Paquetes

- Creación de los siguientes archivos dentro de la carpeta “Reservacion”:
    - Cliente-nuevo
    - Cliente-frecuente
    - Cliente
    - Huéspedes
    - Listado
    - Selección-habitación

Clases

- Ninguna clase o modelo adicional fue necesario crear ya que en el proceso de Check-in se crearon todas las necesarias para el proceso de reservación.

Métodos

Los métodos necesarios que se crearon fueron los siguientes:

- Listado()
- Cliente()
- ClienteF()
- HuespedesCF()
- saveHuespedes()
- ClienteNuevo()

- saveClient()
- saveRoom()
- CheckInReservation()

Rutas

- Se crearon las siguientes rutas necesarias para el proceso dentro del controlador “ReservaController”:
    - /listado
    - /cliente
    - /cliente-nuevo
    - /saveClient
    - /seleccion-habitacion
    - /saveRoom
    - /cliente-frecuente
    - /huespedes
    - /saveHuespedes
    - /cliente-frecuente/seleccion-habitacion
    - /cliente-frecuente/saveRoom

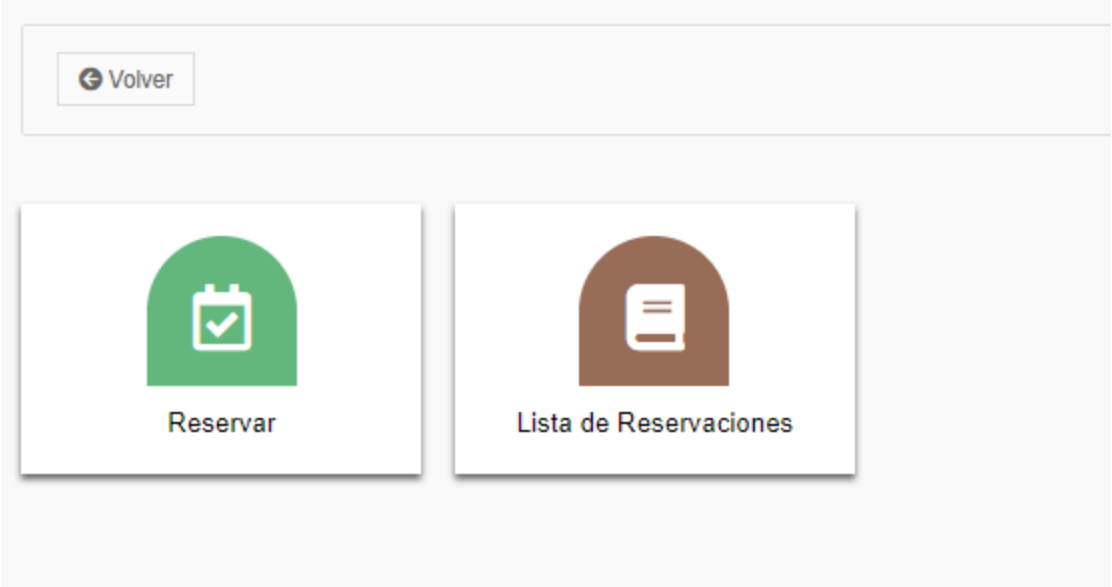


Ilustración 52 Pantalla principal Reservas (Esparza, 2018)

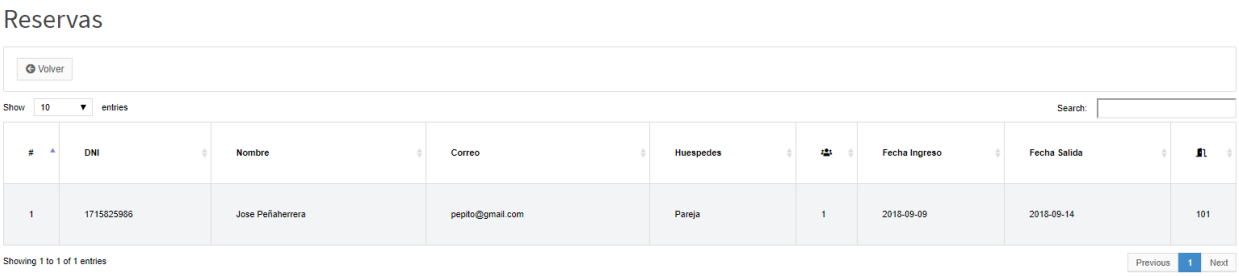


Ilustración 53 Pantalla Listado de Reservas (Esparza, 2018)

3.2.4 Pruebas

Caja blanca: Prueba Unitaria

Se comprueba que los módulos Check-in / Check-out y Reservaciones no retornan ningún error, no posee ningún enlace roto y todos los enlaces externos de librerías funcionan correctamente, la consola del sistema está siempre limpia sin errores en todas las pantallas.

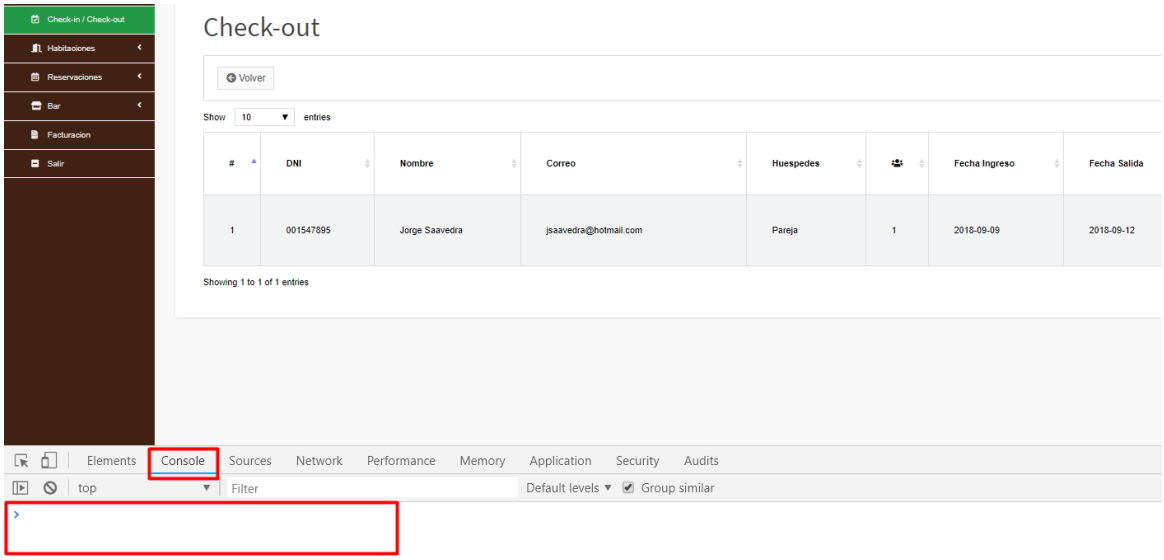


Ilustración 54 Pruebas Check-out Caja Blanca Sin errores (Esparza, 2018)



Ilustración 55 Pruebas Check-in Caja Blanca Sin errores (Esparza, 2018)

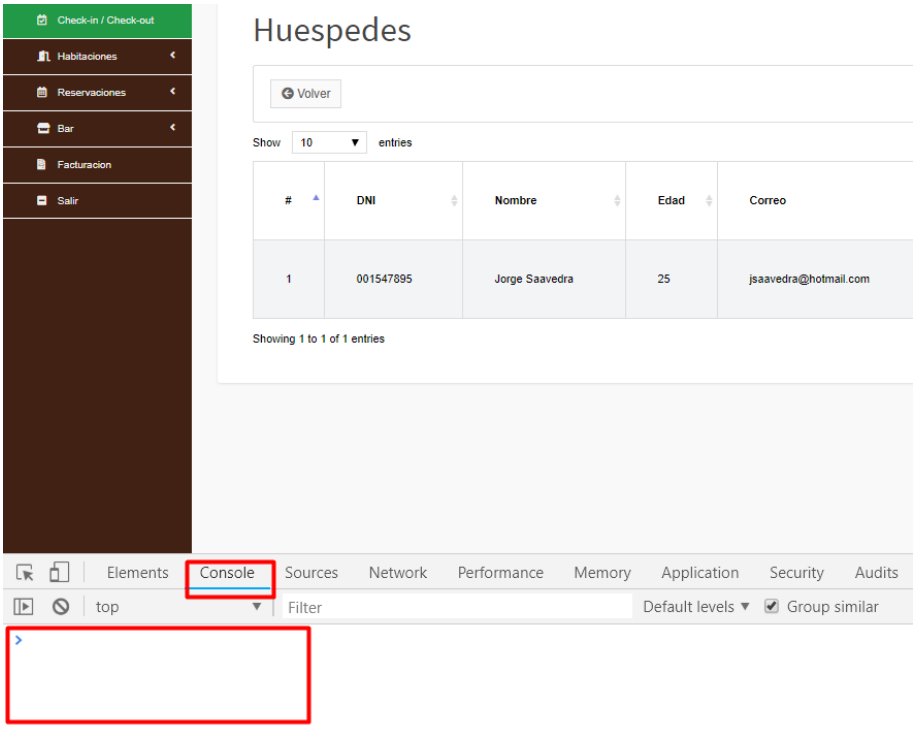


Ilustración 56 Pruebas Listado de Huéspedes Caja Blanca Sin errores (Esparza, 2018)

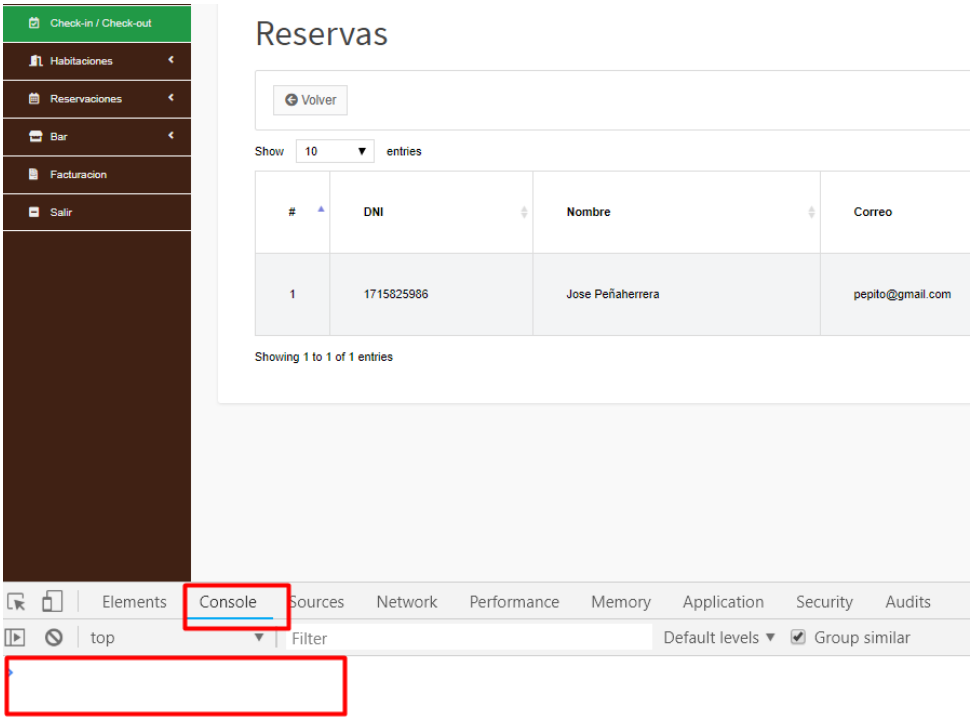


Ilustración 57 Pruebas Listado de Reservas Caja Blanca Sin errores (Esparza, 2018)

Caja Negra: Criterios de Aceptación

- Ingreso de nuevo cliente exitoso
- Check-in exitoso
- Check-out exitoso
- Reservación exitosa
- Páginas de fácil manejo para el usuario
- Información de huéspedes y reservas clara y de fácil visualización

PRUEBA DE ACEPTACIÓN			
Número de prueba:	2	Número de historia de usuario:	4 5
Nombre de Caso de Prueba:	Usuario del Sistema		
Descripción			
Se evaluará la funcionalidad y la experiencia del usuario al manejar los modulos 4 y 5 del sistema.			
Condiciones de ejecución			
Crear por lo menos un cliente frecuente, y una reserva directamente a la base de datos. Conectarse a internet.			
Entradas			
Formulario de Cliente Nuevo: Datos del cliente. Formulario para check-in: Habitación, fechas de estadía descuento y desayuno. Formulario de acompañantes de un cliente frecuente: Cantidad de adultos y niños.			
Resultado esperado:	Cliente Nuevo creado, Check-in / Check-out, Reservación, Presentar y Cargar la información		
Evaluación	Control de selección de habitaciones con menor capacidad de la cantidad de huéspedes que ingresan		
	Registrar en la base de datos el usuario que hizo algun movimiento		
	Agregar mensaje del significado de la iconografía usada		

Ilustración 58 Pruebas Caja Negra Segunda Prueba (Esparza, 2018)

3.3 Tercera Iteración

En el tercer ciclo de desarrollo se va a codificar los módulos Bar y Facturación, además de aplicar la evaluación obtenida del segundo ciclo de desarrollo.

3.3.1 Planificación

Historia de Usuario 6

Se desarrollará el módulo “Bar” el cual gestionará las compras, las ventas, stock e inventario de todos los productos que ofrecerá el negocio. Las compras permitirán comprar más de un producto disponible en el sistema o la compra de un nuevo producto. Las ventas permitirán al cliente final comprar los productos que tengan stock del negocio. La sección stock presentará todos los productos que tengan stock mayor o igual a 1 y finalmente el inventario presentará todos los productos con un gráfico estadístico para conocer el estado de cada uno.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
6	Usuario del sistema	Requiere de un manejo de las compras, ventas e inventario del bar del hostal.	Conocer los productos que hay en stock, compra y venta de los productos y un inventariado para tener un control de un stock continuo de productos.	1	Compras	Boton "compras" para comprar un nuevo producto o uno existente para aumentar su stock.	Click en "Comprar" y llenado de datos o ingreso de una cantidad	Ingreso exitoso de los datos del producto o compra exitosa de las cantidades de cada producto
				2	Ventas	Boton "ventas" para venta de productos al Cliente final.	Click en "Vender" e ingreso de una cantidad de cada producto	Venta de la cantidad de productos solicitados exitosa
				3	Listado de productos en Stock	Boton "Stock" presenta una lista de los productos y su stock	Click en "Stock"	Despliegue en una tabla todos los productos con stock.
				4	Inventario de todos los productos que ofrece el hostal	Boton "inventario" para conocer estado de cada producto del negocio	Click en "Inventario"	Despliegue en una tabla todos los productos del negocio y su stock.

Ilustración 59 Historia de Usuario 6 (Esparza, 2018)

Tareas para realizar:

- Definición del proceso de Compra
- Definición del proceso de Venta
- Definición de un gráfico estadístico para la sección de inventario
- Codificación de todo el proceso de Bar
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 3 a 4 días

Historia de Usuario 7

Se desarrollará el módulo “Facturación” el cual se encargará de recolectar todos los consumos de cada cliente en el hostal, incluyendo las habitaciones que alquiló. Después de ello se procede a emitir el comprobante y se entrega al cliente final. Aquí termina el ciclo de vida de un Cliente en un negocio de un hostal.

Enunciado de la historia				Criterios de Aceptación				
Numero de Historia	*Rol*	*Característica / Funcionalidad*	*Razón / Resultado*	*Escenario*	*Titulo*	*Contexto*	*Evento*	*Resultado*
7	Usuario del sistema	Requiere de recolectar todo lo consumido por el cliente y generar una Factura	Conocer que es todo lo que ha consumido cada cliente incluyendo las habitaciones que se le asigno y emitir una factura a la salida del negocio.	1	Facturar	Selección de cliente a facturar y un boton "Facturar"	Seleccionar cliente y click en "Facturar"	Despliegue de la factura con todos los datos personales del cliente y su
				2	Factura	Boton "Imprimir" para la impresión del comprobante	Verificar datos correctos y click en "Finalizar" e imprimir el comprobante	Datos correctos, procede a impresión del comprobante

Ilustración 60 Historia de Usuario 7 (Esparza, 2018)

Tareas para realizar:

- Definición de la estructura del comprobante de factura
- Codificación del módulo de Facturación
- Prueba unitaria
- Prueba de aceptación

Tiempo:

- De 2 a 3 días

3.3.2 Diseño

El prototipo para el módulo de Bar tendrá 4 botones para realizar los posibles escenarios que son la venta, compra, stock e inventario de los productos que existen en el negocio. La venta y compra presentarán una lista de los productos y permitirán seleccionar la cantidad de cada producto que se desea comprar o vender. El stock será una tabla de únicamente los productos que tienen stock y el inventario será una pantalla con un gráfico estadístico que muestre el estado de cada producto del negocio.

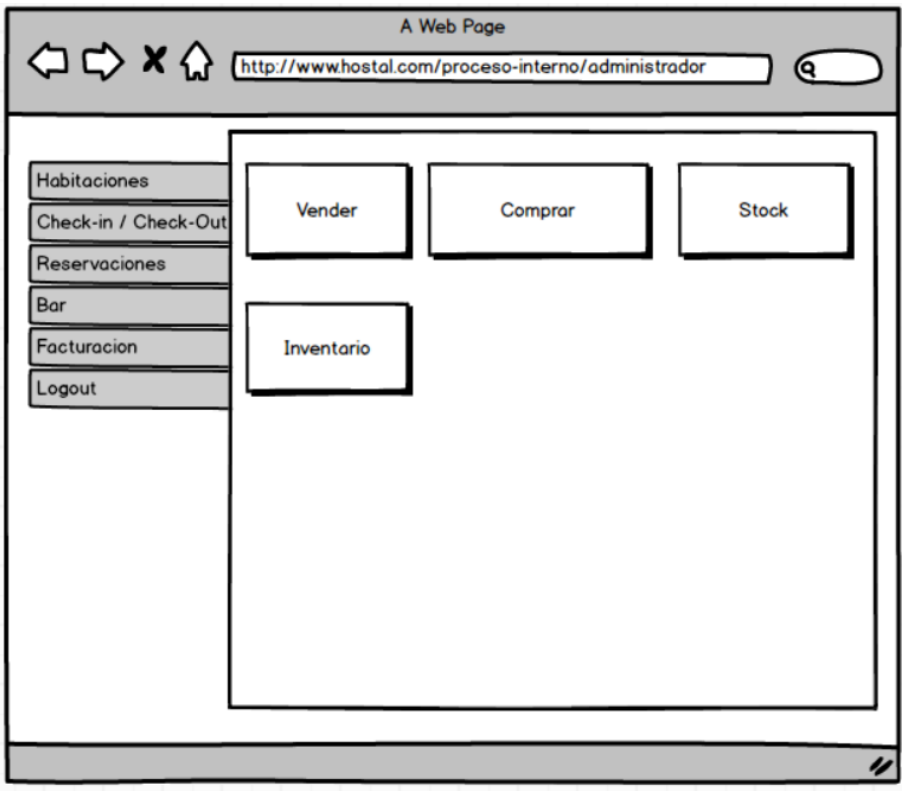


Ilustración 61 Prototipo pantalla principal Modulo Bar (Esparza, 2018)

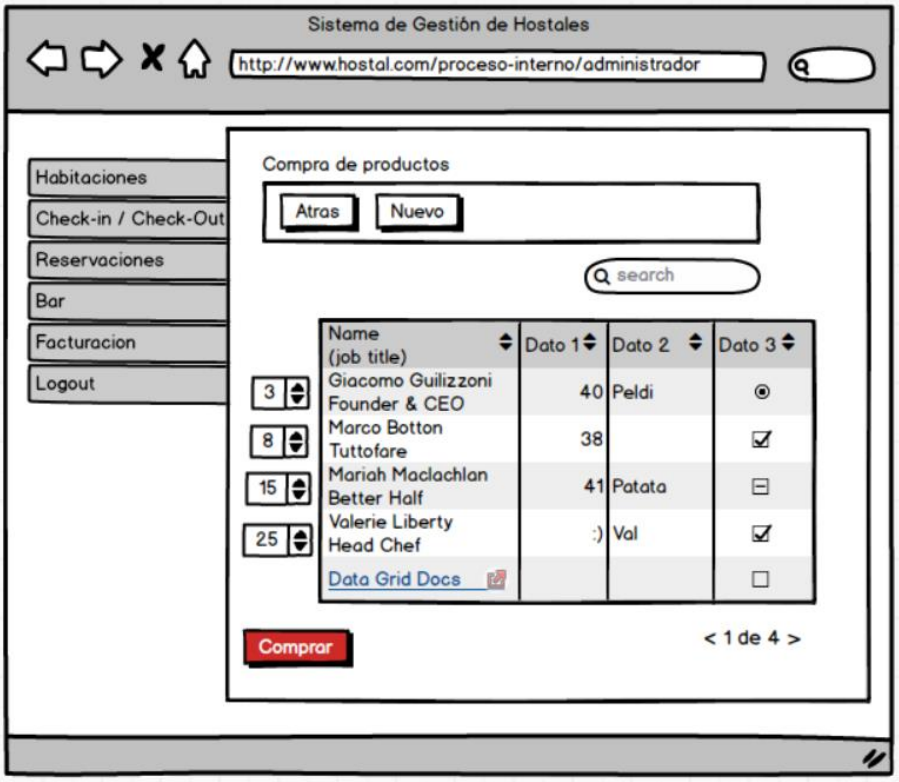


Ilustración 62 Prototipo Pantalla Compra de Productos (Esparza, 2018)

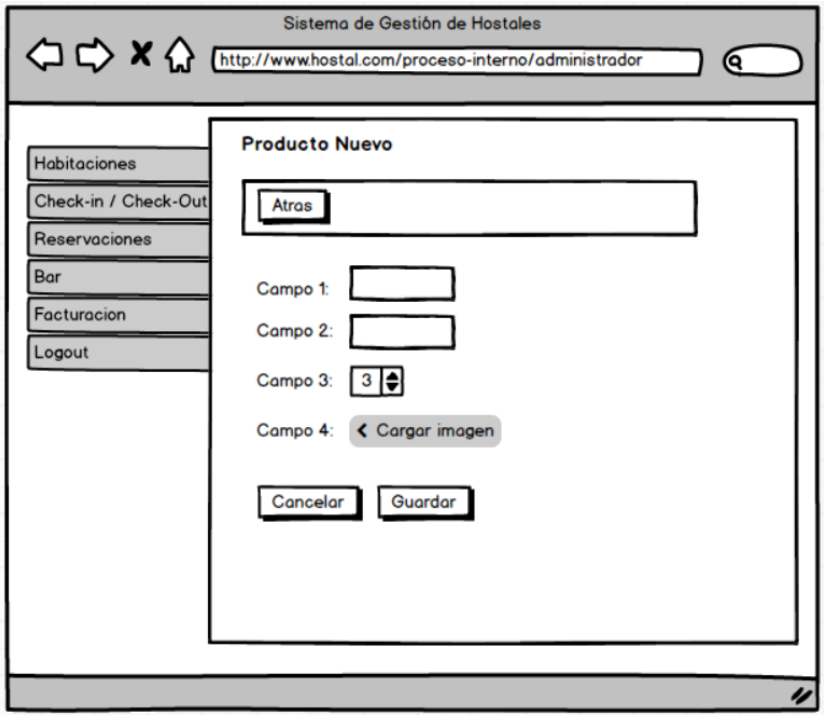


Ilustración 63 Prototipo Pantalla Compra de Nuevo Producto (Esparza, 2018)

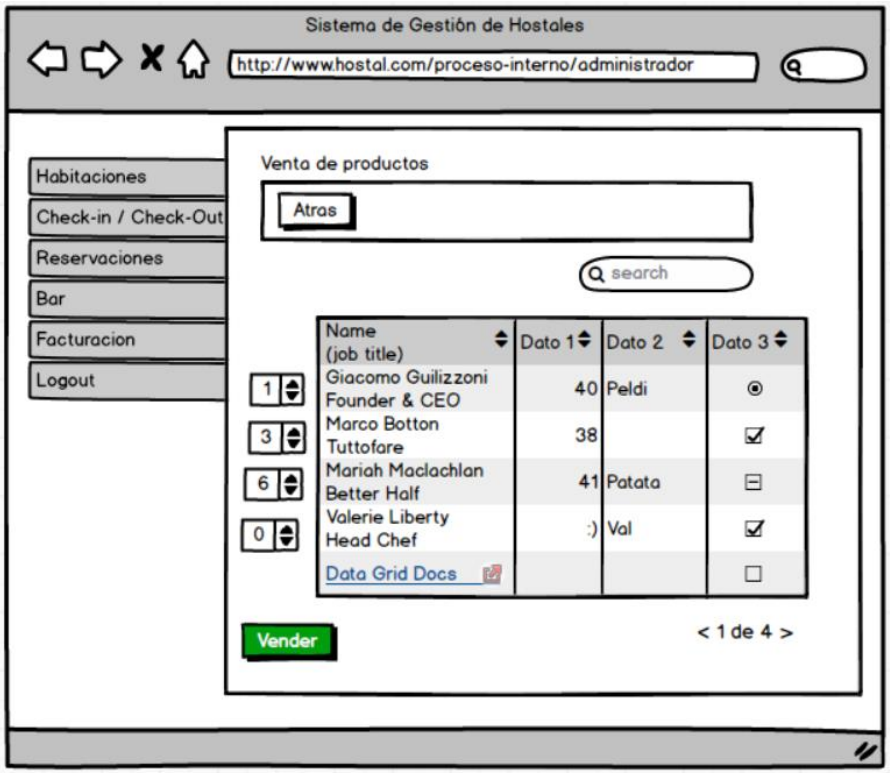


Ilustración 64 Prototipo Pantalla Venta de Productos (Esparza, 2018)

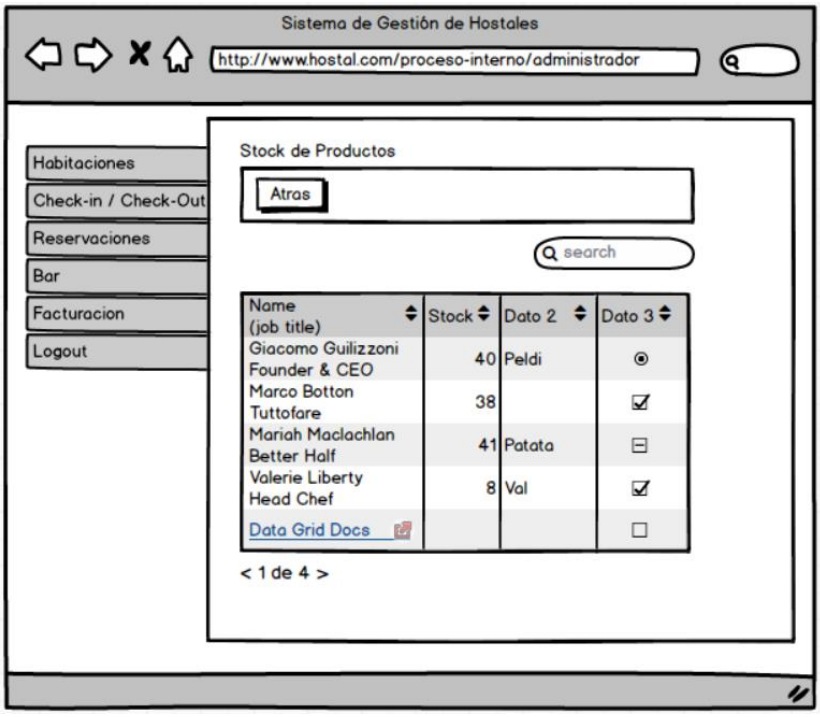


Ilustración 65 Prototipo Pantalla Stock de Productos (Esparza, 2018)

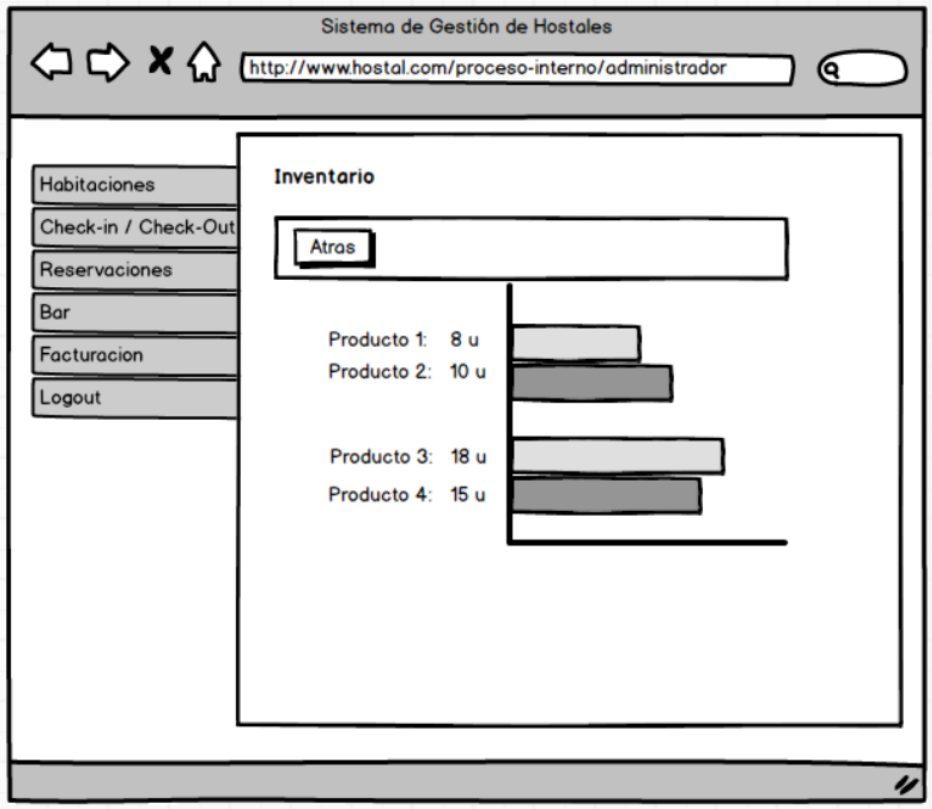


Ilustración 66 Prototipo Pantalla Inventario (Esparza, 2018)

El módulo de facturación se propone desarrollar solo 2 pantallas, una en la que el usuario selecciona al Cliente que va a Facturar y otra en la cual se despliega la Factura junto con un botón que permita imprimir la sección de la factura para entregar al cliente final.

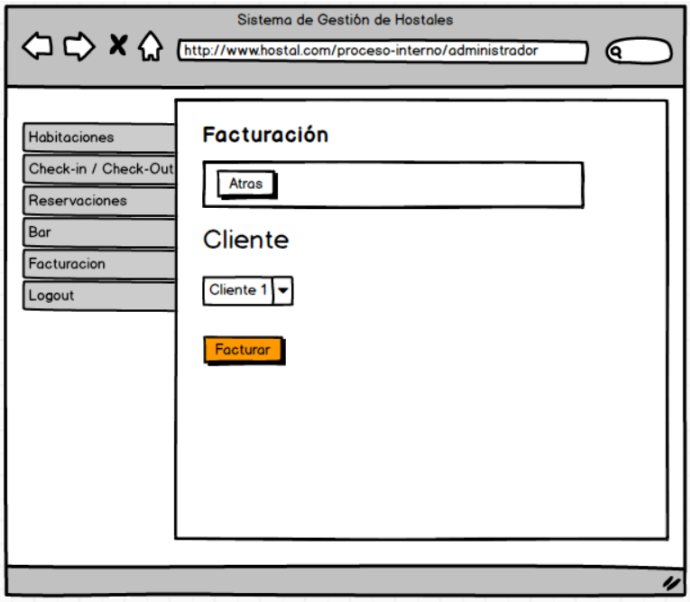


Ilustración 67 Prototipo Pantalla Selección de Cliente a Facturar (Esparza, 2018)

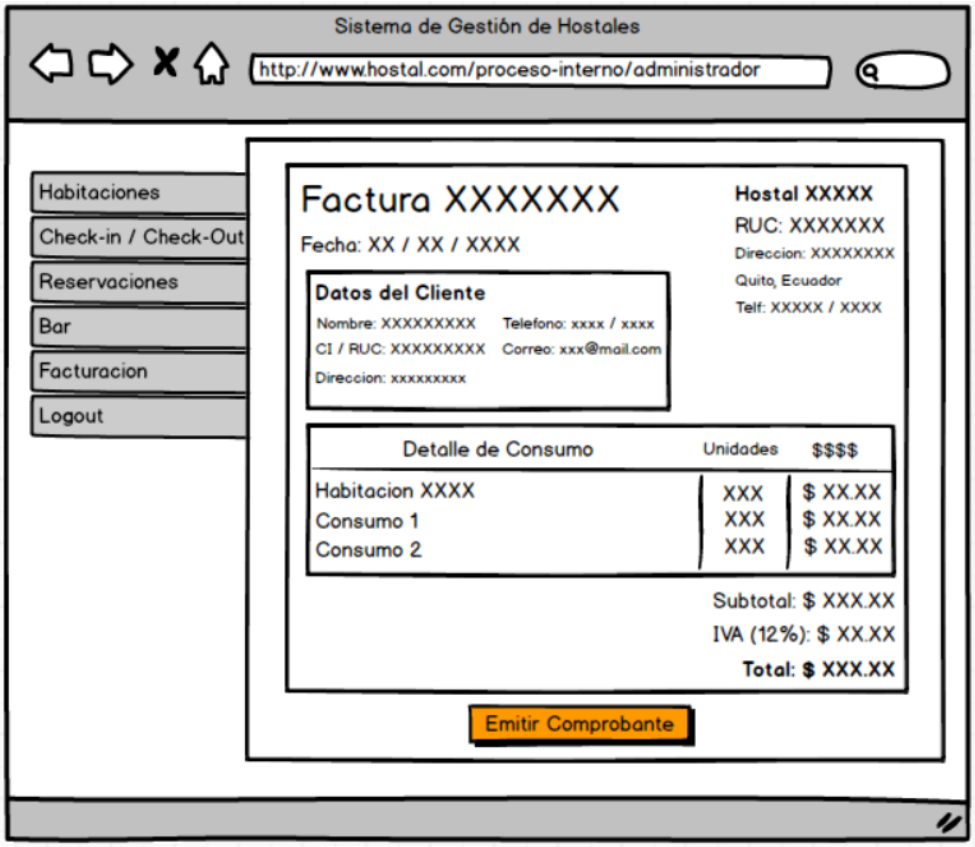


Ilustración 68 Prototipo Pantalla Factura (Esparza, 2018)

3.3.3 Codificación

Historia de Usuario 6

Para la codificación del módulo “Bar” tendremos 4 funcionalidades: Compra, venta, stock e inventario.

El módulo bar necesitó de los siguientes ítems:

Paquetes

- Creación de la carpeta “Bar”
- Creación de los siguientes archivos dentro de la carpeta “Bar”:
    - index
    - stock
    - papelera
    - compra
    - venta
    - inventario
    - editar-producto
    - nuevo-producto

Clases

- Las clases o modelos creados para el presente módulo fueron:
    - Producto
    - ProductoVentas
    - ProductoCompras

Métodos

Los métodos necesarios que se crearon fueron los siguientes:

- Index()
- Inventario()
- Stock()
- Compra()
- buyStuff()
- Venta()
- sellStuff()
- NuevoProducto()
- saveProduct()
- EditarProducto()
- editProduct()
- Papelera()
- deleteRow()
- recoverRow()

Rutas

- Se crearon las siguientes rutas necesarias para el proceso dentro del controlador “BarController”:
    - /stock
    - /inventario
    - /compra
    - /buyStuff
    - /venta
    - /sellStuff

- /nuevo-producto
- /saveProduct
- /editar-producto
- /editProduct
- /papelera
- /eliminar
- /recover

Para el stock de productos, el controlador retorna todos los productos con stock mayor a cero.

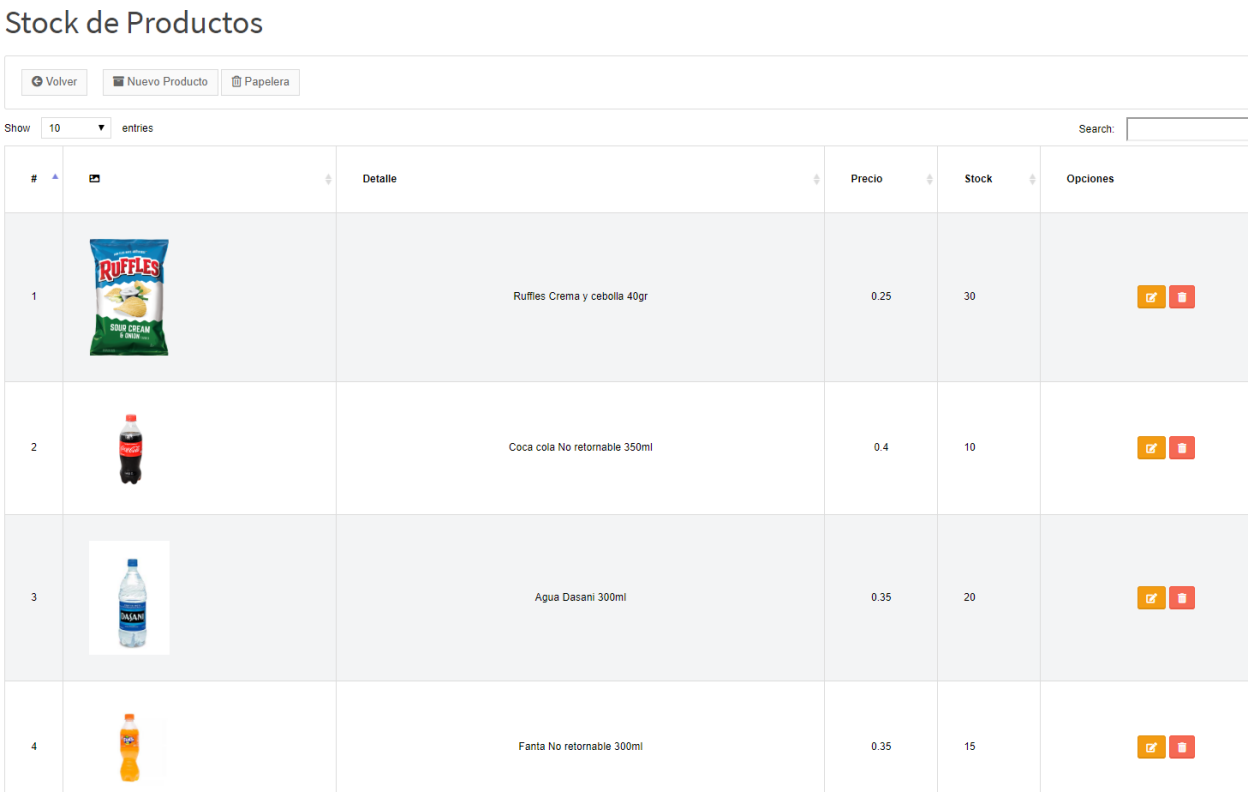


Ilustración 69 Pantalla Stock (Esparza, 2018)

A diferencia del inventario el controlador solicitará todos los productos disponibles.

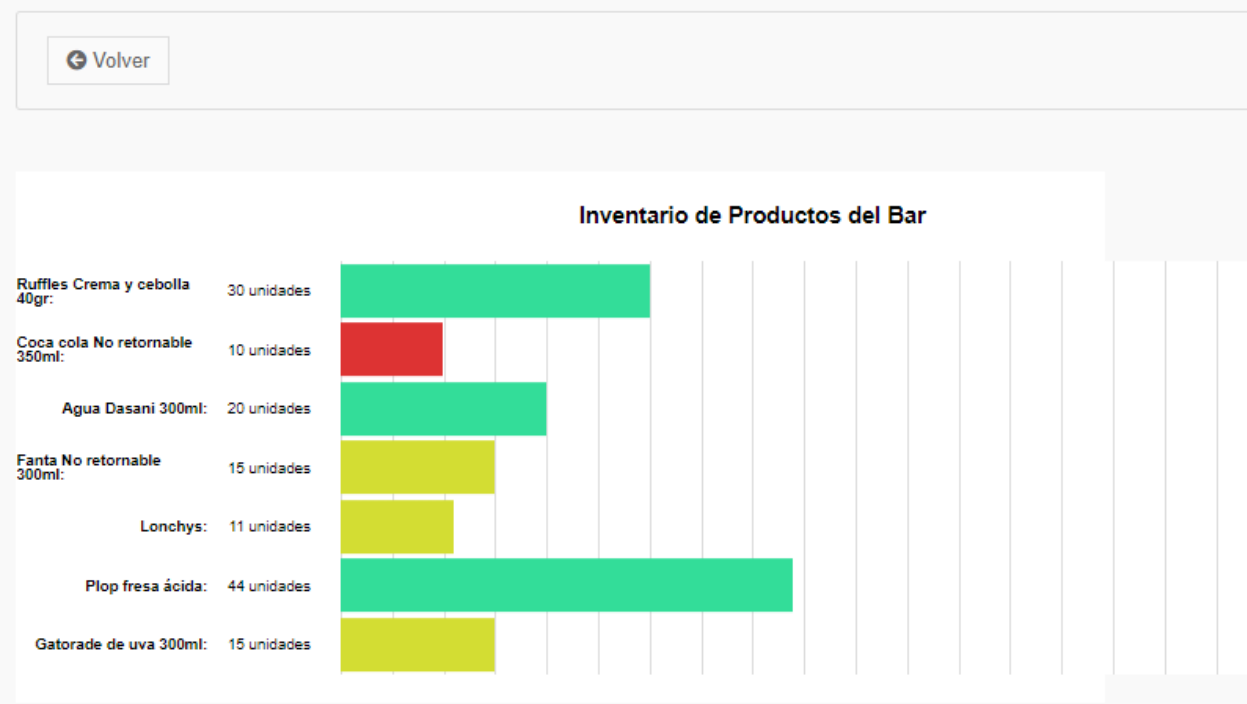


Ilustración 70 Pantalla Inventario (Esparza, 2018)

En el caso de las compras, el controlador presentará una tabla con todos los productos disponibles ordenados desde el producto que menor stock, que se supone se debería comprar primero.

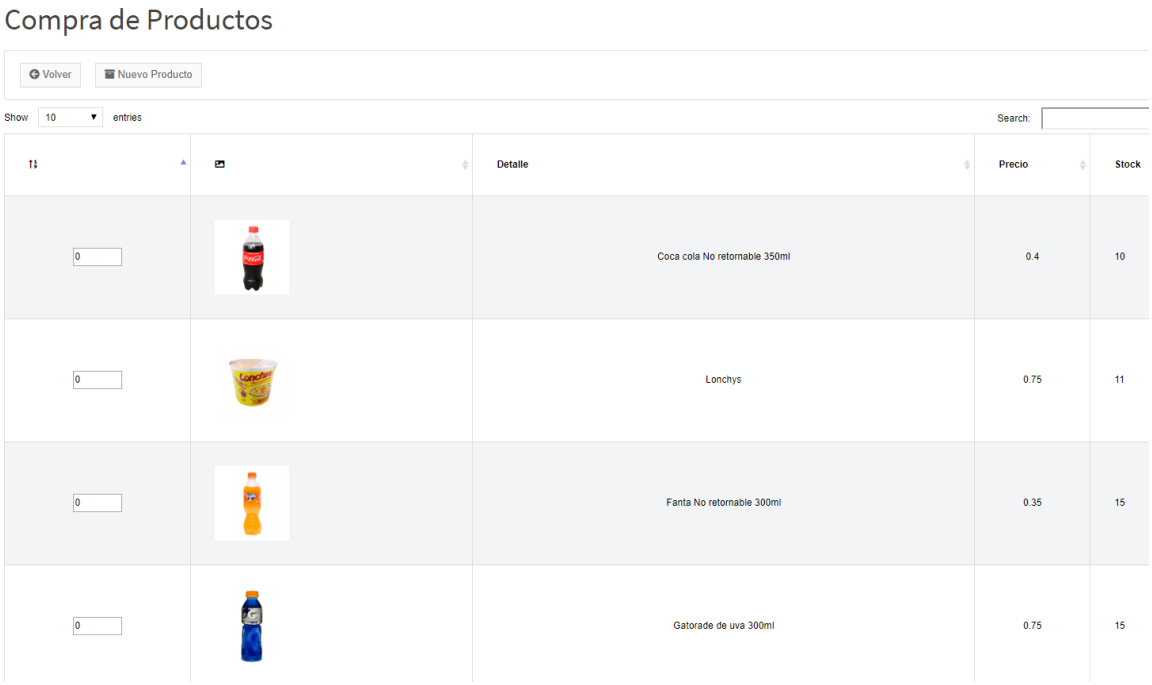


Ilustración 71 Pantalla de Compra de Productos (Esparza, 2018)

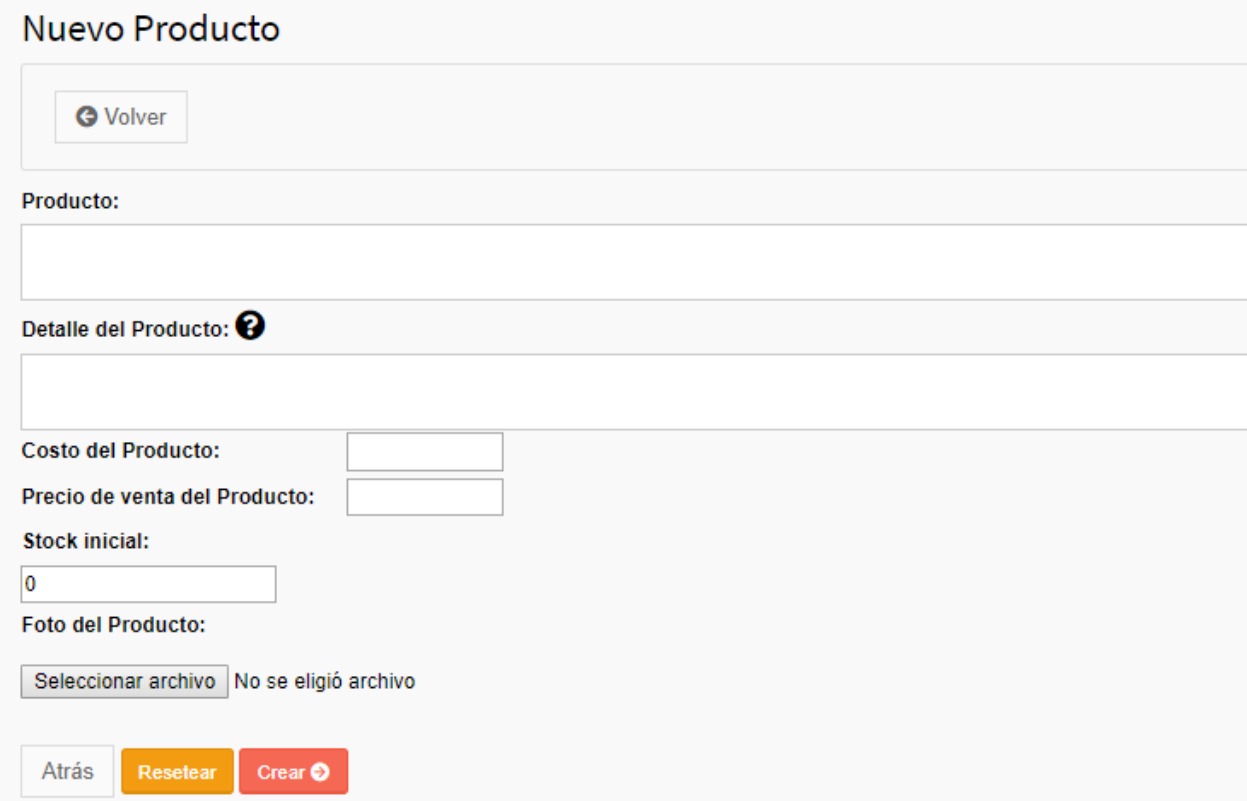


Ilustración 72 Pantalla Compra de Nuevo Producto (Esparza, 2018)

En el proceso de ventas tenemos un proceso equivalente al de compras agregándole el cliente y la habitación del cliente que está comprando los productos al hostal.

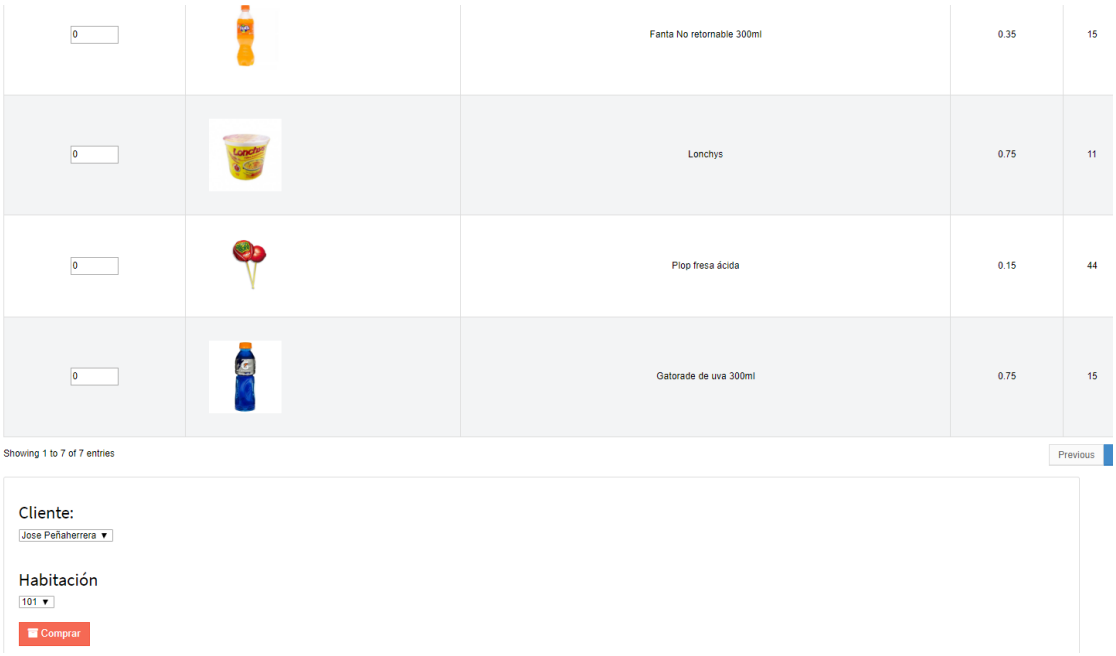


Ilustración 73 Pantalla Venta de Productos (Esparza, 2018)

Historia de Usuario 7

Para la codificación del módulo “Facturación” el controlador retornará todos los clientes que se encuentran actualmente en el hostal hospedados y mediante un botón emite la factura recolectando todos sus consumos incluyendo las habitaciones que alquiló.

El módulo Facturación necesitó de los siguientes ítems:

Paquetes

- Creación de la carpeta “Facturación”
- Creación de los siguientes archivos dentro de la carpeta “Facturación”:
    - factura

- facturar

Clases

- Ninguna clase o modelo adicional fue necesario crear para el módulo de facturación

Métodos

Los métodos necesarios que se crearon fueron los siguientes:

- Index()
- Factura()
- FacturaCli()

Rutas

- Se crearon las siguientes rutas necesarias para el proceso dentro del controlador "FacturaController":
  - /factura
  - /factura/cliente

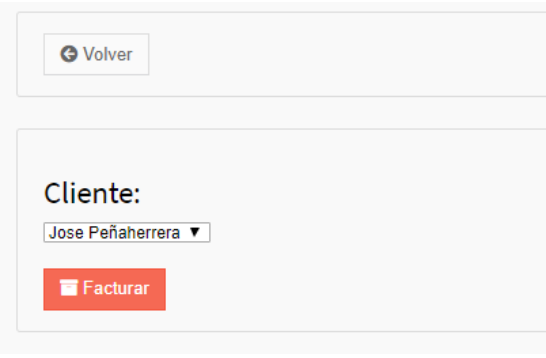


Ilustración 74 Pantalla Facturación - Clientes hospedados (Esparza, 2018)

Factura 00554775

Fecha: 2018-09-09

Hostal Latitud ECUEM
RUC.: 001547214589
Guayaquil y Oriente, Quito, Ecuador
Telf.: XXXX XXXX / XXXX XXXX
latitudecuem@gmail.com

Datos del Cliente

Nombre: Jorge Saavedra
Dirección: Carcelen
Correo: jsaavedra@hotmail.com
CI / RUC: 001547895

Servicios / Productos

Detalle	Costo Unitario	Costo total
Pareja Doble - Habitación 102 (Riobamba)	$ 15.00 1 unidades 1 Noches	$ 15.00
Ruffles Crema y cebolla 40gr Papas	$ 0.25 3 unidades	$ 0.75
Coca cola No retornable 350ml Coca-Cola	$ 0.4 4 unidades	$ 1.6
Plop fresa ácida Dulce	$ 0.15 2 unidades	$ 0.3

Subtotal	$ 17.65
IVA (12%)	$ 2.12
Total	$ 19.77

Gracias por visitarnos
Esperamos tener el gusto de volver a atenderle
~ Latitud ECUEM ~

Ilustración 75 Pantalla Factura (Esparza, 2018)

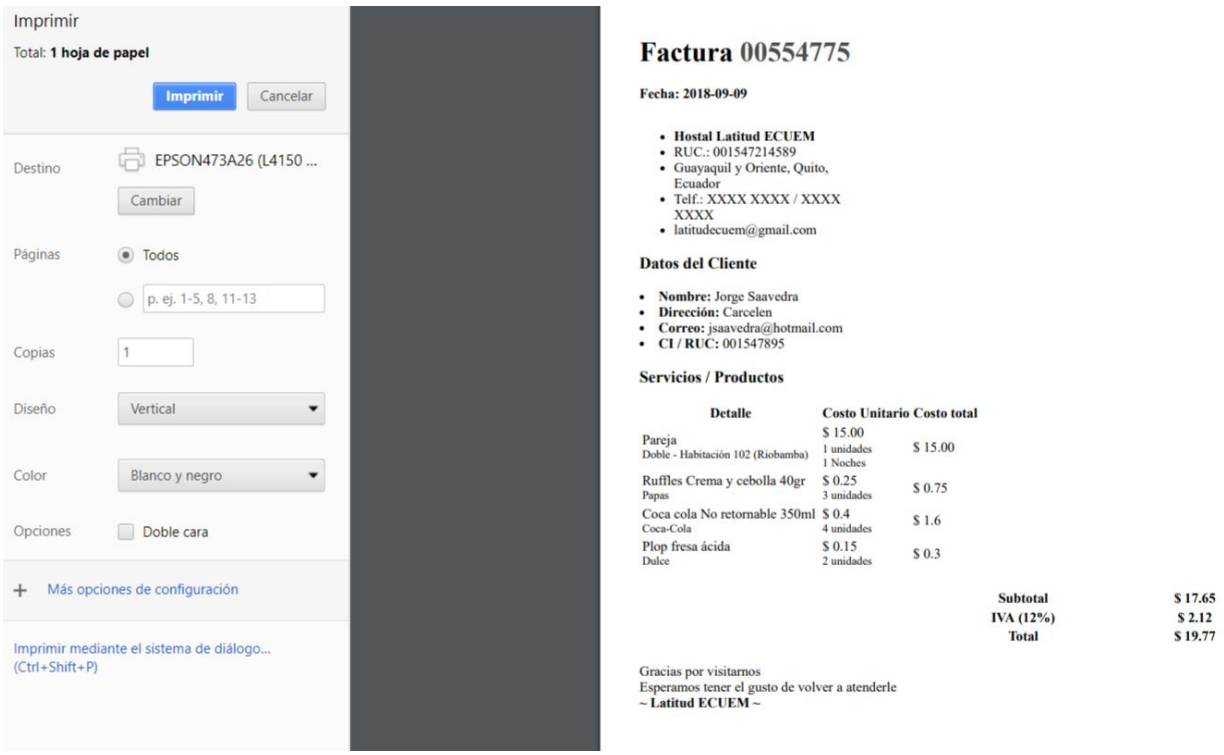


Ilustración 76 Impresión de Factura (Esparza, 2018)

3.3.4 Pruebas

Caja blanca: Prueba Unitaria

Se comprueba que los módulos Bar y Facturación no retornan ningún error, no posee ningún enlace roto y todos los enlaces externos de librerías funcionan correctamente, la consola del sistema está siempre limpia sin errores en todas las pantallas.

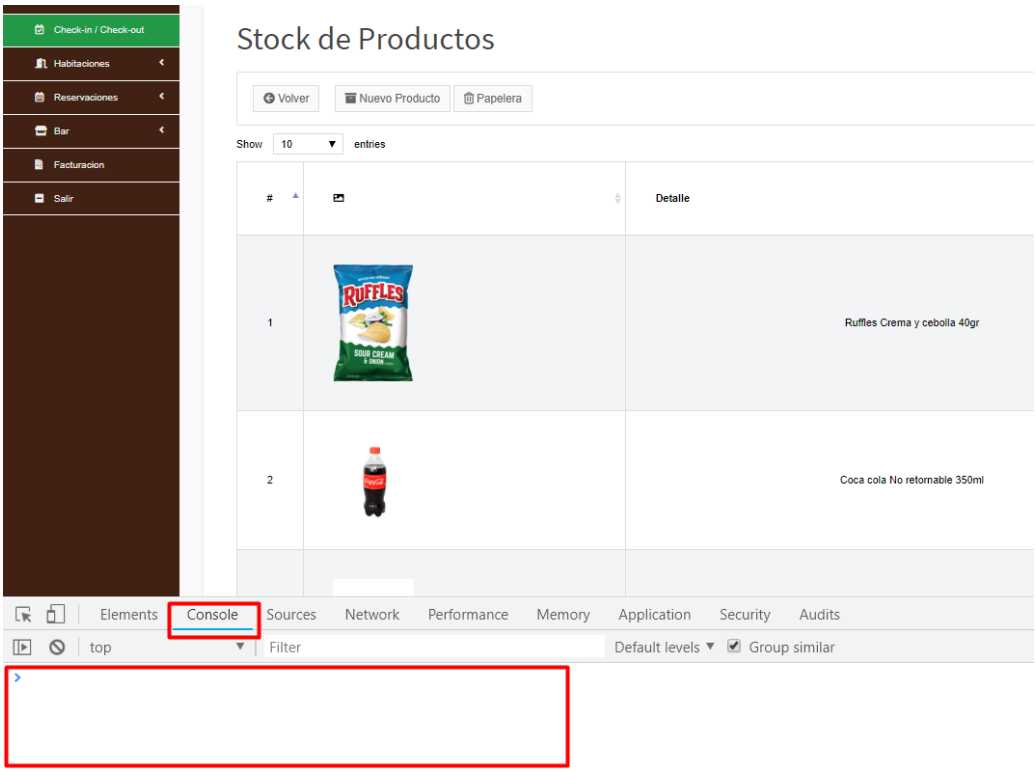


Ilustración 77 Pruebas Stock Caja Blanca Sin errores (Esparza, 2018)

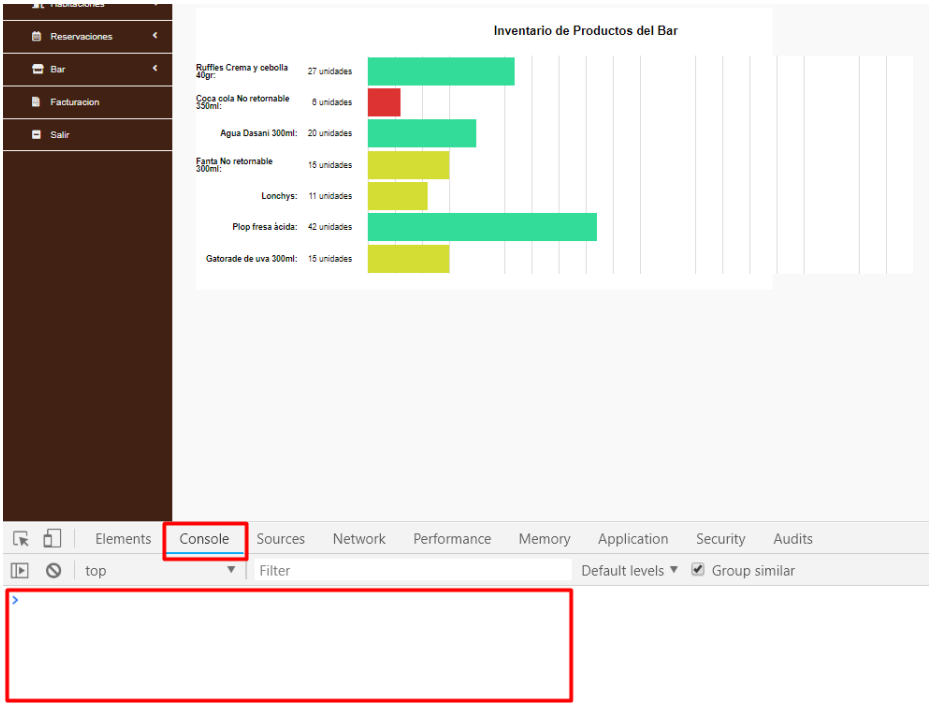


Ilustración 78 Pruebas Inventario Caja Blanca Sin errores (Esparza, 2018)

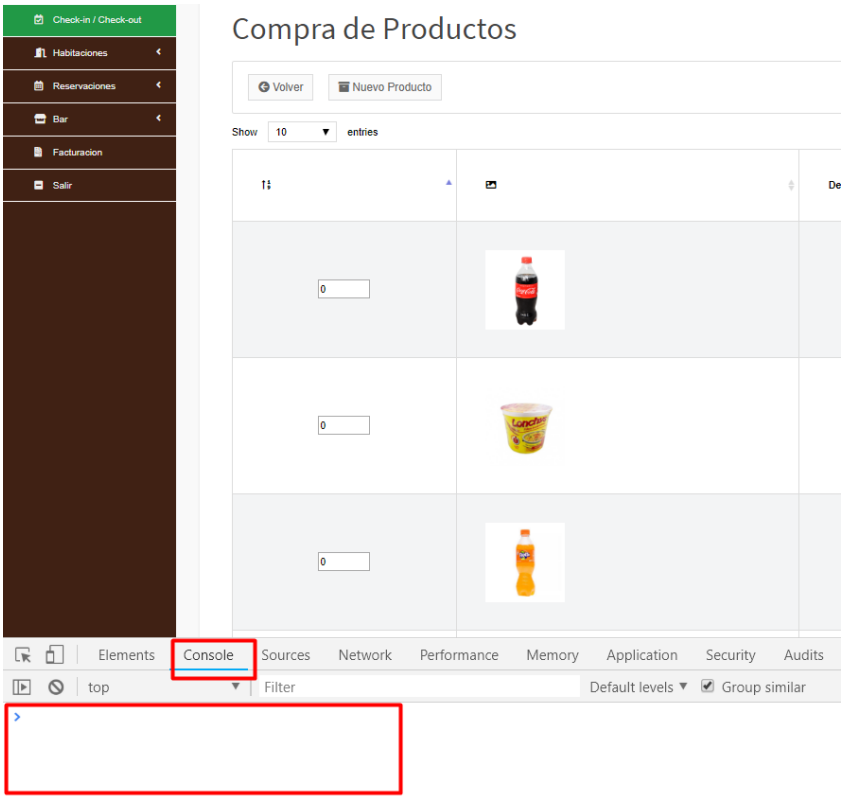


Ilustración 79 Pruebas Compra de Productos Caja Blanca Sin errores (Esparza, 2018)

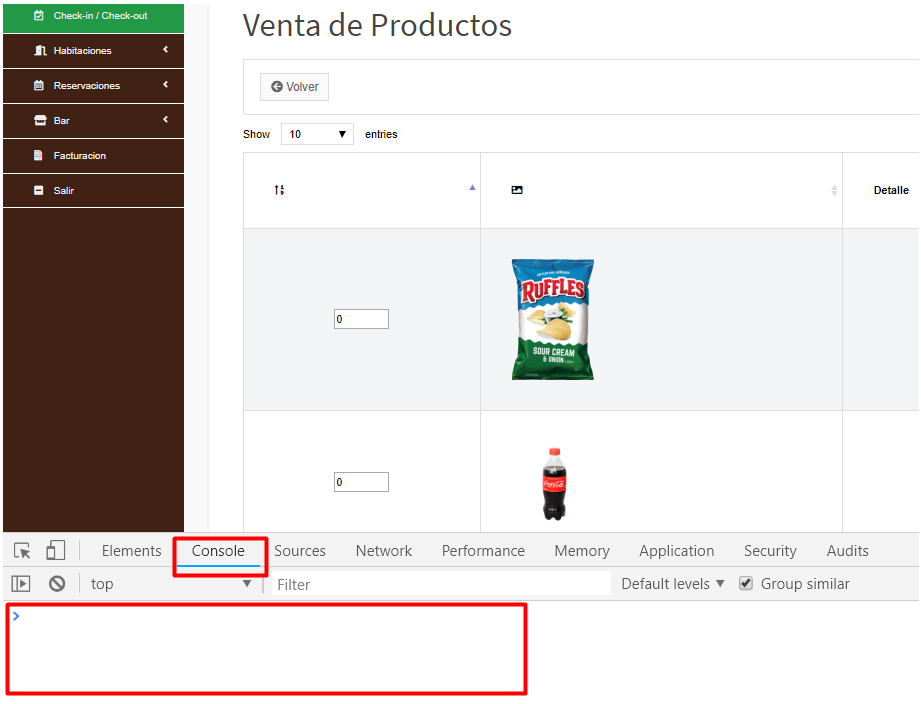


Ilustración 80 Pruebas Venta de productos Caja Blanca Sin errores (Esparza, 2018)

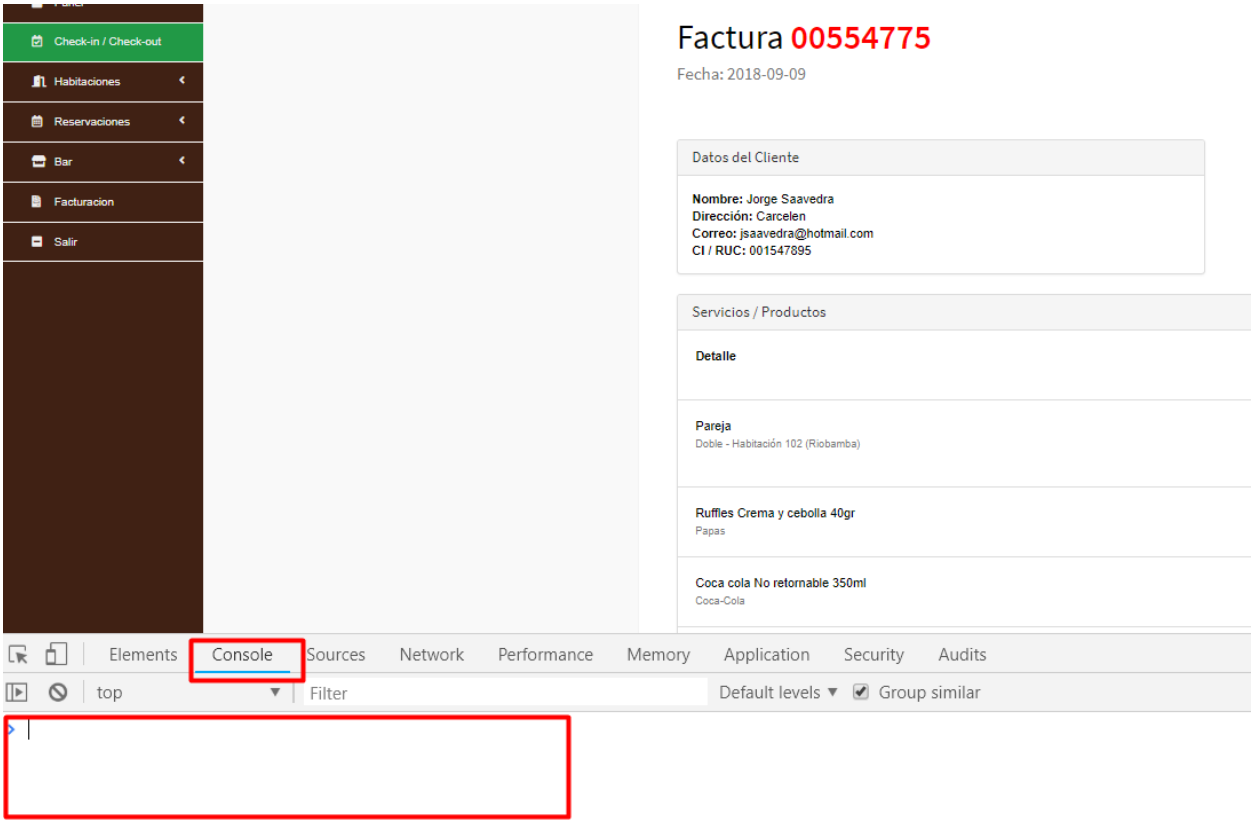


Ilustración 81 Pruebas Factura Caja Blanca Sin errores (Esparza, 2018)

Caja Negra: Criterios de Aceptación

- Ingreso de nuevo producto exitoso
- Compra de productos exitosa
- Venta de productos exitosa
- Inventario con datos reales en gráfico estadístico exitoso
- Páginas de fácil manejo para el usuario
- Factura recopila todos los consumos de cada cliente exitosamente

PRUEBA DE ACEPTACIÓN			
Número de prueba:	3	Número de historia de usuario:	6 7
Nombre de Caso de Prueba:	Usuario del Sistema		
Descripción			
Se evaluará la funcionalidad y la experiencia del usuario al manejar los modulos 6 y 7 del sistema.			
Condiciones de ejecución			
Crear por lo menos 2 productos uno con stock y otro sin stock a la base de datos. Conectarse a internet. Tener por lo menos un Cliente hospedado en el hostal.			
Entradas			
Formulario de Producto Nuevo: Datos del Producto. Formulario para venta de productos: Cantidad de cada producto a vender, Cliente y Habitacion que compra. Formulario para Facturar: Seleccion del cliente que se va a facturar			
Resultado esperado:	Compra y Venta de Productos, Presentar y Cargar la información, Cargar las ventas a un cliente del hostal. Factura con todos los consumos		
Evaluación	Eliminación y Edición de un producto. Realizar correctamente los calculos de la factura.		
	Registrar en la base de datos el usuario que hizo algun movimiento		
	Agregar mensaje del significado de la iconografía usada		

Ilustración 82 Pruebas Caja Negra Tercera Prueba (Esparza, 2018)

Conclusiones y Recomendaciones

Todas las conclusiones y recomendaciones que se realizarán serán enfocadas a la metodología ágil XP, framework Laravel y a nivel general la funcionalidad a nivel macro que el sistema tiene y todas sus posibles aplicaciones en el mercado.

Conclusiones

- Gracias a que el sistema está desarrollado para web, todos los usuarios de las diferentes áreas del negocio pueden conectarse al mismo tiempo, sin interrumpir la actividad de otro.

- El sistema se adapta a cualquier dispositivo Tablet o smartphone permitiendo al usuario su administración en todo momento mientras brinda el mejor servicio al cliente final.

- El sistema permite conocer las habitaciones del negocio que están ocupadas, en mantenimiento y disponibles sin la necesidad de verificarlo presencialmente.

- El sistema permite ahorrar hasta un 50% del tiempo que toma anotar las actividades y consumos en un registro a papel y lápiz.

- Gracias a los colores manejados en el gráfico estadístico para el inventario, el administrador del negocio puede mantener en stock sus productos constantemente.

- La metodología XP ayudó al desarrollo ágil de este sistema gracias a la documentación reducida frente a otras metodologías.

- El sistema maneja tokens de autenticación para cada usuario logeado, por lo que aumenta considerablemente su nivel de seguridad.

- El sistema protege todos los datos ya que estos tienen un borrado lógico, es decir, ningún registro en ningún momento es eliminado definitivamente, solo es cambiado de estado.

- El sistema facilita el Check-in o reserva de un cliente que ya haya estado en el negocio, gracias al módulo de cliente frecuente.

- Gracias a los criterios de aceptación en cada ciclo de desarrollo de pudo darle mantenimiento al ciclo anterior dándole valores agregados y/o corrigiendo errores detectados por el usuario.

Recomendaciones

- Se sugiere definir un proceso de corrección de errores para evitar que la corrección de un error provoque otros fallos.

- El sistema debe ser manejado y actualizado a diario para evitar que los datos de los clientes y las reservas estén desactualizadas.

- Se recomienda que, en la fase de pruebas, el usuario que use el sistema, sea una persona que nunca haya usado uno similar, con el fin de recibir la mayor cantidad de observaciones y brindar el mejor mantenimiento al sistema.

- Se recomienda anotar las consideraciones que la persona encargada de realizar los prototipos detecta mientras diseña los mismos.

- Se recomienda al negocio que use el sistema disponga de una página web para integrar los módulos de Check-in y Reservaciones del sistema con los de la página web.

- Se recomienda a todo negocio que use este sistema, tener un buen proveedor de internet ya que el sistema estaría alojado en la web.

Bibliografía

Beck, K. (2000). *Extreme Programming Explained: embrace change.* Boston, MA, USA: Addison-Wesley Longman Publishing Co.

BitRock Inc. (27 de 05 de 2018). *Apache Friends*. Obtenido de Apache: https://www.apachefriends.org/es/index.html

Capuñay, O. (2013). *Desarrollo Web con PHP: Aprende PHP.* Perú: ocapunay.

Gutierrez, E. (2009). *JavaScript Conceptos básicos y avanzados (bibliotecas Prototype y Script.aculo.us).* Barcelona, España: ENI.

Joskowicz, I. J. (10 de Febrero de 2008). *Universidad de la República.* Obtenido de Instituto de Ingeniería Eléctrica: https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf

Laínez Fuentes, J. R. (2016). *Desarrollo de Software ÁGIL: Extreme Programming y Scrum.* España: IT Campus Academy.

Perez, D. (2008). METODOLOGÍAS ÁGILES.¿CÓMO DESARROLLO UTILIZANDO XP? *Convencion Cientifica de Ingenieria y Arquitectura*, 5.

Robbins, J. N. (2012). *Learning Web Design A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics.* Sebastopol, California, USA: O´Reilly Media Inc.

Sierra, K. (31 de Enero de 2018). *¿Qué es Laravel? Ventajas del desarrollo a medida para tus proyectos*. Obtenido de Synergyweb: https://www.synergyweb.es/blog/laravel-desarrollo-medida.html

Van Lancker, L. (2012). *HTML5 Los fundamentos del lenguaje.* Barcelona, España: ENI.

Wake, W. C. (2002). *Extreme Programming Explored.* Boston, MA, USA: Addison-Wesley.

Weissinger, S. S. (2001). *Introducción a las actividades hoteleras.* Madrid, España: Ma Teresa Gómez-Mascaraque Perez.

Pérez, D. (2008). *Investigación de la metodología ágil Extreme Programming y su aplicación a un caso de estudio.* (Tesis de Pregrado). Instituto Superior Politécnico José Antonio Echeverría, La Habana, Cuba.