UNIVERSIDAD PRIVADA DEL NORTE

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

SISTEMA DE ADMINISTRACIÓN HOTELERA

Informe académico

Autor(es):
Diego Tasaico
Neftali Zapata
Olenka Lazo
Mauricio Villanueva

Curso:
Modelamiento y Análisis de Software

Docente:
Jorge Alfredo Guevara Jimenez

LIMA – PERÚ
2020-2

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN 1
1. Identificación del problema 1
2. Planteamiento de la solución 1
3. Cómo implementar la solución 1
4. Justificación y limitaciones de la investigación 1
5. Ventajas de la solución 2
6. Desventajas de la solución 2
7. Objetivo general 2
8. Objetivos específicos 2

CAPÍTULO 2. MARCO TEÓRICO 2
9. Marco teórico 2
10. Marco metodológico 3

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN 4
11. Implementación de la solución planteada 4

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES 127
12. Conclusiones 127
13. Recomendaciones 128
14. Referencias Bibliográficas 129
15. Anexos 129

ÍNDICE DE TABLAS

Tabla 1: Proceso de Negocio 5

Tabla 2: Trazabilidad de Requisitos 12

Tabla 3: Especificación de Casos de uso - Versión 1 - 1.....47

Tabla 4: Especificación de Casos de uso - Versión 1 - 2.....49

Tabla 5: Especificación de Casos de uso - Versión 1 - 3.....51

Tabla 6: Especificación de Casos de uso - Versión 1 - 4.....53

Tabla 7: Especificación de Casos de uso - Versión 2 - 1.....55

Tabla 8: Especificación de Casos de uso - Versión 2 - 2.....57

Tabla 9: Plan de pruebas Incremental – Versión 1.....60

Tabla 10: Plan de pruebas Incremental – Versión 2.....61

Tabla 11: Modelo FURPS - 1.....119

Tabla 12: Modelo FURPS - 2.....126

ÍNDICE DE FIGURAS

Ilustración 1: Proceso de Negocio....4

Ilustración 2: Proceso de Software....5

Ilustración 3: Diagrama de Casos de uso....6

Ilustración 4: Prototipo - Versión 1 - 1....6

Ilustración 5: Prototipo - Versión 1 - 2....7

Ilustración 6: Prototipo - Versión 1 - 3....7

Ilustración 7: Prototipo - Versión 1 - 4....8

Ilustración 8: Prototipo - Versión 2 - 1....8

Ilustración 9: Prototipo - Versión 2 - 2....9

Ilustración 10: Diagrama de actividades con particiones o calles....10

Ilustración 11: Requisitos no funcionales - Diego Tasaico - 1....13

Ilustración 12: Requisitos no funcionales - Diego Tasaico - 2....13

Ilustración 13: Requisitos no funcionales - Diego Tasaico - 3....14

Ilustración 14: Requisitos no funcionales - Diego Tasaico - 4....14

Ilustración 15: Requisitos no funcionales - Mauricio Villanueva - 1....15

Ilustración 16: Requisitos no funcionales - Mauricio Villanueva - 2....15

Ilustración 17: Requisitos no funcionales - Mauricio Villanueva - 3....16

Ilustración 18: Requisitos no funcionales - Abihail Zapata - 1....16

Ilustración 19: Requisitos no funcionales - Abihail Zapata - 2....17

Ilustración 20: Requisitos no funcionales - Abihail Zapata - 3....17

Ilustración 21: Requisitos no funcionales - Abihail Zapata - 4....18

Ilustración 22: Requisitos no funcionales - Olenka Lazo - 1....18

Ilustración 23: Requisitos no funcionales - Olenka Lazo - 2....19

Ilustración 24: Requisitos no funcionales - Olenka Lazo - 3....19

Ilustración 25: Requisitos no funcionales - Olenka Lazo - 4....20

Ilustración 26: Repositorio GitHub - 1....20

Ilustración 27: Repositorio GitHub - 2 21

Ilustración 28: Repositorio GitHub - 3 21

Ilustración 29: Repositorio GitHub - 4 22

Ilustración 30: Repositorio GitHub - 5 22

Ilustración 31: Repositorio GitHub - 6 23

Ilustración 32: Repositorio GitHub - 7 23

Ilustración 33: Diagrama de Casos de uso Relacionado - General 35

Ilustración 34: Diagrama de Casos de uso Relacionado - Versión 1 36

Ilustración 35: Código de programación del software - 1 37

Ilustración 36: Código de programación del software - 2 37

Ilustración 37: Código de programación del software - 3 38

Ilustración 38: Código de programación del software - 4 38

Ilustración 39: Código de programación del software - 5 39

Ilustración 40: Código de programación del software - 6 39

Ilustración 41: Código de programación del software - 7 40

Ilustración 42: Código de programación del software - 8 40

Ilustración 43: Código de programación del software - 9 41

Ilustración 44: Código de programación del software - 10 41

Ilustración 45: Código de programación del software - 11 42

Ilustración 46: Código de programación del software - 12 42

Ilustración 47: Código de programación del software - 13 43

Ilustración 48: Código de programación del software - 14 43

Ilustración 49: Código de programación del software - 15 44

Ilustración 50: Código de programación del software - 16 44

Ilustración 51: Código de programación del software - 17 45

Ilustración 52: Código de programación del software - 18 45

Ilustración 53: Registro de Cliente - Datos de entrada normal - Versión 1 62

Ilustración 54: Registro de Cliente - Resultado esperado normal - Versión 1 62

Ilustración 55: Registro de Cliente - Datos de entrada anómalo - Versión 1 63

Ilustración 56: Registro de Cliente - Resultado esperado anómalo - Versión 1 63

Ilustración 57: Eliminar Cliente - Datos de entrada normal - Versión 1.....64

Ilustración 58: Eliminar Cliente - Resultado esperado normal - Versión 1.....64

Ilustración 59: Eliminar Cliente - Datos de entrada anómalo - Versión 1.....65

Ilustración 60: Eliminar Cliente - Resultado esperado anómalo - Versión 1.....65

Ilustración 61: Modificar Cliente - Datos de entrada normal - Versión 1.....66

Ilustración 62: Modificar Cliente - Resultado esperado normal - Versión 1.....66

Ilustración 63: Modificar Cliente - Datos de entrada anómalo - Versión 1 67

Ilustración 64: Modificar Cliente - Resultado esperado anómalo - Versión 1 67

Ilustración 65: Consultar Cliente - Datos de entrada normal - Versión 1 68

Ilustración 66: Consultar Cliente - Resultado esperado normal - Versión 1 68

Ilustración 67: Consultar Cliente - Datos de entrada anómalo - Versión 1 69

Ilustración 68: Consultar Cliente - Resultado esperado anómalo - Versión 1 69

Ilustración 69: Registrar Trabajador - Datos de entrada normal - Versión 1.....70

Ilustración 70: Registrar Trabajador - Resultado esperado normal - Versión 1 70

Ilustración 71: Registrar Trabajador - Datos de entrada anómalo - Versión 1.....71

Ilustración 72: Registrar Trabajador - Resultado esperado anómalo - Versión 1.....71

Ilustración 73: Eliminar Trabajador - Datos de entrada normal - Versión 1 72

Ilustración 74: Eliminar Trabajador - Resultado esperado normal - Versión 1 72

Ilustración 75: Eliminar Trabajador - Datos de entrada anómalo - Versión 1 73

Ilustración 76: Eliminar Trabajador - Resultado esperado anómalo - Versión 1 73

Ilustración 77: Modificar Trabajador - Datos de entrada normal - Versión 1 74

Ilustración 78: Modificar Trabajador - Resultado esperado normal - Versión 1 74

Ilustración 79: Modificar Trabajador - Datos de entrada anómalo - Versión 1.....75

Ilustración 80: Modificar Trabajador - Resultado esperado anómalo - Versión 1 75

Ilustración 81: Consultar Trabajador - Datos de entrada normal - Versión 1.....76

Ilustración 82: Consultar Trabajador - Resultado esperado normal - Versión 1 76

Ilustración 83: Consultar Trabajador - Datos de entrada anómalo - Versión 1.....77

Ilustración 84: Consultar Trabajador - Resultado esperado anómalo - Versión 1.....77

Ilustración 85: Validar Trabajador - Datos de entrada normal - Versión 1 78

Ilustración 86: Validar Trabajador - Resultado esperado normal - Versión 1 78

Ilustración 87: Validar Trabajador - Datos de entrada anómalo - Versión 1.....79

Ilustración 88: Validar Trabajador - Resultado esperado anómalo - Versión 1 79

Ilustración 89: Registro de entrada de cliente - Datos de entrada normal - Versión 1.....80

Ilustración 90: Registro de entrada de cliente - Resultado esperado normal - Versión 1 80

Ilustración 91: Registro de entrada de cliente - Datos de entrada anómalo - Versión 1.....81

Ilustración 92: Registro de entrada de cliente - Resultado esperado anómalo - Versión 1....81

Ilustración 93: Eliminar la entrada de cliente - Datos de entrada normal - Versión 1 82

Ilustración 94: Eliminar la entrada de cliente - Resultado esperado normal - Versión 1 82

Ilustración 95: Eliminar la entrada de cliente - Datos de entrada anómalo - Versión 1 83

Ilustración 96: Eliminar la entrada de cliente - Resultado esperado anómalo - Versión 1 83

Ilustración 97: Modificar la entrada de cliente - Datos de entrada normal - Versión 1 84

Ilustración 98: Modificar la entrada de cliente - Resultado esperado normal - Versión 1 84

Ilustración 99: Modificar la entrada de cliente - Datos de entrada anómalo - Versión 1.....85

Ilustración 100: Modificar la entrada de cliente - Resultado esperado anómalo - Versión 1..85

Ilustración 101: Consultar la entrada de cliente - Datos de entrada normal - Versión 1.....86

Ilustración 102: Consultar la entrada de cliente - Resultado esperado normal - Versión 1....86

Ilustración 103: Consultar la entrada de cliente - Datos de entrada anómalo - Versión 1.....87

Ilustración 104: Consultar la entrada de cliente - Resultado esperado anómalo - Versión 1.87

Ilustración 105: Registro de salida de cliente - Datos de entrada normal - Versión 1 88

Ilustración 106: Registro de salida de cliente - Resultado esperado normal - Versión 1 88

Ilustración 107: Registro de salida de cliente - Datos de entrada anómalo - Versión 1.....89

Ilustración 108: Registro de salida de cliente - Resultado esperado anómalo - Versión 1.....89

Ilustración 109: Eliminar la salida de cliente - Datos de entrada normal - Versión 1.....90

Ilustración 110: Eliminar de salida de cliente - Resultado esperado normal - Versión 1.....90

Ilustración 111: Eliminar de salida de cliente - Datos de entrada anómalo - Versión 1 91

Ilustración 112: Eliminar de salida de cliente - Resultado esperado anómalo - Versión 1 91

Ilustración 113: Modificar de salida de cliente - Datos de entrada normal - Versión 1 92

Ilustración 114: Modificar de salida de cliente - Resultado esperado normal - Versión 1 92

Ilustración 115: Modificar de salida de cliente - Datos de entrada anómalo - Versión 1 93

Ilustración 116: Modificar de salida de cliente - Resultado esperado anómalo - Versión 1 ...93

Ilustración 117: Consultar la salida de cliente - Datos de entrada normal - Versión 1 94

Ilustración 118: Consultar la salida de cliente - Resultado esperado normal - Versión 1 94

Ilustración 119: Consultar la salida de cliente - Datos de entrada anómalo - Versión 1.....95

Ilustración 120: Consultar la salida de cliente - Resultado esperado anómalo - Versión 1....95

Ilustración 121: Registrar Habitaciones - Datos de entrada normal - Versión 2.....96

Ilustración 122: Registrar Habitaciones - Resultado esperado normal - Versión 2.....96

Ilustración 123: Registrar Habitaciones - Datos de entrada anómalo - Versión 2 97

Ilustración 124: Registrar Habitaciones - Resultado esperado anómalo - Versión 2.....97

Ilustración 125: Eliminar Habitaciones - Datos de entrada normal - Versión 2 98

Ilustración 126: Eliminar Habitaciones - Resultado esperado normal - Versión 2 98

Ilustración 127: Eliminar Habitaciones - Datos de entrada anómalo - Versión 2.....99

Ilustración 128: Eliminar Habitaciones - Resultado esperado anómalo - Versión 2 99

Ilustración 129: Modificar Habitaciones - Datos de entrada normal - Versión 2.....100

Ilustración 130: Modificar Habitaciones - Resultado esperado normal - Versión 2 100

Ilustración 131: Modificar Habitaciones - Datos de entrada anómalo - Versión 2.....101

Ilustración 132: Modificar Habitaciones - Resultado esperado anómalo - Versión 2.....101

Ilustración 133: Consultar Habitaciones - Datos de entrada normal - Versión 2.....102

Ilustración 134: Consultar Habitaciones - Resultado esperado normal - Versión 2.....102

Ilustración 135: Consultar Habitaciones - Datos de entrada anómalo - Versión 2 103

Ilustración 136: Consultar Habitaciones - Resultado esperado anómalo - Versión 2.....103

Ilustración 137: Registrar Reserva - Datos de entrada normal - Versión 2 104

Ilustración 138: Registrar Reserva - Resultado esperado normal - Versión 2 104

Ilustración 139: Registrar Reserva - Datos de entrada anómalo - Versión 2 105

Ilustración 140: Registrar Reserva - Resultado esperado anómalo - Versión 2 105

Ilustración 141: Eliminar Reserva - Datos de entrada normal - Versión 2.....106

Ilustración 142: Eliminar Reserva - Resultado esperado normal - Versión 2.....106

Ilustración 143: Eliminar Reserva - Datos de entrada anómalo - Versión 2.....107

Ilustración 144: Eliminar Reserva - Resultado esperado anómalo - Versión 2.....107

Ilustración 145: Modificar Reserva - Datos de entrada normal - Versión 2.....108

Ilustración 146: Modificar Reserva - Resultado esperado normal - Versión 2.....108

Ilustración 147: Modificar Reserva - Datos de entrada anómalo - Versión 2 109

Ilustración 148: Modificar Reserva - Resultado esperado anómalo - Versión 2 109

Ilustración 149: Consultar Reserva - Datos de entrada normal - Versión 2 110

Ilustración 150: Consultar Reserva - Resultado esperado normal - Versión 2 110

Ilustración 151: Consultar Reserva - Datos de entrada anómalo - Versión 2 111

Ilustración 152: Consultar Reserva - Resultado esperado anómalo - Versión 2 111

Ilustración 153: Videoconferencia con representante de la empresa 129

Ilustración 154: Integrante - Diego Tasaico 130

Ilustración 155: Integrante - Mauricio Villanueva 130

Ilustración 156: Integrante - Olenka Lazo.....131

Ilustración 157: Integrante - Abihail Zapata 131

Ilustración 158: Acta de compromiso.....132

Ilustración 159: Funcionamiento del Software - Acceso al sistema 133

Ilustración 160: Funcionamiento del Software - Menú Principal.....133

Ilustración 161: Funcionamiento del Software - Vista de los clientes.....134

Ilustración 162: Funcionamiento del Software - Vista de los trabajadores.....134

Ilustración 163: Funcionamiento del Software - Vista de los movimientos 134

Ilustración 164: Funcionamiento del Software - Vista de las reservas.....134

Ilustración 165: Funcionamiento del Software - Vista de las habitaciones.....134

CAPÍTULO 1. INTRODUCCIÓN

1. Identificación del problema

Actualmente el problema que presenta el hotel surge en el proceso de alquiler de habitaciones, ya que no cuenta con un sistema de administración y todo se realiza de forma manual, como el cobro por el servicio, el registro de habitaciones, la verificación de habitaciones disponibles y verificación de la cantidad de clientes que se hospedan durante un intervalo de tiempo. Este problema, si bien cumple con su objetivo que es la atención al cliente, demanda mucho tiempo, esfuerzo y malinterpretación de la información con respecto a los datos del cliente y su reserva.

2. Planteamiento de la solución

Para poder solucionar este problema lo que el hotel necesita es un sistema a través del cual se pueda buscar hospedaje, gestar la reserva de éste y realizar su administración, para ello se empezará con un modelo o prototipo que debe tener estas funcionalidades básicas.

3. Cómo implementar la solución

Se va a crear una solución de software que será usada por el usuario de una manera local para mejorar la productividad de la empresa y sus trabajadores. Para esto tomamos en cuenta las necesidades del cliente, el proceso que se lleva durante el registro y el producto final esperado del sistema desarrollado.

4. Justificación y limitaciones de la investigación

Limitaciones:

- Tiempo: Tendremos todo este ciclo académico para desarrollar sistema.
- Espacio: Todo el proyecto será realizado mediante sistemas digitales.
- Recursos: Tendremos que tomar en cuenta las limitaciones para la adquisición de los aparatos electrónicos.

Justificación:

- Se justifica en base a los avances tecnológicos y en su implementación en las grandes, medianas y pequeñas empresas.
- Se justifica en base a la eficacia y productividad de los trabajadores.
- Se justifica en base a la experiencia y servicio brindado a los clientes.

5. Ventajas de la solución

Como ventaja principal tenemos la eficacia que se generará en el registro de nuevos clientes, se podrá visualizar desde un primer momento un cambio significativo en el tiempo requerido para ello.

6. Desventajas de la solución

Como desventaja principal tenemos la adquisición del material necesario para la implementación del sistema, otra de las desventajas sería el gasto en capacitaciones para el personal administrativo del lugar.

7. Objetivo general

Desarrollar un sistema de administración hotelera, que pueda cumplir las necesidades de la empresa, contando con la posibilidad de hacer las reservas de las habitaciones y realizar el cobro respectivo al cliente.

8. Objetivos específicos

- Implementar un mantenimiento de clientes, donde se podrá registrar, consultar, modificar y eliminar a los clientes, incluyendo la entrada y salida del cliente.
- Implementar el mantenimiento de habitaciones y reservas.
- Permitir al administrador o recepcionista poder consultar sobre las habitaciones que fueron reservadas.

CAPÍTULO 2. MARCO TEÓRICO

9. Marco teórico

- Modelo de negocio: Alonso, Martínez y Segovia (2005) señalan que el modelo de negocio es necesario para poder entender los procesos de negocio de la empresa y así poder tener una descripción detallada de los requisitos del sistema.
- Proceso de software: “Un proceso de software es una secuencia de actividades que conducen a la elaboración de un producto de software.” (Sommerville, 2011, p. 9).
- Prototipos: “permiten a los usuarios ver que tan bien el sistema apoya a su trabajo.” (Sommerville, 2011, p. 45).
- Casos de uso: “un caso de uso identifica a los actores implicados en una interacción” (Sommerville, 2011, p. 107).
- Diagrama de casos de uso: “representa todas las interacciones que se describirán en los requerimientos del sistema.” (Sommerville, 2011, p. 107).

- Requerimientos: Alonso, Martínez y Segovia (2005) señalan que los requerimientos se reconocen a partir de las especificaciones del problema que va a mencionar el usuario.
- Diagramas de actividad: “Los diagramas de actividad intentan mostrar las actividades que incluyen un proceso de sistema, así como el flujo de control de una actividad a otra.” (Sommerville, 2011, p. 123).
- Proceso unificado: Alonso, Martínez y Segovia (2005) señalan que el proceso unificado modela el sistema como un conjunto de bloques interconectados y se implementan mediante componentes.
- GitHub: Es una plataforma de desarrollo inspirada en tu forma de trabajar. Desde el código abierto hasta el negocio, puede alojar y revisar código, administrar proyectos y crear software junto con 50 millones de desarrolladores.
- Lucid chart: Ofrece una interfaz intuitiva y cientos de plantillas que permiten que tus equipos comuniquen sistemas, procesos e ideas complejos con gráficos. Eleva tu perspectiva en todas las áreas de tu negocio para impulsar ventas, simplificar la gestión del personal, mapear tu infraestructura y más.
- NetBeans: Es una herramienta para desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web con Java, JavaScript, HTML5, PHP, C/C++ y más.
- WampServer: Es un ambiente de desarrollo web para Windows. Te permite crear aplicaciones con Apache2, PHP y la base de datos MySQL.
- MySQL WorkBench: Es una herramienta visual unificada para arquitectos de base de datos, desarrolladores y DBAs. Provee herramientas de modelamiento de datos, desarrollo SQL, herramientas de administración de servidores, administración de usuarios, respaldos y mucho más.
- MySQL: Es una base de datos de código abierto. Esta 100% desarrollada, gestionada y sostenida por el equipo de MySQL.
- (NetBeans, s.f.) (MySQL, s.f.) (WampServer, s.f.) (MySQL WorkBench, s.f.)

10. Marco metodológico

- Paso 1: Formar el equipo de trabajo.
- Paso 2: Identificar una empresa.
- Paso 3: Entrevistar al administrador del negocio.
- Paso 4: Modelar y especificar el proceso de negocio.
- Paso 5: Hacer el modelo de proceso de software.
- Paso 6: Hacer el modelo de casos de uso.
- Paso 7: Creación de prototipos.
- Paso 8: Hacer el diagrama de actividades con particiones.
- Paso 9: Avance de informe
- Paso 10: Realizar la captura de los requisitos

- Paso 11: Validar los requisitos
- Paso 12: Realizar la trazabilidad de los requisitos
- Paso 13: Instalar los requisitos no funcionales
- Paso 14: Realizar el diagrama de casos de uso relacionado
- Paso 15: Especificar los casos de uso

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN

11. Implementación de la solución planteada

Ilustración 1: Proceso de Negocio

Procesos	Entradas	Actividades	Salidas	Valor para el cliente
Sistema de Administración hotelera	Necesidad del cliente para poder hospedarse en un ambiente	• Registro de cliente. • Consulta de precio de habitación. • Selección de habitación para el cliente. • Verificar que la habitación se encuentre vacía y limpia. • Realización del pago del cliente al hotel. • Registrar salida del cliente del hotel.	Registro de salida del cliente	Brindar un servicio de calidad al cliente cumpliendo todas las medidas de bioseguridad

Tabla 1: Proceso de Negocio

Ilustración 2: Proceso de Software

Ilustración 3: Diagrama de Casos de uso

Ilustración 4: Prototipo - Versión 1 - 1

Ilustración 5: Prototipo - Versión 1 - 2

Ilustración 6: Prototipo - Versión 1 - 3

Ilustración 7: Prototipo - Versión 1 - 4

Ilustración 8: Prototipo - Versión 2 - 1

Ilustración 9: Prototipo - Versión 2 - 2

Ilustración 10: Diagrama de actividades con particiones o calles

Tipo de Requisito	ID	Requisito	Casos de Uso	Prototipo	Versión	Instalado
Funcional	1	Permite el Registro de trabajadores	Registro de Trabajador	Si	1	/
Funcional	2	Permite hacer un listado de los trabajadores existentes	Consulta de Trabajadores	Si	1	/
Funcional	3	Permite Modificar a los trabajadores registrados	Modificar al Trabajador	Si	1	/
Funcional	4	Permite Eliminar trabajadores registrados	Eliminar Trabajador	Si	1	/
Funcional	5	Verificar el usuario y acceder al sistema	Validar trabajador	Si	1	/
Funcional	6	Permite el Registro de clientes	Registro de Clientes	Si	1	/
Funcional	7	Permite hacer un listado de los clientes existentes	Consulta de Clientes	Si	1	/
Funcional	8	Permite Modificar a los clientes registrados	Modificar al Cliente	Si	1	/
Funcional	9	Permite Eliminar clientes registrados	Eliminar Cliente	Si	1	/
Funcional	10	Permite registrar la fecha de entrada del cliente.	Registro de entrada del cliente	Si	1	/
Funcional	11	Permite registrar la fecha de salida del cliente	Registro de salida del cliente	Si	1	/
Funcional	12	Permite mostrar la fecha de entrada del cliente	Consulta de entrada de los clientes	Si	1	/
Funcional	13	Permite mostrar la fecha de salida del cliente	Consulta de la salida de los clientes	Si	1	/
Funcional	14	Permite modificar la fecha de entrada del cliente	Modificar entrada del cliente	Si	1	/

Funcional	15	Permite modificar la fecha de salida del cliente	Modificar salida del cliente	Si	1	/
Funcional	16	Permite eliminar la fecha de entrada del cliente	Eliminar la entrada del cliente	Si	1	/
Funcional	17	Permite eliminar la fecha de salida del cliente	Eliminar la salida del cliente	Si	1	/
Funcional	18	Permite registrar habitaciones	Registro de habitaciones	Si	2	/
Funcional	19	Saber el estado de las habitaciones	Consulta de habitaciones	Si	2	/
Funcional	20	Permite modificar habitaciones	Modificar habitaciones	Si	2	/
Funcional	21	Permite eliminar habitaciones	Eliminar habitaciones	Si	2	/
Funcional	22	Permite registrar reservas	Registro de reservas	Si	2	/
Funcional	23	Saber el estado de las reservas	Consulta de reservas	Si	2	/
Funcional	24	Permite modificar reservas	Modificar reservas	Si	2	/
Funcional	25	Permite eliminar reservas	Eliminar reservas	Si	2	/
Funcional	26	Saber los clientes habituales	Consulta de clientes habituales	No	3	/
No funcional	27	NetBeans	/	No	/	Si
No funcional	28	MySQL o WampServer	/	No	/	Si
No funcional	29	MySQL WorkBench	/	No	/	Si

Tabla 2: Trazabilidad de Requisitos

Ilustración 11: Requisitos no funcionales - Diego Tasaico - 1

Ilustración 12: Requisitos no funcionales - Diego Tasaico - 2

Ilustración 13: Requisitos no funcionales - Diego Tasaico - 3

Ilustración 14: Requisitos no funcionales - Diego Tasaico - 4

Ilustración 15: Requisitos no funcionales - Mauricio Villanueva - 1

Ilustración 16: Requisitos no funcionales - Mauricio Villanueva - 2

Ilustración 17: Requisitos no funcionales - Mauricio Villanueva - 3

Ilustración 18: Requisitos no funcionales - Abihail Zapata - 1

Ilustración 19: Requisitos no funcionales - Abihail Zapata - 2

Ilustración 20: Requisitos no funcionales - Abihail Zapata - 3

Ilustración 21: Requisitos no funcionales - Abihail Zapata - 4

Ilustración 22: Requisitos no funcionales - Olenka Lazo - 1

Ilustración 23: Requisitos no funcionales - Olenka Lazo - 2

Ilustración 24: Requisitos no funcionales - Olenka Lazo - 3

Ilustración 25: Requisitos no funcionales - Olenka Lazo - 4

Ilustración 26: Repositorio GitHub - 1

Ilustración 27: Repositorio GitHub - 2

Ilustración 28: Repositorio GitHub - 3

Ilustración 29: Repositorio GitHub - 4

Ilustración 30: Repositorio GitHub - 5

Ilustración 31: Repositorio GitHub - 6

Ilustración 32: Repositorio GitHub - 7

Script de MySQL

```
-- MySQL Script generated by MySQL Workbench

-- Sun Nov  1 16:11:40 2020

-- Model: New Model    Version: 1.0

-- MySQL Workbench Forward Engineering


SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_I
N_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_S
UBSTITUTION';


-- -----------------------------------------------------

-- Schema proyecto

-- -----------------------------------------------------


-- -----------------------------------------------------

-- Schema proyecto

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `proyecto` DEFAULT CHARACTER SET utf8
;

USE `proyecto` ;


-- -----------------------------------------------------

-- Table `proyecto`.`cliente`
```

```
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`cliente` (
  `idCliente` INT NOT NULL AUTO_INCREMENT,
  `nombres` VARCHAR(50) NOT NULL,
  `apellido_paterno` VARCHAR(45) NOT NULL,
  `apellido_materno` VARCHAR(45) NOT NULL,
  `tipo_documento` VARCHAR(20) NOT NULL,
  `num_documento` VARCHAR(15) NOT NULL,
  `celular` VARCHAR(15) NOT NULL,
  `email` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idCliente`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;



-- -----------------------------------------------------
-- Table `proyecto`.`habitacion`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`habitacion` (
  `idHabitacion` INT NOT NULL AUTO_INCREMENT,
  `numero` VARCHAR(3) NOT NULL,
  `piso` VARCHAR(1) NOT NULL,
  `precio` DECIMAL(10,0) NOT NULL,
  `tipo_habitacion` VARCHAR(20) NOT NULL,
  `estado_habitacion` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`idHabitacion`))
```

```
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;


-- -----------------------------------------------------
-- Table `proyecto`.`trabajador`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`trabajador` (
  `idTrabajador` INT NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(60) NOT NULL,
  `apellido_paterno_trabajador` VARCHAR(45) NOT NULL,
  `apellido_materno_trabajador` VARCHAR(45) NOT NULL,
  `tipo_documento_trabajador` VARCHAR(20) NOT NULL,
  `num_documento_trabajador` VARCHAR(15) NOT NULL,
  `celular_trabajador` VARCHAR(15) NOT NULL,
  `email_trabajador` VARCHAR(45) NOT NULL,
  `sueldo` DECIMAL NOT NULL,
  `acceso` VARCHAR(20) NOT NULL,
  `login` VARCHAR(20) NOT NULL,
  `password` VARCHAR(20) NOT NULL,
  `estado_trabajador` TINYINT(1) NOT NULL,
  PRIMARY KEY (`idTrabajador`),
  UNIQUE INDEX `login_UNIQUE` (`login` ASC) VISIBLE)
ENGINE = InnoDB
AUTO_INCREMENT = 2
DEFAULT CHARACTER SET = utf8;
```

```
-- -----------------------------------------------------
-- Table `proyecto`.`reserva`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`reserva` (
  `idReserva` INT NOT NULL AUTO_INCREMENT,
  `idCliente` INT NOT NULL,
  `idTrabajador` INT NOT NULL,
  `idHabitacion` INT NOT NULL,
  `tipo_reserva` VARCHAR(20) NOT NULL,
  `fecha_reserva` DATE NOT NULL,
  `fecha_ingreso` DATE NOT NULL,
  `fecha_salida` DATE NOT NULL,
  `costo_alojamiento` DECIMAL(10,0) NOT NULL,
  `estado_reserva` TINYINT(1) NOT NULL,
  PRIMARY KEY (`idReserva`),
  INDEX `fk_reserva_habitacion_idx` (`idHabitacion` ASC) VISIBLE,
  INDEX `fk_reserva_cliente_idx` (`idCliente` ASC) VISIBLE,
  INDEX `fk_reserva_trabajador_idx` (`idTrabajador` ASC) VISIBLE,
  CONSTRAINT `fk_reserva_cliente`
    FOREIGN KEY (`idCliente`)
    REFERENCES `proyecto`.`cliente` (`idCliente`),
  CONSTRAINT `fk_reserva_habitacion`
    FOREIGN KEY (`idHabitacion`)
    REFERENCES `proyecto`.`habitacion` (`idHabitacion`),
```

```
  CONSTRAINT `fk_reserva_trabajador`
    FOREIGN KEY (`idTrabajador`)
    REFERENCES `proyecto`.`trabajador` (`idTrabajador`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;



-- -----------------------------------------------------
-- Table `proyecto`.`pago`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`pago` (
  `idPago` INT NOT NULL AUTO_INCREMENT,
  `idReserva` INT NOT NULL,
  `tipo_comprobante` VARCHAR(20) NOT NULL,
  `num_comprobante` VARCHAR(20) NOT NULL,
  `igv` DECIMAL(10,0) NOT NULL,
  `total_pago` DECIMAL(10,0) NOT NULL,
  `fecha_emision` DATE NOT NULL,
  `fecha_pago` DATE NOT NULL,
  PRIMARY KEY (`idPago`),
  INDEX `fk_pago_reserva_idx` (`idReserva` ASC) VISIBLE,
  CONSTRAINT `fk_pago_reserva`
    FOREIGN KEY (`idReserva`)
    REFERENCES `proyecto`.`reserva` (`idReserva`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

```
-- -----------------------------------------------------
-- Table `proyecto`.`movimiento`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `proyecto`.`movimiento` (
  `idMovimiento` INT NOT NULL AUTO_INCREMENT,
  `nombreCliente` VARCHAR(45) NULL,
  `apellido_p_Cliente` VARCHAR(45) NULL,
  `apellido_m_Cliente` VARCHAR(45) NULL,
  `habitacionCliente` VARCHAR(45) NULL,
  `fecha_hora_ingreso` TIMESTAMP NULL,
  `fecha_hora_salida` TIMESTAMP NULL,
  `estado_movimiento` TINYINT(1) NULL,
  PRIMARY KEY (`idMovimiento`))
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

CREATE TRIGGER mov_trig AFTER INSERT ON cliente FOR EACH ROW
INSERT INTO movimiento (idMovimiento, nombreCliente, apellido_p_Cliente,
apellido_m_Cliente, habitacionCliente, fecha_hora_ingreso, fecha_hora_salida,
estado_movimiento)
```

```
VALUES (null, NEW.nombres, NEW.apellido_paterno, NEW.apellido_materno,
null, null,null,1);


INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,
`nombre`,
`apellido_paterno_trabajador`,
`apellido_materno_trabajador`,
`tipo_documento_trabajador`,
`num_documento_trabajador`,
`celular_trabajador`,
`email_trabajador`,
`sueldo`,
`acceso`,
`login`,
`password`,
`estado_trabajador`)
VALUES
(null,
"Diego",
"Tasaico",
"Freundt",
"DNI",
"73194743",
"966107790",
"tasaicod4@gmail.com",
1500,
```

```
"Administrador",
"userdiego",
"1234",
1);

INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,
`nombre`,
`apellido_paterno_trabajador`,
`apellido_materno_trabajador`,
`tipo_documento_trabajador`,
`num_documento_trabajador`,
`celular_trabajador`,
`email_trabajador`,
`sueldo`,
`acceso`,
`login`,
`password`,
`estado_trabajador`)
VALUES
(null,
"Mauricio",
"Villanueva",
"Yaranga",
"DNI",
"73191743",
"966106790",
```

```
"mauvill@gmail.com",
1500,
"Administrador",
"usermau",
"1234",
1);

INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,
`nombre`,
`apellido_paterno_trabajador`,
`apellido_materno_trabajador`,
`tipo_documento_trabajador`,
`num_documento_trabajador`,
`celular_trabajador`,
`email_trabajador`,
`sueldo`,
`acceso`,
`login`,
`password`,
`estado_trabajador`)
VALUES
(null,
"Olenka",
"Lazo",
"Pichi",
"DNI",
```

```
"73194793",
"966109790",
"ole4@gmail.com",
1500,
"Administrador",
"userole",
"1234",
1);

INSERT INTO `proyecto`.`trabajador`(`idTrabajador`,
`nombre`,
`apellido_paterno_trabajador`,
`apellido_materno_trabajador`,
`tipo_documento_trabajador`,
`num_documento_trabajador`,
`celular_trabajador`,
`email_trabajador`,
`sueldo`,
`acceso`,
`login`,
`password`,
`estado_trabajador`)
VALUES
(null,
"Abihail",
"Zapata",
```

"Zamora",

"DNI",

"73194143",

"916107790",

"abih@gmail.com",

1500,

"Administrador",

"userabi",

"1234",

1);

Ilustración 33: Diagrama de Casos de uso Relacionado - General

Ilustración 34: Diagrama de Casos de uso Relacionado - Versión 1

Ilustración 35: Código de programación del software - 1

Ilustración 36: Código de programación del software - 2

```
70          if(e.getSource() == vista.btn_nuevo_cliente){
71              limpiarCajas();;
72              vista.btn_nuevo_cliente.setEnabled(false);
73              vista.btn_registrar_cliente.setEnabled(true);
74              vista.btn_actualizar_cliente.setEnabled(false);
75          }
76          if(e.getSource() == vista.btn_salir_cliente){
77              salir();
78          }
79      }
80      private void registrar(){
81          if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().equa
82              || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().equal
83              JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
84              listar = false;
85          }
86          else{
87              listar = true;
88              String nombres = vista.txt_nombre_cliente.getText();
89              String apellido_paterno = vista.txt_apellido_p_cliente.getText();
90              String apellido_materno = vista.txt_apellido_m_cliente.getText();
91              String tipo_documento = vista.txt_tipo_doc_cliente.getText();
92              String num_documento = vista.txt_num_doc_cliente.getText();
93              String celular = vista.txt_cel_cliente.getText();
94              String email = vista.txt_email_cliente.getText();
95
96              Cliente cliente = new Cliente(nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
97              int listo = this.dao.agregar(cliente);
98              if(listo == 1){
99                  JOptionPane.showMessageDialog(vista, "Cliente registrado con éxito");
100             }else{
101                 JOptionPane.showMessageDialog(vista, "Error al registrar cliente");
102             }
103         }
104     }
```

Ilustración 37: Código de programación del software - 3

```
144     private void limpiarCajasNoID(){
145         vista.txt_nombre_cliente.setText("");
146         vista.txt_apellido_p_cliente.setText("");
147         vista.txt_apellido_m_cliente.setText("");
148         vista.txt_tipo_doc_cliente.setText("");
149         vista.txt_num_doc_cliente.setText("");
150         vista.txt_cel_cliente.setText("");
151         vista.txt_email_cliente.setText("");
152     }
153
154     private void editar(){
155         int fila = vista.tb_datos_cliente.getSelectedRow();
156             if(fila == -1){
157                 JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
158             }else{
159                 this.vista.btn_registrar_cliente.setEnabled(false);
160                 this.vista.btn_actualizar_cliente.setEnabled(true);
161                 this.vista.btn_nuevo_cliente.setEnabled(true);
162                 int codigo = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
163                 String nombres = (String)vista.tb_datos_cliente.getValueAt(fila, 1);
164                 String apellido_paterno = (String)vista.tb_datos_cliente.getValueAt(fila, 2);
165                 String apellido_materno = (String)vista.tb_datos_cliente.getValueAt(fila, 3);
166                 String tipo_Documento = (String)vista.tb_datos_cliente.getValueAt(fila, 4);
167                 String num_documento = (String)vista.tb_datos_cliente.getValueAt(fila, 5);
168                 String celular = (String)vista.tb_datos_cliente.getValueAt(fila, 6);
169                 String email = (String)vista.tb_datos_cliente.getValueAt(fila, 7);
170
171                 vista.txt_idCliente.setText(codigo+"");
172                 vista.txt_nombre_cliente.setText(nombres);
173                 vista.txt_apellido_p_cliente.setText(apellido_paterno);
174                 vista.txt_apellido_m_cliente.setText(apellido_materno);
175                 vista.txt_tipo_doc_cliente.setText(tipo_Documento);
176                 vista.txt_num_doc_cliente.setText(num_documento);
177                 vista.txt_cel_cliente.setText(celular);
178                 vista.txt_email_cliente.setText(email);
179             }
180     }
```

Ilustración 38: Código de programación del software - 4

```
182     private void actualizar(){
183         if (vista.txt_nombre_cliente.getText().equals("") || vista.txt_apellido_p_cliente.getText().equals("") || vista.txt_apellido_m_cliente.getText().
184             || vista.txt_tipo_doc_cliente.getText().equals("") || vista.txt_num_doc_cliente.getText().equals("") || vista.txt_cel_cliente.getText().eq
185             JOptionPane.showMessageDialog(vista, "Debe completar los campos de nombres, apellidos, tipo y numero de documento y celular");
186             listar = false;
187         }else{
188             actualizar = true;
189             vista.btn_registrar_cliente.setEnabled(true);
190             vista.btn_actualizar_cliente.setEnabled(false);
191             vista.btn_nuevo_cliente.setEnabled(false);
192             int idCliente = Integer.parseInt(vista.txt_idCliente.getText());
193             String nombres = vista.txt_nombre_cliente.getText();
194             String apellido_paterno = vista.txt_apellido_p_cliente.getText();
195             String apellido_materno = vista.txt_apellido_m_cliente.getText();
196             String tipo_documento = vista.txt_tipo_doc_cliente.getText();
197             String num_documento = vista.txt_num_doc_cliente.getText();
198             String celular = vista.txt_cel_cliente.getText();
199             String email = vista.txt_email_cliente.getText();
200
201             Cliente cliente = new Cliente(idCliente, nombres, apellido_paterno, apellido_materno, tipo_documento, num_documento, celular, email);
202             int r = this.dao.actualizar(cliente);
203             if(r == 1){
204                 JOptionPane.showMessageDialog(vista, "Cliente actualizado con éxito");
205             }else{
206                 JOptionPane.showMessageDialog(vista, "Error al actualizar cliente");
207             }
208         }
209     }
```

Ilustración 39: Código de programación del software - 5

```
208             }
209         }
210
211         private void eliminar(){
212             int fila = vista.tb_datos_cliente.getSelectedRow();
213             if(fila == -1){
214                 JOptionPane.showMessageDialog(vista, "Debe seleccionar una fila");
215             }else{
216                 int id = Integer.parseInt((String)vista.tb_datos_cliente.getValueAt(fila, 0).toString());
217                 int iden = this.dao.eliminar(id);
218                 if (iden == 1){
219                     JOptionPane.showMessageDialog(vista, "Cliente eliminado con éxito");
220                     vista.btn_registrar_cliente.setEnabled(true);
221                     vista.btn_nuevo_cliente.setEnabled(false);
222                     vista.btn_actualizar_cliente.setEnabled(false);
223                 }else{
224                     JOptionPane.showMessageDialog(vista, "Error al eliminar cliente");
225                 }
226             }
227         }
228         private void salir(){
229             System.exit(0);
230         }
231     }
```

Ilustración 40: Código de programación del software - 6

```java
package controllers;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;
import models.Habitacion;
import models.HabitacionDAO;
import views.VistaHabitaciones;

public class HabitacionController implements ActionListener{
    boolean listar;
    boolean actualizar;

    DefaultTableModel modeloTabla = new DefaultTableModel();
    VistaHabitaciones vista = new VistaHabitaciones();
    HabitacionDAO dao = new HabitacionDAO();

    public HabitacionController(VistaHabitaciones vistaHabitaciones){
        this.vista=vistaHabitaciones;
        this.vista.btnListar_hab.addActionListener(this);
        this.vista.btnnuevo_hab.addActionListener(this);
        this.vista.btnguardar_hab.addActionListener(this);
        this.vista.btncancelar_hab.addActionListener(this);
        this.vista.btnbuscar_hab.addActionListener(this);
        this.vista.btneliminar_hab.addActionListener(this);
        this.vista.btnsalir_hab.addActionListener(this);
        this.vista.btneditar_hab.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {

    }
}
```

Ilustración 41: Código de programación del software - 7

```java
    private void listar(JTable tabla){
        this.vista.btn_editar_cliente.setEnabled(true);
        this.vista.btn_eliminar_cliente.setEnabled(true);
        this.modeloTabla = (DefaultTableModel)tabla.getModel();
        List<Cliente> listarClientes = this.dao.listar();
        Object[] objeto = new Object[9];
        for(int i=0;i<listarClientes.size();i++){
            objeto[0] = listarClientes.get(i).getIdCliente();
            objeto[1] = listarClientes.get(i).getNombres();
            objeto[2] = listarClientes.get(i).getApellido_paterno();
            objeto[3] = listarClientes.get(i).getApellido_materno();
            objeto[4] = listarClientes.get(i).getTipo_documento();
            objeto[5] = listarClientes.get(i).getNum_documento();
            objeto[6] = listarClientes.get(i).getCelular();
            objeto[7] = listarClientes.get(i).getEmail();
            modeloTabla.addRow(objeto);
        }
        vista.tb_datos_cliente.setModel(modeloTabla);
    }

    private void limpiarTabla (){
        for(int i=0;i<vista.tb_datos_cliente.getRowCount();i++){
            modeloTabla.removeRow(i);
            i = i-1;
        }
    }

    private void limpiarCajas(){
        vista.txt_idCliente.setText("");
        vista.txt_nombre_cliente.setText("");
        vista.txt_apellido_p_cliente.setText("");
        vista.txt_apellido_m_cliente.setText("");
        vista.txt_tipo_doc_cliente.setText("");
        vista.txt_num_doc_cliente.setText("");
        vista.txt_cel_cliente.setText("");
        vista.txt_email_cliente.setText("");
    }
```

Ilustración 42: Código de programación del software - 8

Ilustración 43: Código de programación del software - 9

Ilustración 44: Código de programación del software - 10

Hotel - NetBeans IDE 8.2

```java
package models;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;


public class ClienteDAO {
    Conexion conectar = new Conexion();
    Connection con;
    PreparedStatement ps;
    ResultSet rs;

    public List<Cliente> listar(){
        List<Cliente> listarCliente = new ArrayList<>();
        String sql = "SELECT * FROM cliente";
        try {
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while(rs.next()){
                Cliente cli = new Cliente();
                cli.setIdCliente(rs.getInt(1));
                cli.setNombres(rs.getString(2));
                cli.setApellido_paterno(rs.getString(3));
                cli.setApellido_materno(rs.getString(4));
                cli.setTipo_documento(rs.getString(5));
                cli.setNum_documento(rs.getString(6));
                cli.setCelular(rs.getString(7));
                cli.setEmail(rs.getString(8));
                listarCliente.add(cli);
            }
        } catch (Exception e) {
            System.err.println("Error al listar: "+e.getMessage());
        }
        return listarCliente;
```

Ilustración 45: Código de programación del software - 11

Hotel - NetBeans IDE 8.2

```java
    public int agregar(Cliente cli){
        String sql = "INSERT INTO cliente VALUES(null,?,?,?,?,?,?,?)";
        try{
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            ps.setString(1, cli.getNombres());
            ps.setString(2, cli.getApellido_paterno());
            ps.setString(3, cli.getApellido_materno());
            ps.setString(4, cli.getTipo_documento());
            ps.setString(5, cli.getNum_documento());
            ps.setString(6, cli.getCelular());
            ps.setString(7, cli.getEmail());
            ps.executeUpdate();
        } catch (Exception e) {
            System.err.println("Error al agregar: "+e.getMessage());
            return 0;
        }
        return 1;
    }

    public int actualizar(Cliente cli){
        int r = 0;
        String sql = "UPDATE cliente SET nombres = ?, apellido_paterno = ?, apellido_materno = ?, tipo_documento = ?, num_documento = ?, celular = ?, emai
        try {
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            ps.setString(1, cli.getNombres());
            ps.setString(2, cli.getApellido_paterno());
            ps.setString(3, cli.getApellido_materno());
            ps.setString(4, cli.getTipo_documento());
            ps.setString(5, cli.getNum_documento());
            ps.setString(6, cli.getCelular());
            ps.setString(7, cli.getEmail());
            ps.setInt(8, cli.getIdCliente());
            r = ps.executeUpdate();
```

Ilustración 46: Código de programación del software - 12

```
76                  if(r == 1){
77                      return 1;
78                  }else{
79                      return 0;
80                  }
                } catch (Exception e) {
82                  System.err.println("Error al actualizar: "+e.getMessage());
83              }
84              return r;
85          }
86
87          public int eliminar(int idCliente){
88              String sql = "DELETE FROM cliente WHERE idCliente = "+idCliente;
89              try {
90                  con = conectar.conectarServidor();
91                  ps = con.prepareStatement(sql);
92                  ps.executeUpdate();
93                  return 1;
                } catch (Exception e) {
95                  System.err.println("Error al eliminar: "+e.getMessage());
96                  return 0;
97              }
98          }
99      }
```

Ilustración 47: Código de programación del software - 13

```
1   package models;
2
3   import java.sql.Connection;
4   import java.sql.DriverManager;
5
6   public class Conexion {
7       Connection con;
8       public Connection conectarServidor(){
9           String url = "jdbc:mysql://localhost:3306/proyecto?serverTimezone=America/Lima";
10          String user = "root";
11          String pass = "";
12          try {
13              Class.forName("com.mysql.cj.jdbc.Driver");
14              con = DriverManager.getConnection(url,user,pass);
            } catch (Exception e) {
16              System.err.println("No se conectó a la Base de Datos: "+e.getMessage());
17          }
18          return con;
19      }
20  }
```

Ilustración 48: Código de programación del software - 14

```java
package models;

public class Habitacion {
    private int idHabitacion;
    private String numero;
    private String piso;
    private Double precio;
    private String estado;
    private String tipo_Habitacion;

    public Habitacion(int idHabitacion, String numero, String piso, Double precio, String estado, String tipo_Habitacion) {
        this.idHabitacion = idHabitacion;
        this.numero = numero;
        this.piso = piso;
        this.precio = precio;
        this.estado = estado;
        this.tipo_Habitacion = tipo_Habitacion;
    }

    public Habitacion(String numero, String piso, Double precio, String estado, String tipo_Habitacion) {
        this.numero = numero;
        this.piso = piso;
        this.precio = precio;
        this.estado = estado;
        this.tipo_Habitacion = tipo_Habitacion;
    }

    public Habitacion() {
    }

    public int getIdHabitacion() {
        return idHabitacion;
    }

    public void setIdHabitacion(int idHabitacion) {
        this.idHabitacion = idHabitacion;
    }
```

Ilustración 49: Código de programación del software - 15

```java
package models;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

public class HabitacionDAO {
    Conexion conectar = new Conexion();
    Connection con;
    PreparedStatement ps;
    ResultSet rs;

    public List<Habitacion> listar(){
        List<Habitacion> listarHabitacion = new ArrayList<>();
        String sql = "SELECT * FROM habitacion";
        try {
            con = conectar.conectarServidor();
            ps = con.prepareStatement(sql);
            rs = ps.executeQuery();
            while(rs.next()){
                Habitacion hab = new Habitacion();
                hab.setIdHabitacion(rs.getInt(1));
                hab.setNumero(rs.getString(2));
                hab.setPiso(rs.getString(3));
                hab.setPrecio(rs.getDouble(4));
                hab.setEstado(rs.getString(5));
                hab.setTipo_Habitacion(rs.getString(6));
                listarHabitacion.add(hab);
            }
        } catch (Exception e) {
            System.err.println("Error al listar: "+e.getMessage());
        }
        return listarHabitacion;
    }
```

Ilustración 50: Código de programación del software - 16

Ilustración 51: Código de programación del software - 17

Ilustración 52: Código de programación del software - 18

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Trabajador
2	Actores	Administrador
3	Objetivo	Ingresar un nuevo trabajador a nuestro sistema.
4	Disparador	El administrador desea registrar un nuestro trabajador en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El trabajador será registrado exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de trabajadores • Administrador solicita la consulta de todos los trabajadores. • El sistema le muestra los datos de todos los trabajadores. Escenario 3: Modificar al trabajador • Administrador solicita modificar los datos del trabajador • El sistema le permite modificar los datos del trabajador y se realiza los cambios respectivos Escenario 4: Eliminar al trabajador • Administrador solicita eliminar un trabajador • El sistema le permite eliminar al trabajador. ➢ El administrador registra todos los datos del trabajador ➢ El sistema valida todos los datos del trabajador ➢ El sistema registra el nombre del trabajador ➢ El sistema registra el apellido paterno del trabajador ➢ El sistema registra el apellido materno del trabajador ➢ El sistema registra el tipo de documento del trabajador ➢ El sistema registra el número de documento del trabajador ➢ El sistema registra el celular del trabajador ➢ El sistema registra el email del trabajador ➢ El sistema registra el sueldo del trabajador

		➢ El sistema registra el acceso del trabajador ➢ El sistema registra el login del trabajador ➢ El sistema registra el password del trabajador ➢ El sistema registra el estado del trabajador
8	Escenario Alternativo	➢ Si no se registra el trabajador, enviar mensaje de ayuda. ➢ Si faltan datos del trabajador, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de un trabajador no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica datos de un trabajador no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina un trabajador no seleccionado, enviar mensaje de ayuda.
9	Prioridad	Versión 1

Tabla 3: Especificación de Casos de uso - Versión 1 - 1

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Clientes
2	Actores	Administrador y Recepcionista
3	Objetivo	Ingresar un nuevo cliente a nuestro sistema.
4	Disparador	El administrador o recepcionista desea registrar un nuestro cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El cliente será registrado exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador o recepcionista ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de clientes • Administrador o recepcionista solicita la consulta de todos los clientes. • El sistema le muestra los datos de todos los clientes. Escenario 3: Modificar al cliente • Administrador o recepcionista solicita modificar los datos del cliente • El sistema le permite modificar los datos del cliente y se realiza los cambios respectivos Escenario 4: Eliminar al cliente • Administrador o recepcionista solicita eliminar un cliente • El sistema le permite eliminar al cliente. ➢ El administrador o recepcionista registra todos los datos del cliente ➢ El sistema valida todos los datos del cliente ➢ El sistema registra el nombre del cliente ➢ El sistema registra el apellido paterno del cliente ➢ El sistema registra el apellido materno del cliente

		➢ El sistema registra el tipo de documento del cliente ➢ El sistema registra el número de documento del cliente ➢ El sistema registra el celular del cliente ➢ El sistema registra el email del cliente
8	Escenario Alternativo	➢ Si no se registra el cliente, enviar mensaje de ayuda. ➢ Si faltan datos del cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de un cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica datos de un cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina un cliente no seleccionado, enviar mensaje de ayuda.
9	Prioridad	Versión 1

Tabla 4: Especificación de Casos de uso - Versión 1 - 2

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de ingreso de cliente
2	Actores	Administrador
3	Objetivo	Ingresar el ingreso de cliente a nuestro sistema.
4	Disparador	El administrador desea registrar el ingreso del cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	El ingreso de cliente será registrado exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de ingreso de clientes • Administrador solicita la consulta de ingreso de todos los clientes. • El sistema le muestra el ingreso de todos los clientes. Escenario 3: Modificar el ingreso de cliente • Administrador solicita modificar el ingreso del cliente • El sistema le permite modificar el ingreso del cliente y se realiza los cambios respectivos Escenario 4: Eliminar el ingreso del cliente • Administrador solicita eliminar el ingreso de cliente • El sistema le permite eliminar el ingreso de cliente. ➢ El administrador registra el ingreso del cliente ➢ El sistema valida los datos de ingreso del cliente ➢ El sistema registra el ingreso del cliente
8	Escenario Alternativo	➢ Si no se registra el ingreso del cliente, enviar mensaje de ayuda. ➢ Si faltan datos en el ingreso de cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda.

		➢ Si se consulta el ingreso del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica el ingreso del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina el ingreso del cliente no seleccionado, enviar mensaje de ayuda.
9	Prioridad	Versión 1

Tabla 5: Especificación de Casos de uso - Versión 1 - 3

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de salida de cliente
2	Actores	Administrador
3	Objetivo	Ingresar la salida de cliente a nuestro sistema.
4	Disparador	El administrador desea registrar la salida del cliente en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	La salida de cliente será registrada exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de salida de clientes • Administrador solicita la consulta de salida de todos los clientes. • El sistema le muestra la salida de todos los clientes. Escenario 3: Modificar la salida de cliente • Administrador solicita modificar la salida del cliente • El sistema le permite modificar la salida del cliente y se realiza los cambios respectivos Escenario 4: Eliminar la salida del cliente • Administrador solicita eliminar la salida de cliente • El sistema le permite eliminar la salida de cliente. ➢ El administrador registra la salida del cliente ➢ El sistema valida los datos de salida del cliente ➢ El sistema registra la salida del cliente
8	Escenario Alternativo	➢ Si no se registra la salida del cliente, enviar mensaje de ayuda. ➢ Si faltan datos en la salida de cliente, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda.

		➢ Si se consulta la salida del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se modifica la salida del cliente no seleccionado, enviar mensaje de ayuda. ➢ Si se elimina la salida del cliente no seleccionado, enviar mensaje de ayuda.
9	Prioridad	Versión 1

Tabla 6: Especificación de Casos de uso - Versión 1 - 4

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Habitaciones
2	Actores	Administrador
3	Objetivo	Ingresar una habitación a nuestro sistema.
4	Disparador	El administrador desea registrar una habitación en el sistema
5	Pre condiciones	Se debe validar el trabajador mediante el Login
6	Post condiciones	La habitación será registrada exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de habitaciones • Administrador solicita la consulta de todas las habitaciones. • El sistema le muestra los datos de todas las habitaciones. Escenario 3: Modificar habitación • Administrador solicita modificar los datos de la habitación • El sistema le permite modificar los datos de la habitación y se realiza los cambios respectivos Escenario 4: Eliminar habitación • Administrador solicita eliminar una habitación • El sistema le permite eliminar la habitación ➢ El administrador registra todos los datos de la habitación ➢ El sistema valida todos los datos de la habitación ➢ El sistema registra el número de la habitación ➢ El sistema registra el piso de la habitación ➢ El sistema registra el precio de la habitación ➢ El sistema registra el tipo de habitación ➢ El sistema registra el estado de la habitación
8	Escenario Alternativo	➢ Si no se registra la habitación, enviar mensaje de ayuda. ➢ Si faltan datos de la habitación, enviar mensaje de ayuda.

		➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de una habitación no seleccionada, enviar mensaje de ayuda. ➢ Si se modifica datos de una habitación no seleccionada, enviar mensaje de ayuda. ➢ Si se elimina una habitación no seleccionada, enviar mensaje de ayuda.
9	Prioridad	Versión 2

Tabla 7: Especificación de Casos de uso - Versión 2 - 1

ID	ITEM	DESCRIPCIÓN
1	Nombre Corto	Registro de Reserva
2	Actores	Administrador y Recepcionista
3	Objetivo	Ingresar nueva reserva a nuestro sistema
4	Disparador	El administrador o recepcionista desea registrar una reserva en el sistema
5	Pre condiciones	• Validar trabajador mediante el Login • Validar registro del cliente
6	Post condiciones	La reserva será registrada exitosamente.
7	Escenario básico	Escenario 1: Validar trabajador. • Administrador o recepcionista ingresa su usuario y su contraseña • El sistema valida el usuario y contraseña Escenario 2: Consulta de reservas • Administrador o recepcionista solicita la consulta de todas las reservas. • El sistema le muestra los datos de todas las reservas. Escenario 3: Modificar reserva • Administrador o recepcionista solicita modificar reserva del cliente. • El sistema le permite modificar los datos de la reserva y se realiza los cambios respectivos Escenario 4: Eliminar reserva • Administrador o recepcionista solicita eliminar una reserva. • El sistema le permite eliminar la reserva. ➢ El administrador o recepcionista registra todos los datos de la reserva ➢ El sistema valida todos los datos de la reserva ➢ El sistema registra el tipo de reserva ➢ El sistema registra la fecha de reserva

		➢ El sistema registra la fecha de ingreso de la reserva ➢ El sistema registra la fecha de salida de la reserva ➢ El sistema registra el costo de la reserva ➢ El sistema registra el estado de la reserva
8	Escenario Alternativo	➢ Si no se registra la reserva, enviar mensaje de ayuda. ➢ Si faltan datos de la reserva, enviar mensaje de ayuda. ➢ Si el usuario no es válido, enviar mensaje de ayuda. ➢ Si la contraseña no es válida, enviar mensaje de ayuda. ➢ Si se consulta datos de una reserva no seleccionada, enviar mensaje de ayuda. ➢ Si se modifica datos de una reserva no seleccionada, enviar mensaje de ayuda. ➢ Si se elimina una reserva no seleccionada, enviar mensaje de ayuda.
9	Prioridad	Versión 2

Tabla 8: Especificación de Casos de uso - Versión 2 - 2

Caso de uso	Datos de Entrada normal	Resultado Esperado normal	Datos de entrada anómalo	Resultado esperado anómalo
Registro de cliente	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email	Cliente registrado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Faltan dígitos -Email ingresado incorrectamente	Debe completar todos los campos
Eliminar cliente	Seleccionar cliente de la tabla	Cliente eliminado con éxito	No selecciona cliente de la tabla	Debe seleccionar una fila
Modificar cliente	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email	Cliente actualizado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos -Email ingresado incorrectamente	Debe completar todos los campos
Consultar cliente	Si existen clientes en la base de datos	Cliente listado con éxito	No existen clientes en la base de datos	No hay clientes para listar
Registrar trabajador	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email Sueldo Acceso Login	Trabajador registrado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos	Debe completar todos los campos

	Password Estado		-Login inexistente -Password incorrecto	
Eliminar trabajador	Seleccionar al trabajador de la tabla	Trabajador eliminado con éxito	No selecciona al trabajador de la tabla	Debe seleccionar una fila
Modificar trabajador	Nombres Apellido Paterno Apellido Materno Tipo de Documento Número de Documento Celular Email Sueldo Acceso Login Password Estado	Trabajador actualizado con éxito	-Nombre en espacios -Apellido en espacios -Tipo de documento incorrecto -Número de documento inexistente -Falta dígitos -Login inexistente -Password incorrecto	Debe completar todos los campos
Consultar trabajador	Si existen trabajadores en la base de datos	Trabajador listado con éxito	No existen trabajadores en la base de datos	No hay trabajadores para listar
Validar trabajador	Usuario Contraseña	Trabajador validado con éxito	-Usuario inexistente -Contraseña incorrecta	Usuario o contraseña incorrecto
Registro de Entrada de Cliente	Seleccionar la entrada de cliente de la tabla	Entrada registrada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Eliminar la Entrada de Cliente	Seleccionar la entrada de cliente de la tabla	Movimiento eliminado con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Modificar la Entrada de Cliente	Seleccionar la entrada de cliente de la tabla	Entrada actualizada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Consultar la Entrada de Cliente	Si existen movimientos en la base de datos	Consulta realizada con éxito	No existen movimientos en la base de datos	No hay movimientos para listar
Registro de Salida de Cliente	Seleccionar la entrada de cliente de la tabla	Salida registrada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Eliminar la Salida de Cliente	Seleccionar la entrada de cliente de la tabla	Movimiento eliminado con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila

Modificar Salida de Cliente	Seleccionar la entrada de cliente de la tabla	Salida actualizada con éxito	No selecciona la entrada de cliente de la tabla	Debe seleccionar una fila
Consultar la Salida de Cliente	Si existen movimientos en la base de datos	Consulta realizada con éxito	No existen movimientos en la base de datos	No hay movimientos para listar

Tabla 9: Plan de pruebas Incremental – Versión 1

Caso de uso	**Datos de Entrada normal**	**Resultado Esperado normal**	**Datos de entrada anómalo**	**Resultado esperado anómalo**
Registro de reserva	Tipo de reserva Fecha de reserva Fecha de ingreso Fecha de salida Costo Estado	Reserva realizada con éxito	-Tipo de reserva incorrecta -Fecha de reserva inválida -Fecha de ingreso y salida inválida -Estado inválido	Debe completar todos los campos.
Modificar reserva	Tipo de reserva Fecha de reserva Fecha de ingreso Fecha de salida Costo Estado	Reserva modificada con éxito	-Tipo de reserva incorrecta -Fecha de reserva inválida -Fecha de ingreso y salida inválida -Estado inválido	Debe completar todos los campos.
Eliminar reserva	Seleccionar reserva de la tabla	Reserva eliminada con éxito	No selecciona reserva de la tabla	Debe seleccionar una fila
Consultar reserva	Si existen reservas en la base de datos	Consulta realizada con éxito	No existen reservas en la base de datos	No hay reservas para listar
Registro de habitación	Número Piso Descripción Características Precio diario Estado Tipo de habitación	Habitación registrada con éxito	-Número de habitación inválida -Piso inválido -Exceso de límites de caracteres -Estado inválido -Tipo de habitación inválida	Debe completar todos los campos.
Modificar habitación	Número Piso	Habitación modificada con éxito	-Número de habitación inválida -Piso inválido	Debe completar

	Descripción Características Precio diario Estado Tipo de habitación		-Exceso de límites de caracteres -Estado inválido -Tipo de habitación inválida	todos los campos.
Eliminar habitación	Seleccionar habitación de la tabla	Habitación eliminada con éxito	No selecciona habitación de la tabla	Debe seleccionar una fila
Consultar habitación	Si existen habitaciones en la base de datos	Consulta realizada con éxito	No existen habitaciones en la base de datos	No hay habitaciones para listar

Tabla 10: Plan de pruebas Incremental – Versión 2

Ejecución del Plan de Pruebas:

Caso de uso Registro de Cliente:

Ilustración 53: Registro de Cliente - Datos de entrada normal - Versión 1

Ilustración 54: Registro de Cliente - Resultado esperado normal - Versión 1

Ilustración 55: Registro de Cliente - Datos de entrada anómalo - Versión 1

Ilustración 56: Registro de Cliente - Resultado esperado anómalo - Versión 1

Caso de uso Eliminar Cliente:

Ilustración 57: Eliminar Cliente - Datos de entrada normal - Versión 1

Ilustración 58: Eliminar Cliente - Resultado esperado normal - Versión 1

Ilustración 59: Eliminar Cliente - Datos de entrada anómalo - Versión 1

Ilustración 60: Eliminar Cliente - Resultado esperado anómalo - Versión 1

Caso de uso Modificar Cliente:

Ilustración 61: Modificar Cliente - Datos de entrada normal - Versión 1

Ilustración 62: Modificar Cliente - Resultado esperado normal - Versión 1

Ilustración 63: Modificar Cliente - Datos de entrada anómalo - Versión 1

Ilustración 64: Modificar Cliente - Resultado esperado anómalo - Versión 1

Caso de uso Consultar Cliente:

Ilustración 65: Consultar Cliente - Datos de entrada normal - Versión 1

Ilustración 66: Consultar Cliente - Resultado esperado normal - Versión 1

Ilustración 67: Consultar Cliente - Datos de entrada anómalo - Versión 1

Ilustración 68: Consultar Cliente - Resultado esperado anómalo - Versión 1

Caso de uso Registrar Trabajador:

Ilustración 69: Registrar Trabajador - Datos de entrada normal - Versión 1

Ilustración 70: Registrar Trabajador - Resultado esperado normal - Versión 1

Ilustración 71: Registrar Trabajador - Datos de entrada anómalo - Versión 1

Ilustración 72: Registrar Trabajador - Resultado esperado anómalo - Versión 1

Caso de uso Eliminar Trabajador:

Ilustración 73: Eliminar Trabajador - Datos de entrada normal - Versión 1

Ilustración 74: Eliminar Trabajador - Resultado esperado normal - Versión 1

Ilustración 75: Eliminar Trabajador - Datos de entrada anómalo - Versión 1

Ilustración 76: Eliminar Trabajador - Resultado esperado anómalo - Versión 1

Caso de uso Modificar Trabajador:

Ilustración 77: Modificar Trabajador - Datos de entrada normal - Versión 1

Ilustración 78: Modificar Trabajador - Resultado esperado normal - Versión 1

Ilustración 79: Modificar Trabajador - Datos de entrada anómalo - Versión 1

Ilustración 80: Modificar Trabajador - Resultado esperado anómalo - Versión 1

Caso de uso Consultar Trabajador:

Ilustración 81: Consultar Trabajador - Datos de entrada normal - Versión 1

Ilustración 82: Consultar Trabajador - Resultado esperado normal - Versión 1

Ilustración 83: Consultar Trabajador - Datos de entrada anómalo - Versión 1

Ilustración 84: Consultar Trabajador - Resultado esperado anómalo - Versión 1

Caso de uso Validar Trabajador:

Ilustración 85: Validar Trabajador - Datos de entrada normal - Versión 1

Ilustración 86: Validar Trabajador - Resultado esperado normal - Versión 1

Ilustración 87: Validar Trabajador - Datos de entrada anómalo - Versión 1

Ilustración 88: Validar Trabajador - Resultado esperado anómalo - Versión 1

Caso de uso Registro de Entrada Cliente:

Ilustración 89: Registro de entrada de cliente - Datos de entrada normal - Versión 1

Ilustración 90: Registro de entrada de cliente - Resultado esperado normal - Versión 1

Ilustración 91: Registro de entrada de cliente - Datos de entrada anómalo - Versión 1

Ilustración 92: Registro de entrada de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Eliminar la Entrada Cliente:

Ilustración 93: Eliminar la entrada de cliente - Datos de entrada normal - Versión 1

Ilustración 94: Eliminar la entrada de cliente - Resultado esperado normal - Versión 1

Ilustración 95: Eliminar la entrada de cliente - Datos de entrada anómalo - Versión 1

Ilustración 96: Eliminar la entrada de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Modificar la Entrada Cliente:

Ilustración 97: Modificar la entrada de cliente - Datos de entrada normal - Versión 1

Ilustración 98: Modificar la entrada de cliente - Resultado esperado normal - Versión 1

Ilustración 99: Modificar la entrada de cliente - Datos de entrada anómalo - Versión 1

Ilustración 100: Modificar la entrada de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Consultar la Entrada Cliente:

Ilustración 101: Consultar la entrada de cliente - Datos de entrada normal - Versión 1

Ilustración 102: Consultar la entrada de cliente - Resultado esperado normal - Versión 1

Ilustración 103: Consultar la entrada de cliente - Datos de entrada anómalo - Versión 1

Ilustración 104: Consultar la entrada de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Registro de Salida Cliente:

Ilustración 105: Registro de salida de cliente - Datos de entrada normal - Versión 1

Ilustración 106: Registro de salida de cliente - Resultado esperado normal - Versión 1

Ilustración 107: Registro de salida de cliente - Datos de entrada anómalo - Versión 1

Ilustración 108: Registro de salida de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Eliminar la Salida Cliente:

Ilustración 109: Eliminar la salida de cliente - Datos de entrada normal - Versión 1

Ilustración 110: Eliminar de salida de cliente - Resultado esperado normal - Versión 1

Ilustración 111: Eliminar de salida de cliente - Datos de entrada anómalo - Versión 1

Ilustración 112: Eliminar de salida de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Modificar Salida de Cliente:

Ilustración 113: Modificar de salida de cliente - Datos de entrada normal - Versión 1

Ilustración 114: Modificar de salida de cliente - Resultado esperado normal - Versión 1

Ilustración 115: Modificar de salida de cliente - Datos de entrada anómalo - Versión 1

Ilustración 116: Modificar de salida de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Consultar la Salida Cliente:

Ilustración 117: Consultar la salida de cliente - Datos de entrada normal - Versión 1

Ilustración 118: Consultar la salida de cliente - Resultado esperado normal - Versión 1

Ilustración 119: Consultar la salida de cliente - Datos de entrada anómalo - Versión 1

Ilustración 120: Consultar la salida de cliente - Resultado esperado anómalo - Versión 1

Caso de uso Registrar Habitaciones:

Ilustración 121: Registrar Habitaciones - Datos de entrada normal - Versión 2

Ilustración 122: Registrar Habitaciones - Resultado esperado normal - Versión 2

Ilustración 123: Registrar Habitaciones - Datos de entrada anómalo - Versión 2

Ilustración 124: Registrar Habitaciones - Resultado esperado anómalo - Versión 2

Caso de uso Eliminar Habitaciones:

Ilustración 125: Eliminar Habitaciones - Datos de entrada normal - Versión 2

Ilustración 126: Eliminar Habitaciones - Resultado esperado normal - Versión 2

Ilustración 127: Eliminar Habitaciones - Datos de entrada anómalo - Versión 2

Ilustración 128: Eliminar Habitaciones - Resultado esperado anómalo - Versión 2

Caso de uso Modificar Habitaciones:

Ilustración 129: Modificar Habitaciones - Datos de entrada normal - Versión 2

Ilustración 130: Modificar Habitaciones - Resultado esperado normal - Versión 2

Ilustración 131: Modificar Habitaciones - Datos de entrada anómalo - Versión 2

Ilustración 132: Modificar Habitaciones - Resultado esperado anómalo - Versión 2

Caso de uso Consultar Habitaciones:

Ilustración 133: Consultar Habitaciones - Datos de entrada normal - Versión 2

Ilustración 134: Consultar Habitaciones - Resultado esperado normal - Versión 2

Ilustración 135: Consultar Habitaciones - Datos de entrada anómalo - Versión 2

Ilustración 136: Consultar Habitaciones - Resultado esperado anómalo - Versión 2

Caso de uso Registrar Reserva:

Ilustración 137: Registrar Reserva - Datos de entrada normal - Versión 2

Ilustración 138: Registrar Reserva - Resultado esperado normal - Versión 2

Ilustración 139: Registrar Reserva - Datos de entrada anómalo - Versión 2

Ilustración 140: Registrar Reserva - Resultado esperado anómalo - Versión 2

Caso de uso Eliminar Reserva:

Ilustración 141: Eliminar Reserva - Datos de entrada normal - Versión 2

Ilustración 142: Eliminar Reserva - Resultado esperado normal - Versión 2

Ilustración 143: Eliminar Reserva - Datos de entrada anómalo - Versión 2

Ilustración 144: Eliminar Reserva - Resultado esperado anómalo - Versión 2

Caso de uso Modificar Reserva:

Ilustración 145: Modificar Reserva - Datos de entrada normal - Versión 2

Ilustración 146: Modificar Reserva - Resultado esperado normal - Versión 2

Ilustración 147: Modificar Reserva - Datos de entrada anómalo - Versión 2

Ilustración 148: Modificar Reserva - Resultado esperado anómalo - Versión 2

Caso de uso Consultar Reserva:

Ilustración 149: Consultar Reserva - Datos de entrada normal - Versión 2

Ilustración 150: Consultar Reserva - Resultado esperado normal - Versión 2

Ilustración 151: Consultar Reserva - Datos de entrada anómalo - Versión 2

Ilustración 152: Consultar Reserva - Resultado esperado anómalo - Versión 2

Hallazgos

Clientes

- Tipo Documento acepta números cuando sólo debe aceptar texto
- Número de Documento acepta letras cuando sólo debe aceptar números
- Celular acepta letras cuando sólo debe aceptar números
- Email se registra en la base de datos aun cuando no se ingresa el @

Trabajador

- Cuando se ingresa letras en el campo Sueldo, no muestra mensaje de ayuda.
- Cuando se ingresa otro tipo de dato en Estado, no muestra mensaje de ayuda.
- Tipo de Documento acepta números cuando sólo debe aceptar texto.
- Número de Documento acepta letras cuando sólo debe aceptar números
- Celular acepta letras cuando sólo debe aceptar números
- Email se registra en la base de datos aun cuando no se ingresa el @
- Login duplicado si acepta el sistema cuando debería ser único.
- Password se registra con un sólo valor, cuando debería haber un mínimo de 4 caracteres

MODELO FURPS+ APLICADO AL PROYECTO AL 27/10/2020

FACTOR	NRO	PREGUNTA	ARTEFACTO	SI CUMPLE	NO CUMPLE	ACCCIÓN CORRECTIVA	RESPONSABLE	FECHA SOLUCIÓN	EVIDENCIA	Puntaje obtenido PO	Puntaje deseado PD	Grado Apego = (PO/PD) * 100%
Funcionalidad versión o incremento 1	1	¿El software ejecuta el caso de uso Registro de trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas	60	85	70.59%
	2	¿El software ejecuta el caso de uso Modificar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	3	¿El software ejecuta el caso de uso Eliminar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	4	¿El software ejecuta el caso de uso Consulta de trabajadores?	Software funcionando	X					Sección ejecución del plan de pruebas			
	5	¿El software ejecuta el caso de uso Validar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	6	¿El software ejecuta el caso de uso Registro de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			

	7	¿El software ejecuta el caso de uso Modificar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	8	¿El software ejecuta el caso de uso Eliminar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	9	¿El software ejecuta el caso de uso Consulta de clientes?	Software funcionando	X					Sección ejecución del plan de pruebas			
	10	¿El software ejecuta el caso de uso Registro de Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	11	¿El software ejecuta el caso de uso Eliminar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Abihail Zapata	27/10/2020	Sección ejecución del plan de pruebas			
	12	¿El software ejecuta el caso de uso Modificar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Diego Tasaico	27/10/2020	Sección ejecución del plan de pruebas			
	13	¿El software ejecuta el caso de uso Consultar la Entrada de cliente?	Software funcionando		X	Incluir caso de uso	Mauricio Villanueva	27/10/2020	Sección ejecución del plan de pruebas			

	14	¿El software ejecuta el caso de uso Registro de Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	15	¿El software ejecuta el caso de uso Eliminar la Salida de cliente?	Software funcionando		X	Incluir caso de uso	Olenka Lazo	27/10/2020	Sección ejecución del plan de pruebas			
	16	¿El software ejecuta el caso de uso Modificar Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	17	¿El software ejecuta el caso de uso Consultar la Salida de cliente?	Software funcionando		X	Incluir caso de uso	Mauricio Villanueva	27/10/2020	Sección ejecución del plan de pruebas			
Funcionalidad versión o incremento 2	18	¿El software ejecuta el caso de uso Registro de reserva?	Código fuente		X	Incluir caso de uso	Diego Tasaico	28/10/2020	Sección ejecución del plan de pruebas	0	40	0%
	19	¿El software ejecuta el caso de uso Modificar reserva?	Código fuente		X	Incluir caso de uso	Diego Tasaico	28/10/2020	Sección ejecución del plan de pruebas			
	20	¿El software ejecuta el caso de uso Eliminar reserva?	Código fuente		X	Incluir caso de uso	Mauricio Villanueva	28/10/2020	Sección ejecución del plan de pruebas			
	21	¿El software ejecuta el caso de uso Consultar reserva?	Código fuente		X	Incluir caso de uso	Olenka Lazo	29/10/2020	Sección ejecución del plan de pruebas			

	22	¿El software ejecuta el caso de uso Registro de habitación?	Código fuente		X	Incluir caso de uso	Olenka Lazo	29/10/2020	Sección ejecución del plan de pruebas			
	23	¿El software ejecuta el caso de uso Modificar habitación?	Código fuente		X	Incluir caso de uso	Mauricio Villanueva	30/10/2020	Sección ejecución del plan de pruebas			
	24	¿El software ejecuta el caso de uso Eliminar habitación?	Código fuente		X	Incluir caso de uso	Abihail Zapata	30/10/2020	Sección ejecución del plan de pruebas			
	25	¿El software ejecuta el caso de uso Consultar habitación?	Código fuente		X	Incluir caso de uso	Abihail Zapata	30/10/2020	Sección ejecución del plan de pruebas			
Usabilidad	21	¿Se muestra las imágenes de la carpeta de software y de informe en el GitHub?	GitHub	X					Sección repositorio GitHub	20	25	80%
	22	¿Si los matices de color de texto y de fondo son cómodos para el usuario?	Software funcionando	X					Sección funcionamiento del software			
	23	¿Es de fácil uso para el usuario?	Software funcionando	X					Sección funcionamiento del software			

	24	¿Permite una fácil salida del software?	Software funcionando		X	Corregir la salida del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software			
	25	¿Se adapta al tipo de usuario?	Software funcionando	X					Sección funcionamiento del software			
Confiablidad	26	¿Si el software controla el acceso no disponible de la Base de datos?	Software funcionando		X	Controlar el acceso a Base de Datos si no se conecta.	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software	5	25	20%
	27	¿Es de uso seguro para los usuarios?	Software funcionando		X	Corregir seguridad del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software			
	28	¿Es resistente ante fallos?	Software funcionando		X	Corregir la resistencia de fallos del sistema	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
	29	¿Todas las partes del sistema se encuentran conectadas entre sí?	Software funcionando	X					Sección funcionamiento del software			
	30	¿Tiene un comportamiento adecuado ante situaciones inusuales?	Software funcionando		X	Corregir la respuesta ante situaciones inusuales	Neftali Zapata	31/10/2020	Sección funcionamiento de software			
Rendimiento	31	¿Si el tiempo de respuesta es menor a 7 segundos?	Software funcionando	X					Sección funcionamiento del software	15	25	60%

	32	¿Si el software ocupa espacio de memoria mínima?	Administrador de tareas del SO		X	Corrección de espacio de memoria	Neftali Zapata	31/10/2020	Sección de funcionamiento de software			
	33	¿Disponible las 24 horas del día?	Software funcionando	X					Sección funcionamiento del software			
	34	¿Soluciona errores?	Software funcionando		X	Mejorar la solución de errores	Olenka Lazo	31/10/2]20	Sección funcionamiento de software			
	35	¿Detecta errores?	Software funcionando	X					Sección funcionamiento del software			
Soporte	36	¿Si el software cuenta con un procedimiento de compilación e instalación?	Procedimiento		X	Completar procedimiento	Mauricio Villanueva	31/10/2020	Sección procedimientos	0	25	0%
	37	¿Cuenta con un resguardo de datos?	Software funcionando		X	Crear la opción de resguardo de datos	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
	38	¿Puede ejecutarse ante alguna condición extrema?	Software funcionando		X	Corregir al ejecutar	Olenka Lazo	31/10/2020	Sección funcionamiento de software			
	39	¿Detecta errores en el sistema de manera rápida?	Software funcionando		X	Corregir el tiempo de solución	Neftali Zapata	31/10/2020	Sección funcionamiento de software			

	40	¿Cuenta con una opción de ayuda al usuario?	Software funcionando		X	Implementar opción de ayuda	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
PLUS (+)	41	¿Se ha completado el marco teórico con citas y referencias?	Informe		X	Completar el marco teórico con citas y referencias	Olenka Zapata	01/11/2020	Sección marco teórico del informe	5	20	25%
	42	¿Se ha completado el plan de pruebas de la versión 1 y 2?	Informe	X					Sección plan de pruebas incremental			
	43	¿Se ha completado la ejecución de pruebas de la versión 1 y 2?	Informe		X	Falta completar plan de pruebas en versión 2	Mauricio Villanueva	01/11/2020	Sección ejecución de plan de pruebas			
	44	¿Se ha completado las conclusiones y recomendaciones de la versión 2?	Informe		X	Completar conclusiones y recomendaciones de la versión 2	Abihail Zapata	01/11/2020	Sección conclusiones y recomendaciones			

Tabla 11: Modelo FURPS - 1

MODELO FURPS+ APLICADO AL PROYECTO AL 01/11/2020

FACTOR	NRO	PREGUNTA	ARTEFACTO	SI CUMPLE	NO CUMPLE	ACCCIÓN CORRECTIVA	RESPONSABLE	FECHA SOLUCIÓN	EVIDENCIA	Puntaje obtenido PO	Puntaje deseado PD	Grado Apego = (PO/PD) * 100%
Funcionalidad versión o incremento 1	1	¿El software ejecuta el caso de uso Registro de trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas	85	85	100%
	2	¿El software ejecuta el caso de uso Modificar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	3	¿El software ejecuta el caso de uso Eliminar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	4	¿El software ejecuta el caso de uso Consulta de trabajadores?	Software funcionando	X					Sección ejecución del plan de pruebas			
	5	¿El software ejecuta el caso de uso Validar trabajador?	Software funcionando	X					Sección ejecución del plan de pruebas			
	6	¿El software ejecuta el caso de uso Registro de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			

	7	¿El software ejecuta el caso de uso Modificar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	8	¿El software ejecuta el caso de uso Eliminar cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	9	¿El software ejecuta el caso de uso Consulta de clientes?	Software funcionando	X					Sección ejecución del plan de pruebas			
	10	¿El software ejecuta el caso de uso Registro de Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	11	¿El software ejecuta el caso de uso Eliminar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	12	¿El software ejecuta el caso de uso Modificar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	13	¿El software ejecuta el caso de uso Consultar la Entrada de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			

	14	¿El software ejecuta el caso de uso Registro de Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	15	¿El software ejecuta el caso de uso Eliminar la Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	16	¿El software ejecuta el caso de uso Modificar Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
	17	¿El software ejecuta el caso de uso Consultar la Salida de cliente?	Software funcionando	X					Sección ejecución del plan de pruebas			
Funcionalidad versión o incremento 2	18	¿El software ejecuta el caso de uso Registro de reserva?	Código fuente	X					Sección ejecución del plan de pruebas	40	40	100%
	19	¿El software ejecuta el caso de uso Modificar reserva?	Código fuente	X					Sección ejecución del plan de pruebas			
	20	¿El software ejecuta el caso de uso Eliminar reserva?	Código fuente	X					Sección ejecución del plan de pruebas			
	21	¿El software ejecuta el caso de uso Consultar reserva?	Código fuente	X					Sección ejecución del plan de pruebas			

	22	¿El software ejecuta el caso de uso Registro de habitación?	Código fuente	X					Sección ejecución del plan de pruebas			
	23	¿El software ejecuta el caso de uso Modificar habitación?	Código fuente	X					Sección ejecución del plan de pruebas			
	24	¿El software ejecuta el caso de uso Eliminar habitación?	Código fuente	X					Sección ejecución del plan de pruebas			
	25	¿El software ejecuta el caso de uso Consultar habitación?	Código fuente	X					Sección ejecución del plan de pruebas			
Usabilidad	21	¿Se muestra las imágenes de la carpeta de software y de informe en el GitHub?	GitHub	X					Sección repositorio GitHub	20	25	80%
	22	¿Si los matices de color de texto y de fondo son cómodos para el usuario?	Software funcionando	X					Sección funcionamiento del software			
	23	¿Es de fácil uso para el usuario?	Software funcionando	X					Sección funcionamiento del software			

	24	¿Permite una fácil salida del software?	Software funcionando		X	Corregir la salida del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software			
	25	¿Se adapta al tipo de usuario?	Software funcionando	X					Sección funcionamiento del software			
Confiablidad	26	¿Si el software controla el acceso no disponible de la Base de datos?	Software funcionando		X	Controlar el acceso a Base de Datos si no se conecta.	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software	5	25	20%
	27	¿Es de uso seguro para los usuarios?	Software funcionando		X	Corregir seguridad del sistema	Mauricio Villanueva	31/10/2020	Sección funcionamiento de software			
	28	¿Es resistente ante fallos?	Software funcionando		X	Corregir la resistencia de fallos del sistema	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
	29	¿Todas las partes del sistema se encuentran conectadas entre sí?	Software funcionando	X					Sección funcionamiento del software			
	30	¿Tiene un comportamiento adecuado ante situaciones inusuales?	Software funcionando		X	Corregir la respuesta ante situaciones inusuales	Neftali Zapata	31/10/2020	Sección funcionamiento de software			
Rendimiento	31	¿Si el tiempo de respuesta es menor a 7 segundos?	Software funcionando	X					Sección funcionamiento del software	15	25	60%

	32	¿Si el software ocupa espacio de memoria mínima?	Administrador de tareas del SO		X	Corrección de espacio de memoria	Neftali Zapata	31/10/2020	Sección de funcionamiento de software			
	33	¿Disponible las 24 horas del día?	Software funcionando	X					Sección funcionamiento del software			
	34	¿Soluciona errores?	Software funcionando		X	Mejorar la solución de errores	Olenka Lazo	31/10/2]20	Sección funcionamiento de software			
	35	¿Detecta errores?	Software funcionando	X					Sección funcionamiento del software			
Soporte	36	¿Si el software cuenta con un procedimiento de compilación e instalación?	Procedimiento		X	Completar procedimiento	Mauricio Villanueva	31/10/2020	Sección procedimientos	0	25	0%
	37	¿Cuenta con un resguardo de datos?	Software funcionando		X	Crear la opción de resguardo de datos	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
	38	¿Puede ejecutarse ante alguna condición extrema?	Software funcionando		X	Corregir al ejecutar	Olenka Lazo	31/10/2020	Sección funcionamiento de software			
	39	¿Detecta errores en el sistema de manera rápida?	Software funcionando		X	Corregir el tiempo de solución	Neftali Zapata	31/10/2020	Sección funcionamiento de software			

	40	¿Cuenta con una opción de ayuda al usuario?	Software funcionando		X	Implementar opción de ayuda	Diego Tasaico	31/10/2020	Sección funcionamiento de software			
PLUS (+)	41	¿Se ha completado el marco teórico con citas y referencias?	Informe		X	Completar el marco teórico con citas y referencias	Olenka Zapata	01/11/2020	Sección marco teórico del informe	10	20	50%
	42	¿Se ha completado el plan de pruebas de la versión 1 y 2?	Informe	X					Sección plan de pruebas incremental			
	43	¿Se ha completado la ejecución de pruebas de la versión 1 y 2?	Informe		X	Falta completar plan de pruebas en versión 2	Mauricio Villanueva	01/11/2020	Sección ejecución de plan de pruebas			
	44	¿Se ha completado las conclusiones y recomendaciones de la versión 2?	Informe	X					Sección conclusiones y recomendaciones			

Tabla 12: Modelo FURPS - 2

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

12. Conclusiones

- Se presentó a la empresa una solicitud para obtener el permiso respectivo, mencionándole el objetivo del estudio. Una vez aceptado el documento se coordinó con el administrador, para la recolección de datos con el fin de obtener las facilidades pertinentes para llevar a cabo el estudio.

- La recopilación de información permitió comprender mejor el sistema, que después fue representado a través de los diagramas utilizados. De dicho análisis, del cual se pudo extraer los requerimientos del sistema que sirven de base para el desarrollo de este.

- Se desarrolló el sistema utilizando NetBeans IDE 8.2 como lenguaje de programación y SQL server para la gestión de la base de datos, logrando la automatización eficiente del sistema.

- Se utilizó adicionalmente, las herramientas informáticas: Lucidchart, MySQL WorkBench y Balsamiq que nos permitió realizar los diagramas y prototipos, logrando así un mejor desarrollo y conceptualización básica que requiere el sistema.

- Por último, se cumplió con el objetivo del sistema, permitiendo así la implementación de sus respectivas opciones entre ellas mantenimiento de clientes, donde se podrá registrar, consultar, modificar y eliminar a los clientes, incluyendo la entrada y salida del cliente.

- Esta versión del sistema permite la administración de los procesos de registro de habitaciones y reservas, además del control de hospedaje de los huéspedes durante su estadía, hasta la salida de las instalaciones del hotel.

- El diseño de la base de datos asegura la integridad y confidencialidad de los datos, además de la centralización de la información. Asimismo, el diseño de interfaz permite una visualización previa de las pantallas de los nuevos registros que se van a utilizar en el sistema.

- El sistema desarrollado se ajusta a las diversas formas existentes para controlar el hospedaje y reservación, esto gracias a los parámetros con los que cuenta el sistema.

- Los nuevos casos de uso permitieron adaptar de mejor manera el desarrollo del sistema con una serie de requerimientos los cuales fueron creciendo mediante se iba avanzando en la elaboración del sistema.

- Para el funcionamiento del sistema se llevó a cabo la ejecución de planes de prueba, los cuales tuvo como finalidad poder encontrar las debilidades del sistema y a su vez poder reforzarlos.

13. Recomendaciones

- Se recomienda continuar la realización de los próximos prototipos en la aplicación Balsamiq ya que nos permite comprender de manera visual el proyecto y explorar más opciones.

- Para seguir mejorando el sistema de administración hotelera se ha recomendado seguir solicitando información a la empresa para así culminar los objetivos proyectados.

- Al ser el presente proyecto, un sistema desarrollado a base del mantenimiento de registro de clientes y trabajadores, se recomienda la implementación del mantenimiento de las habitaciones y reservas para un mejor manejo.

- Se recomienda usar el mensaje de ayuda con el objetivo de resolver algún inconveniente que se presente.

- Y finalmente, se hará un énfasis a la revisión constante de las funciones y procesos, debido a que existirán cambios en cada una de las funcionalidades a fin de así poder mejorar el desempeño del sistema.

- Se recomienda efectuar los respaldos del sistema periódicamente con el fin de disponer de un medio para recuperación de datos inscritos por última vez, en caso de que haya ocurrido un problema o perdida.

- Es recomendable seguir investigando de manera continua para así ir actualizando la información del sistema.

- La optimización de procesos y mejora continua es indispensable para reducir fallos en las áreas operativas del sistema.

- Es recomendable observar la calidad del sistema, si se realiza un uso correcto y el grado de satisfacción para asegurar el resultado que tiene este en el rendimiento.

- Por último, para seguir mejorando su productividad es recomendable implementar un sistema de reporte para la versión 3.

14. Referencias Bibliográficas

Alonso, F., Martínez, L., & Segovia , J. (2005). *Introducción a la ingenieria de Software.* Zaragoza : Delta Publicaciones.

Git Hub. (s.f.). Obtenido de https://github.com/

Lucidchart. (s.f.). Obtenido de https://www.lucidchart.com/pages/es

MySQL. (s.f.). Obtenido de https://www.mysql.com/

MySQL WorkBench. (s.f.). Obtenido de https://www.mysql.com/products/workbench/

NetBeans. (s.f.). Obtenido de https://netbeans.org/

Sommerville, I. (2011). *Ingeniería de software (9a. ed.).* Pearson Educación.

WampServer. (s.f.). Obtenido de https://www.wampserver.com/en/

15. Anexos

Ilustración 153: Videoconferencia con representante de la empresa

Ilustración 154: Integrante - Diego Tasaico

Ilustración 155: Integrante - Mauricio Villanueva

Ilustración 156: Integrante - Olenka Lazo

Ilustración 157: Integrante - Abihail Zapata

Starlight Inn

Pisco, 13 octubre del 2020

Srs. Representantes

Universidad Privada del Norte

Con la finalidad de completar y modernizar nuestro sistema administrativo, se autoriza para que los jóvenes estudiantes (Diego Tasaico, Mauricio Villanueva, Abihail Zapata y Olenka Lazo), realicen un programa virtual que nos facilite nuestro registro diario que actualmente usamos. De ser necesario, se aprueba la visita a las instalaciones del HOTEL, así como cualquier otra información que ayude en la elaboración de este importante proyecto.

Atte.

Eduardo Enrique Albarracin Trillo

Administrador

Ilustración 158: Acta de compromiso

Funcionamiento del Software:

Vista de entrada al sistema:

Ilustración 159: Funcionamiento del Software - Acceso al sistema

Ilustración 160: Funcionamiento del Software - Menú Principal

Ilustración 161: Funcionamiento del Software - Vista de los clientes

Ilustración 162: Funcionamiento del Software - Vista de los trabajadores

Ilustración 163: Funcionamiento del Software - Vista de los movimientos

Ilustración 164: Funcionamiento del Software - Vista de las reservas

Ilustración 165: Funcionamiento del Software - Vista de las habitaciones