



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	23
12. Referencias Bibliográficas	23
13. Anexos	24

ÍNDICE DE TABLAS

Tabla 1: Proceso de Negocio	4
Tabla 2: Trazabilidad de Requisitos	9

ÍNDICE DE FIGURAS

Ilustración 1: Proceso de Negocio.....	4
Ilustración 2: Proceso de Software.....	5
Ilustración 3: Diagrama de Casos de uso	5
Ilustración 4: Prototipo	6
Ilustración 5: Prototipo	7
Ilustración 6: Diagrama de actividades con particiones o calles.....	8

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 el registro del cliente donde se solicitará sus datos personales.
- Añadir en registro del cliente, la fecha de ingreso y salida del cliente.
- 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.
- Git hub: 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.
- Lucidchart: 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.

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

CAPÍTULO 3. DESARROLLO DE LA INVESTIGACIÓN

11. Implementación de la solución planteada

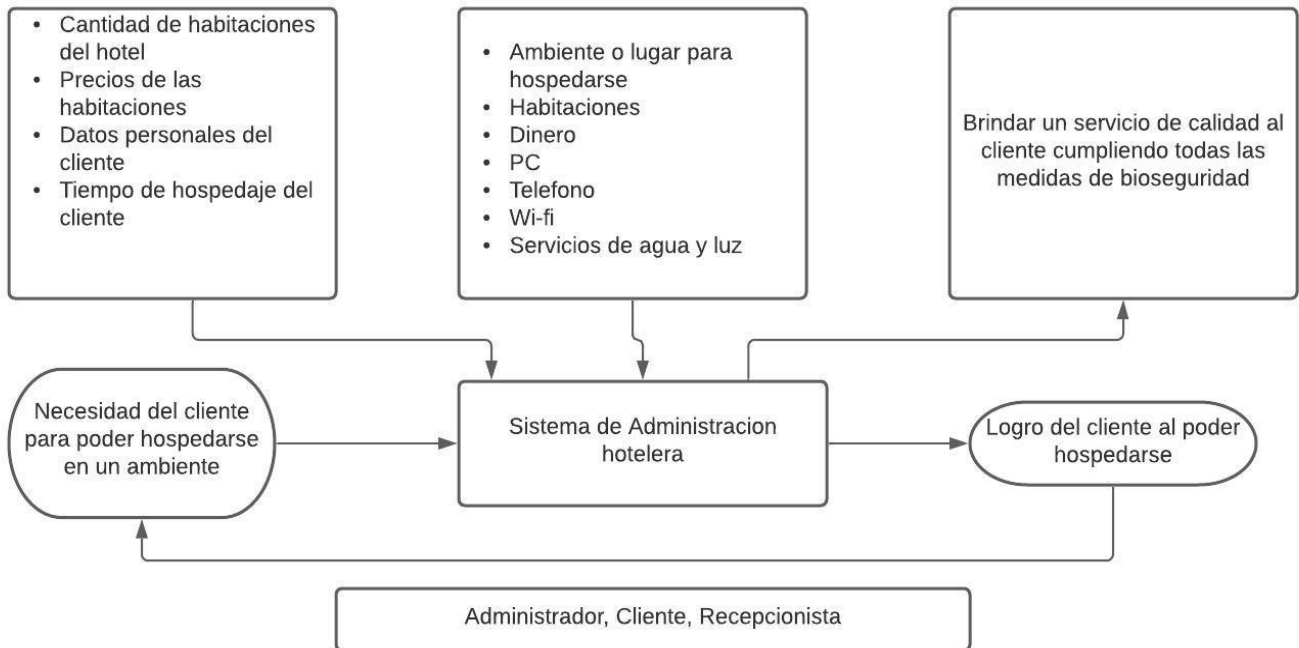


Ilustración 1: Proceso de Negocio

Tabla 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	<ul style="list-style-type: none"> 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

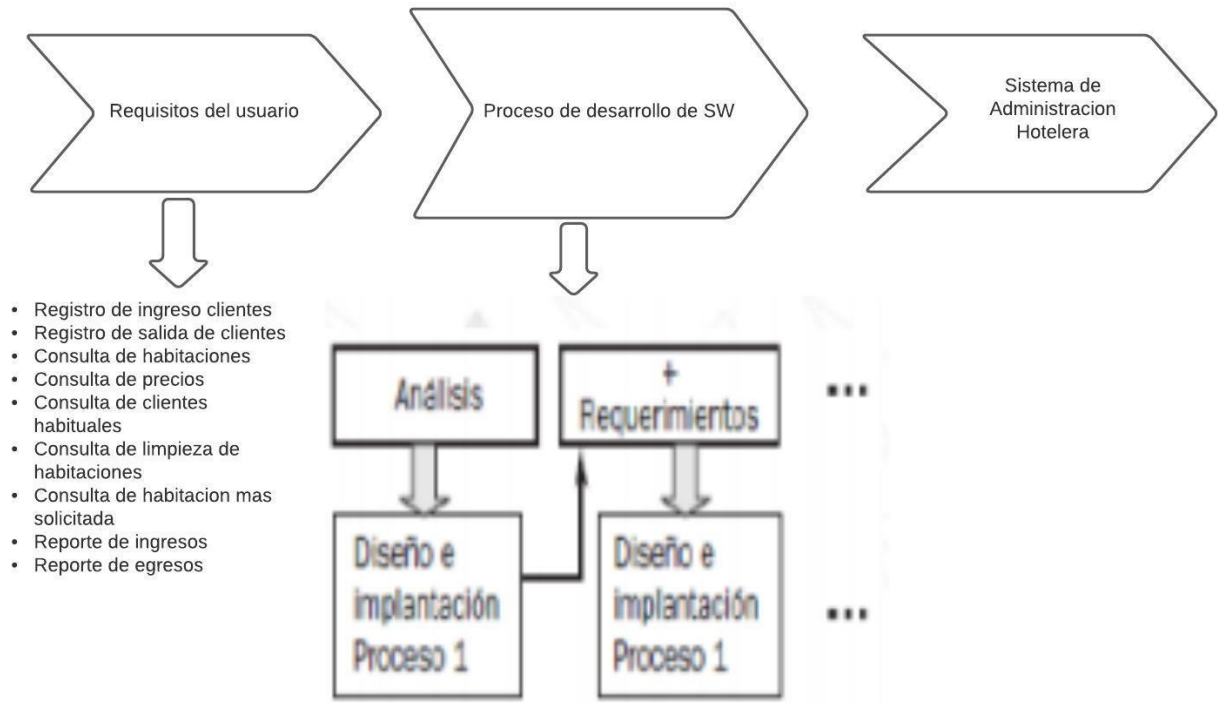


Ilustración 2: Proceso de Software

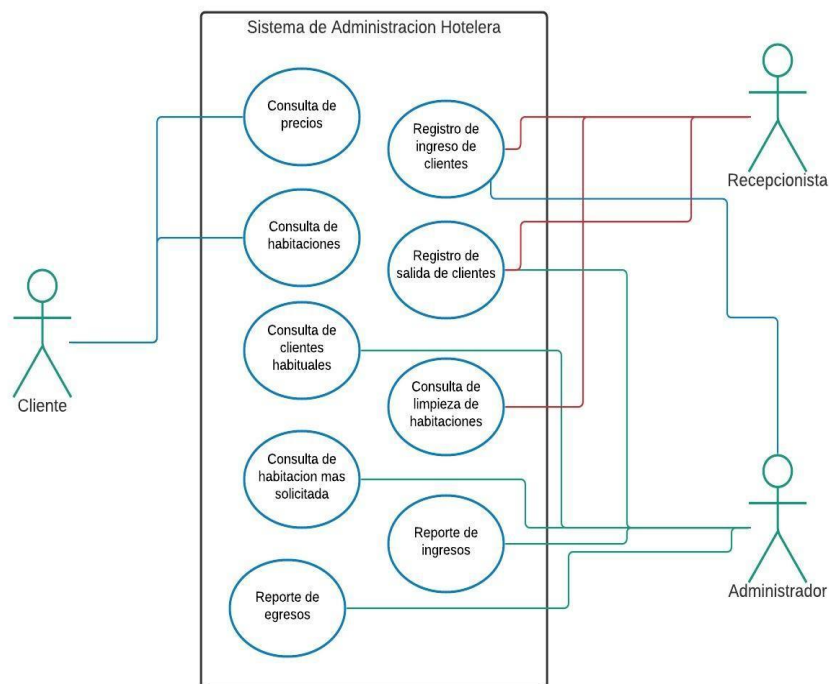



Ilustración 3: Diagrama de Casos de uso

Acceso al sistema



Usuario:


Contraseña:

Registro de cliente

Nombres


Apellidos

DNI

Fecha de nacimiento 

E-mail

Teléfono

Fecha de ingreso 


Fecha de salida 

Ilustración 4: Prototipo

Consultas de habitaciones					
Reserva	Ingreso	Salida	Cliente	Habitación	Estado
51	30/07/2020	31/07/2020	Mateo Cardenas	301	Ocupada
52	30/07/2020	01/08/2020	Lucia Rodriguez	202	Sin ocupar
53	30/07/2020	01/08/2020	Ariana Mendez	201	Ocupada
54	31/08/2020	02/08/2020	Hector Vilchez	402	Reserva
55	01/08/2020	02/08/2020	Roberto Gomez	303	Reserva
56	02/08/2020	03/08/2020	Daniel Zamora	103	Ocupada

Ilustración 5: Prototipo

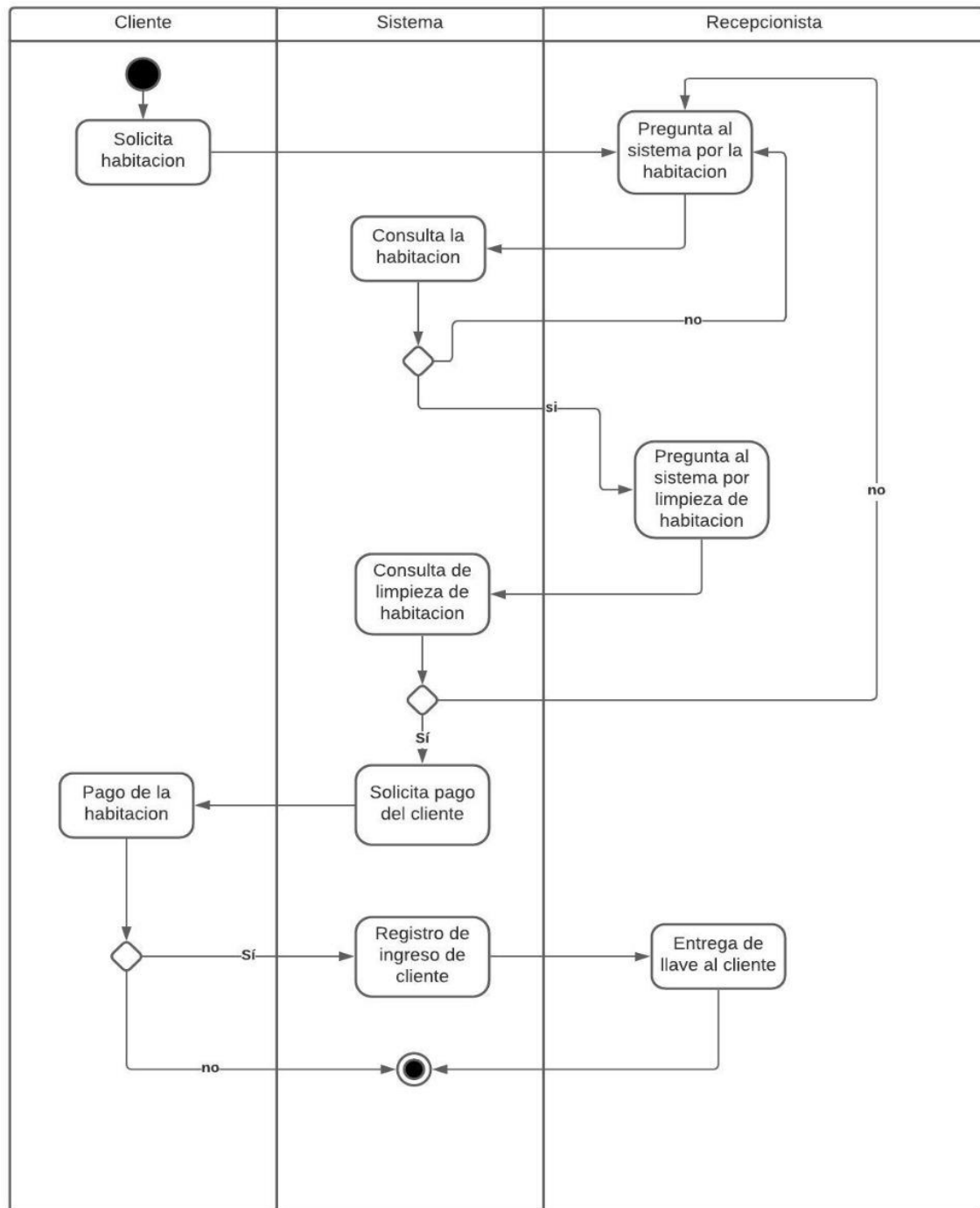
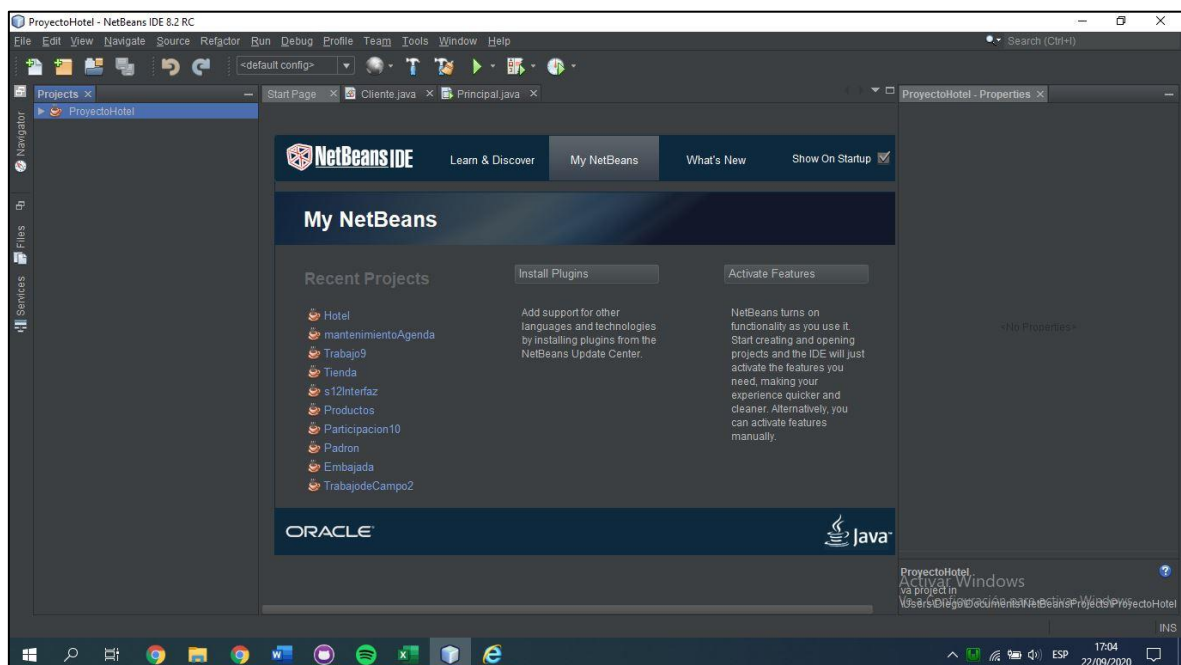
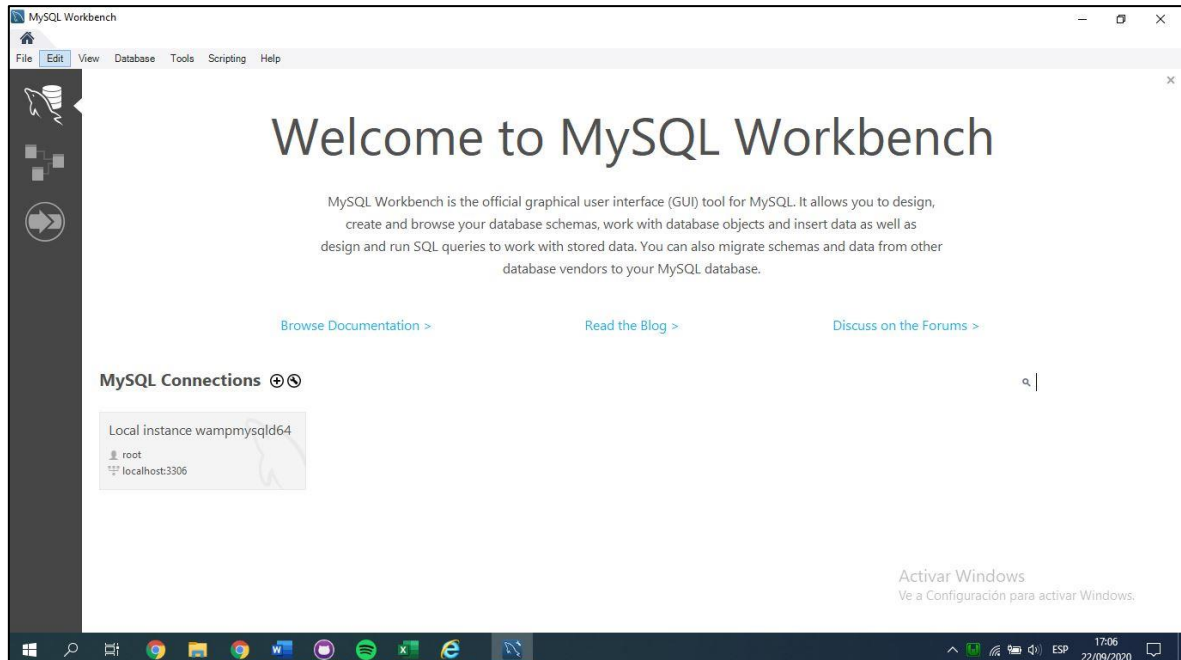


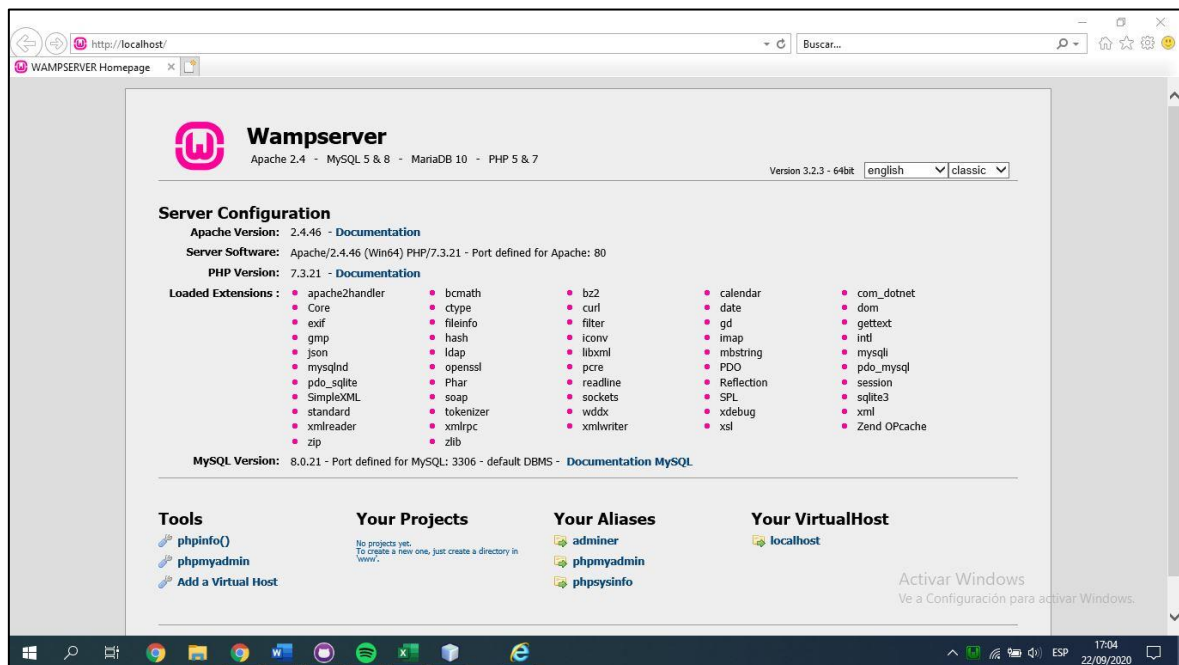
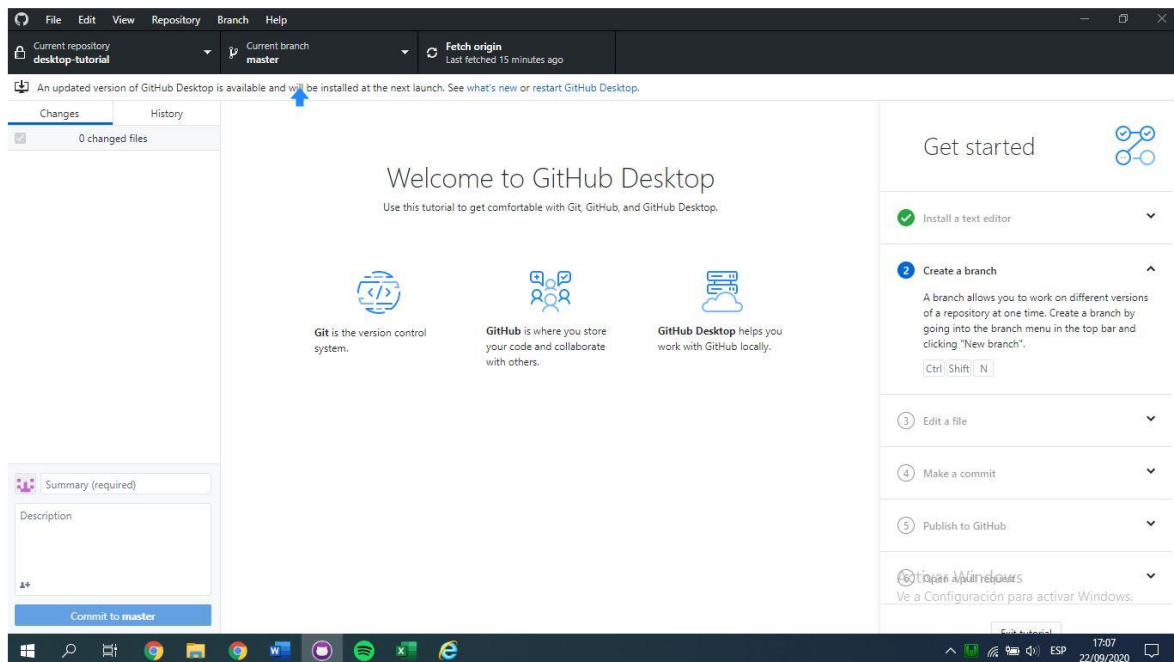
Ilustración 6: Diagrama de actividades con particiones o calles

Tabla 2: Trazabilidad de Requisitos

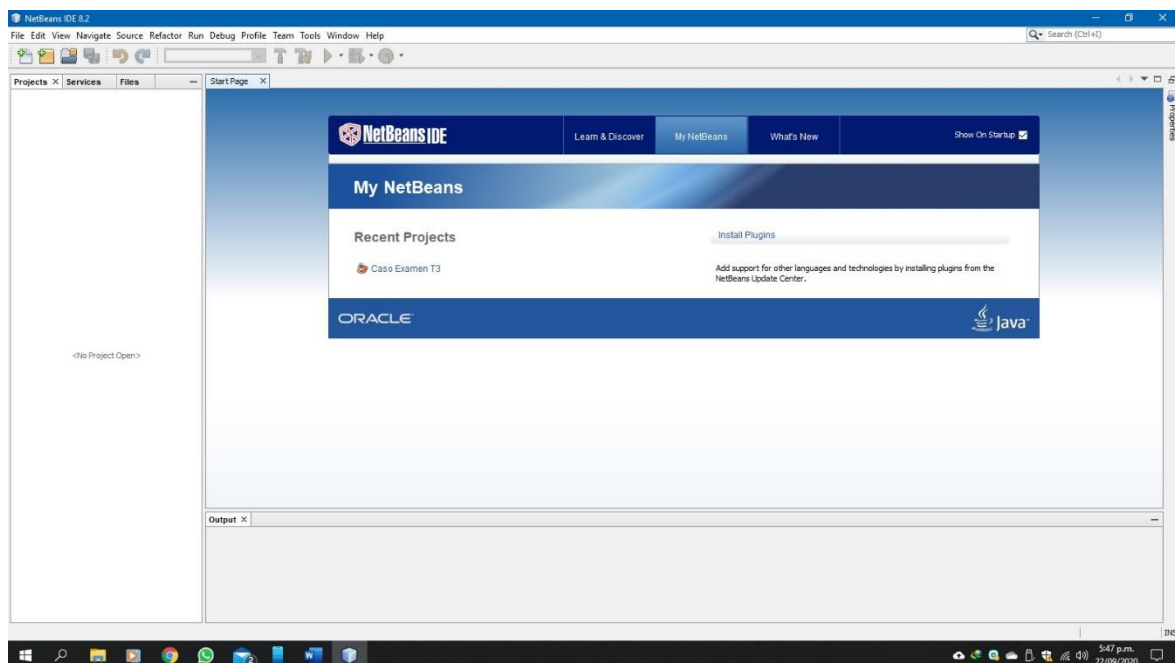
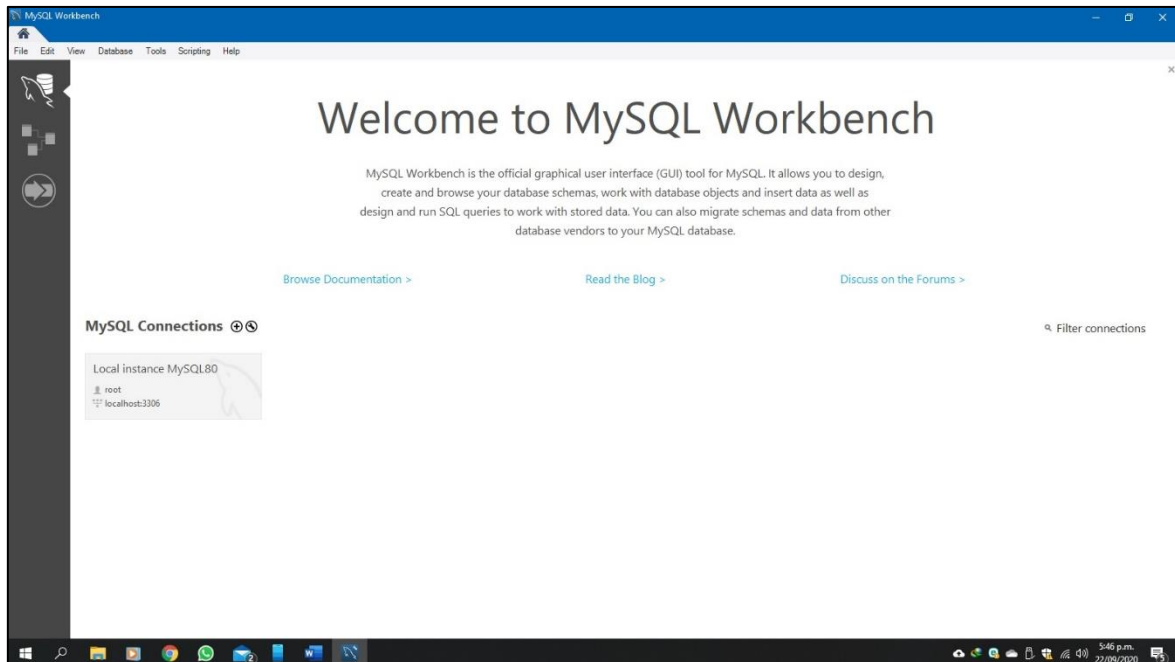
Tipo de Requisito	ID	Requisito	Casos de Uso	Prototipo	Versión	Instalado
Funcional	1	Verificar el usuario y acceder al sistema	Login al sistema	Si	1	/
Funcional	2	Permite el Registro de clientes	Registro de Clientes	Si	1	/
Funcional	3	Permite hacer un listado de los clientes existentes	Lista de Clientes	No	1	/
Funcional	4	Permite Modificar a los clientes registrados	Modificar al Cliente	No	1	/
Funcional	5	Permite Eliminar clientes registrados	Eliminar Cliente	No	1	/
Funcional	6	Saber el estado de las habitaciones	Consulta de habitaciones	Si	2	/
Funcional	7	Saber los clientes habituales	Consulta de clientes habituales	No	3	/
No funcional	8	NetBeans	/	No	/	Si
No funcional	9	MySQL o WampServer	/	No	/	Si
No funcional	10	MySQL WorkBench	/	No	/	Si

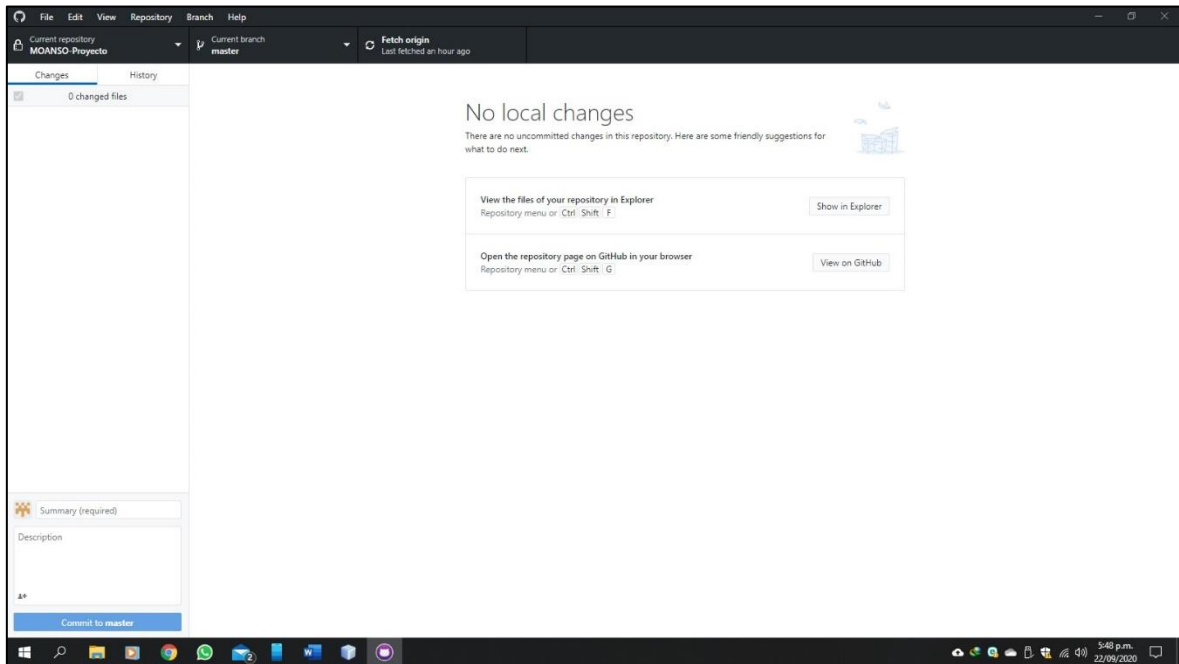
Computadora Diego Tasaico



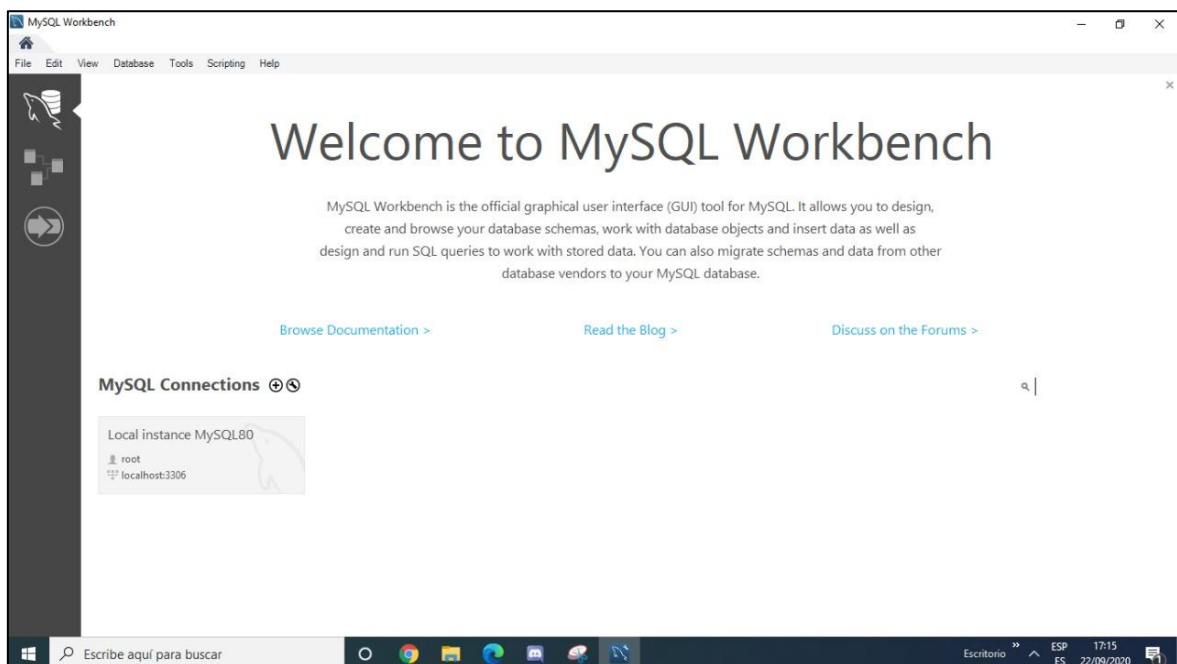


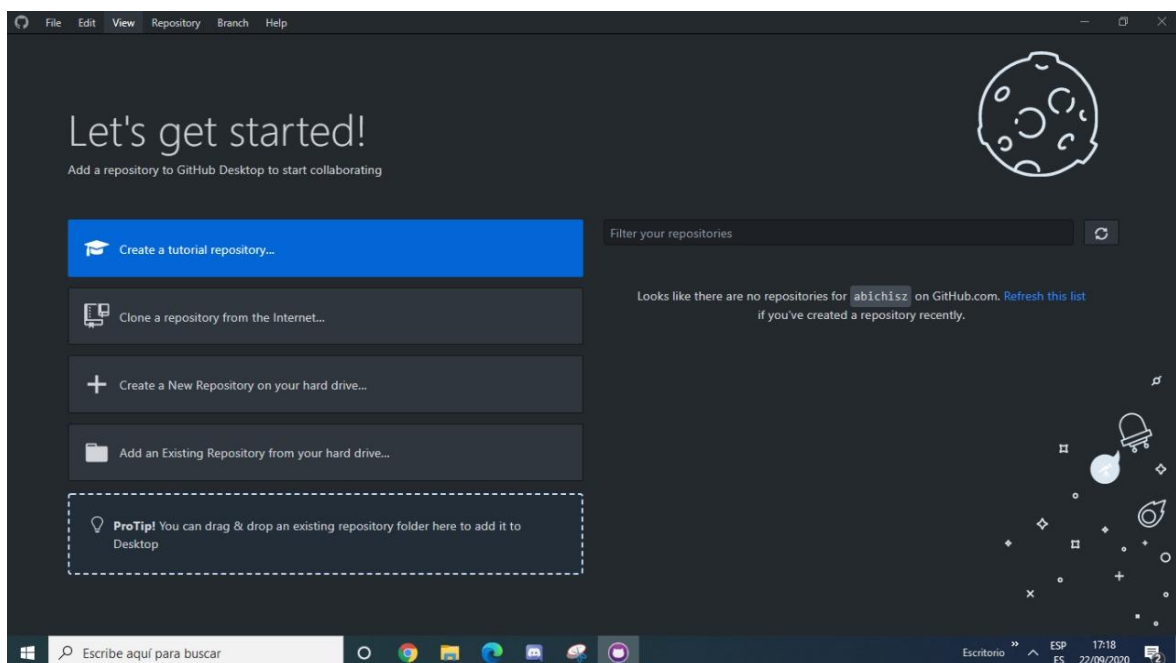
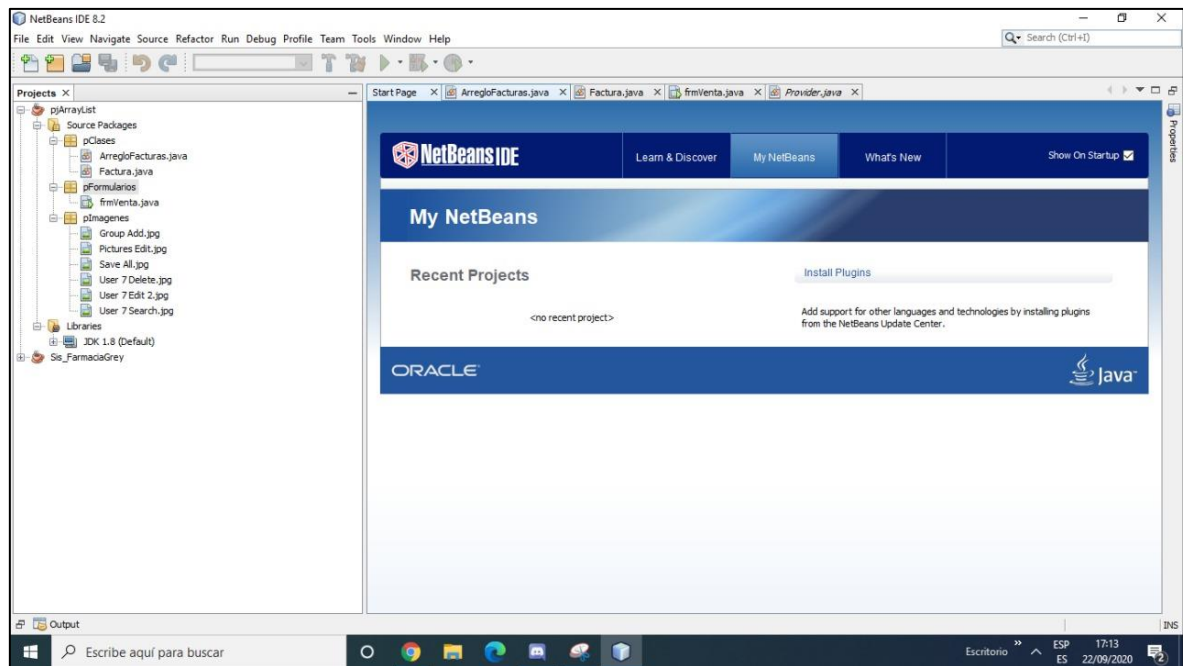
Computadora Mauricio Villanueva

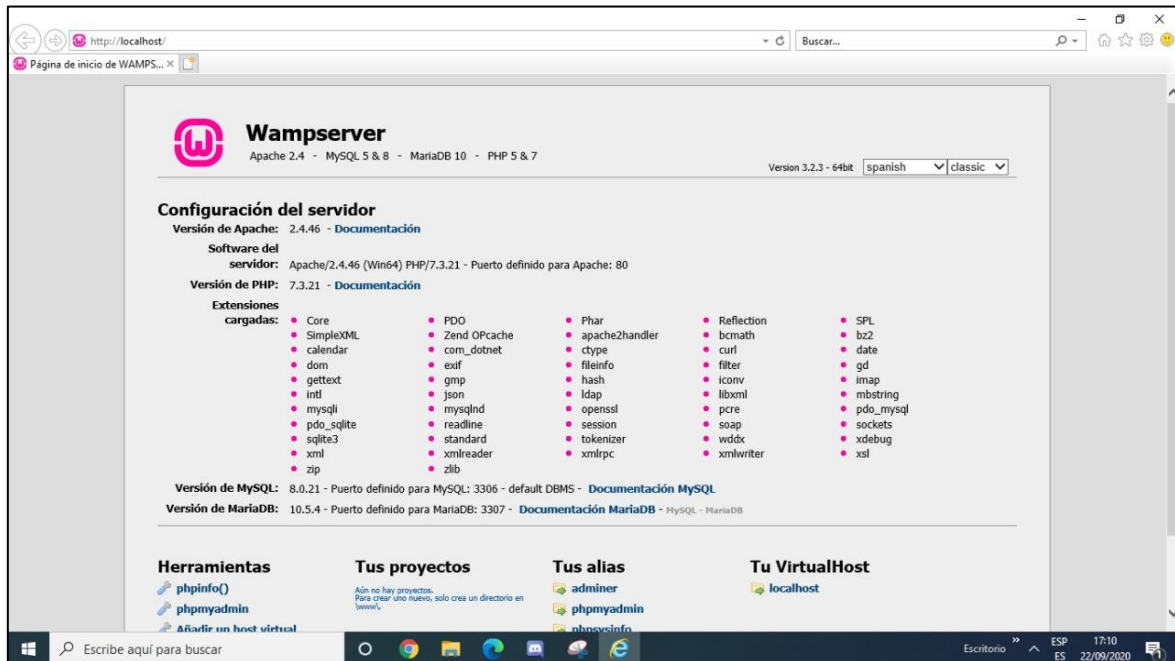




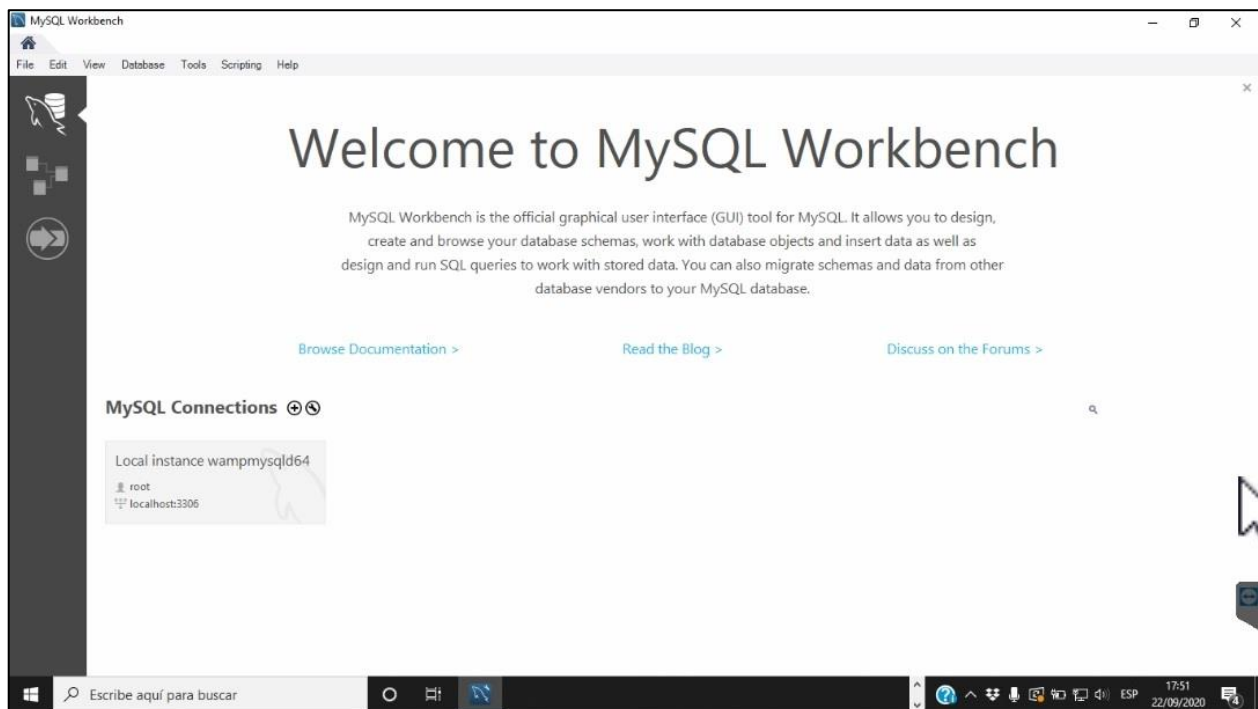
Computadora Abihail Zapata

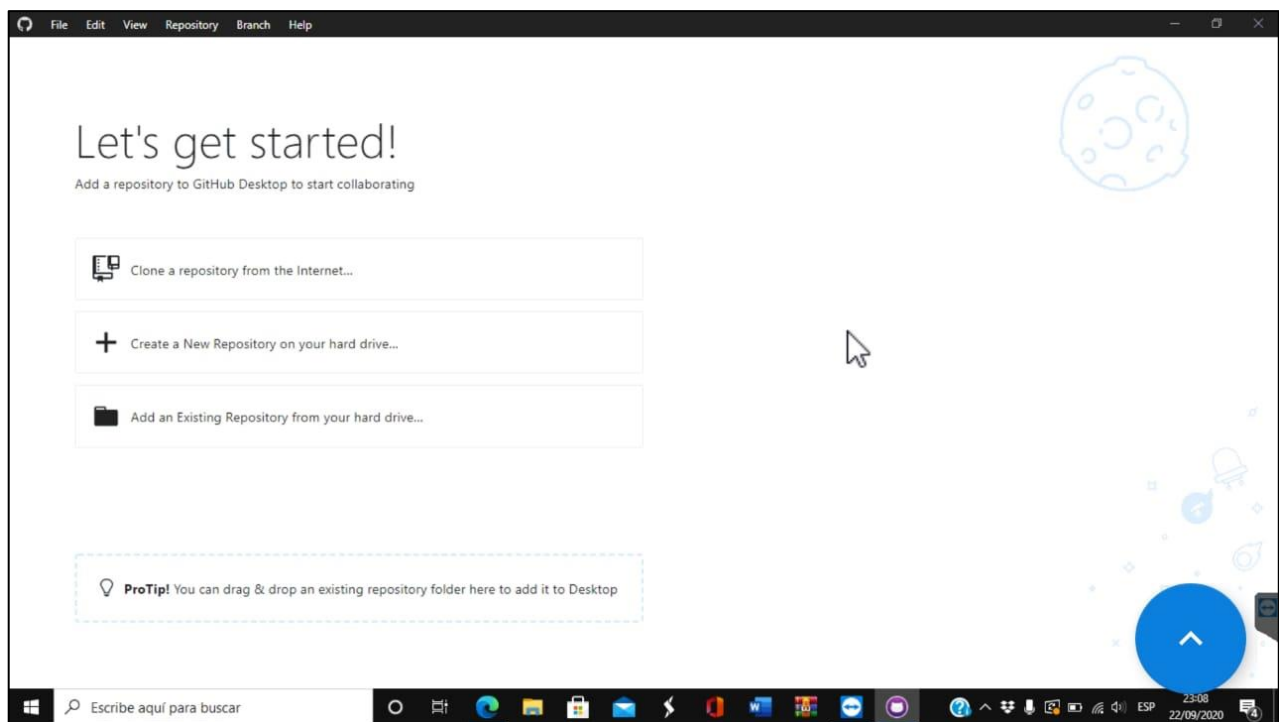
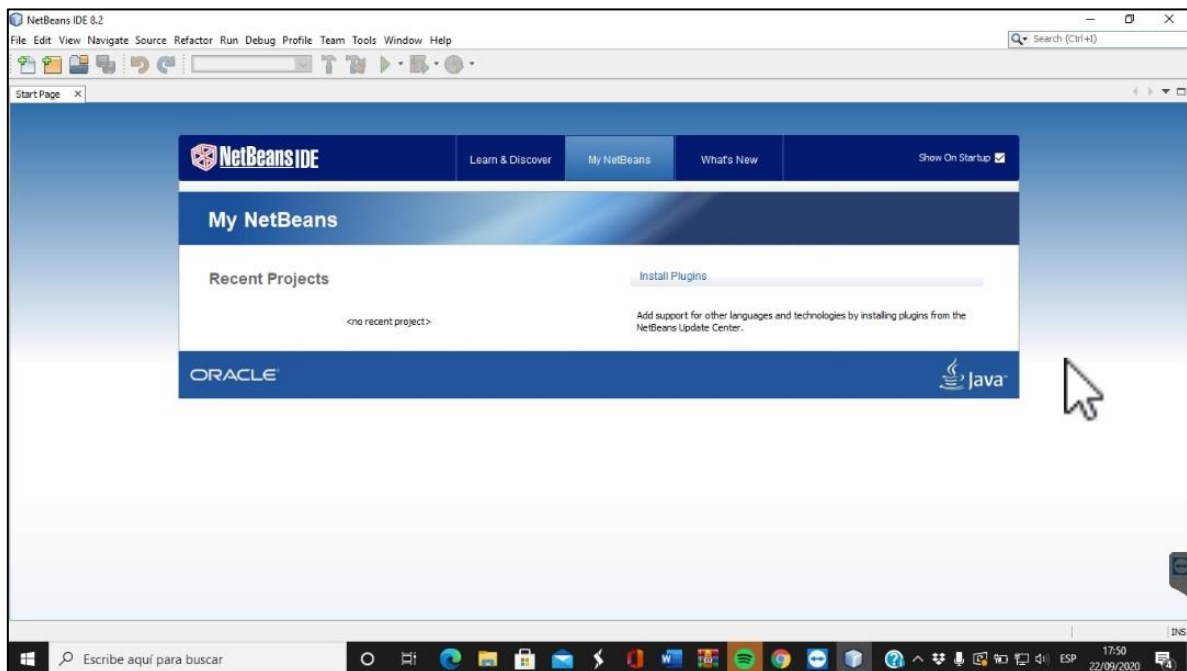


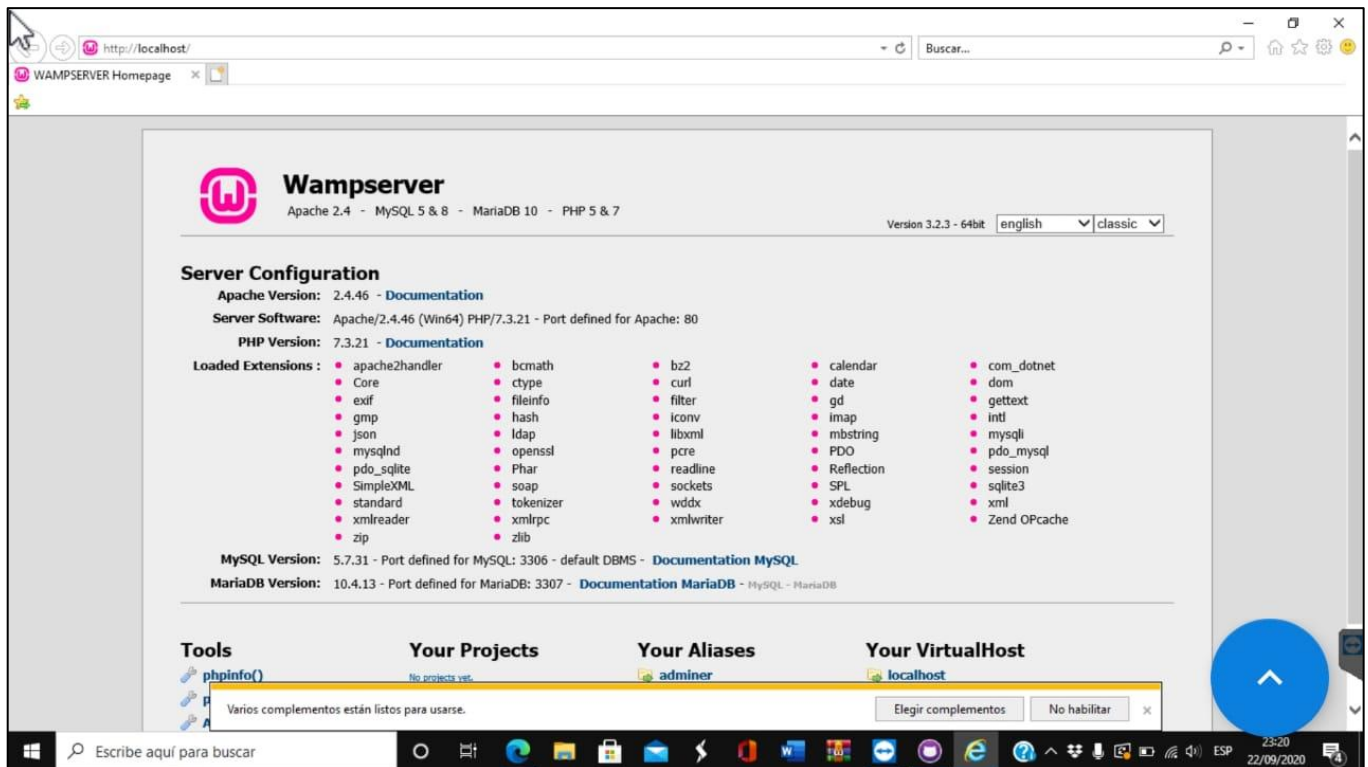




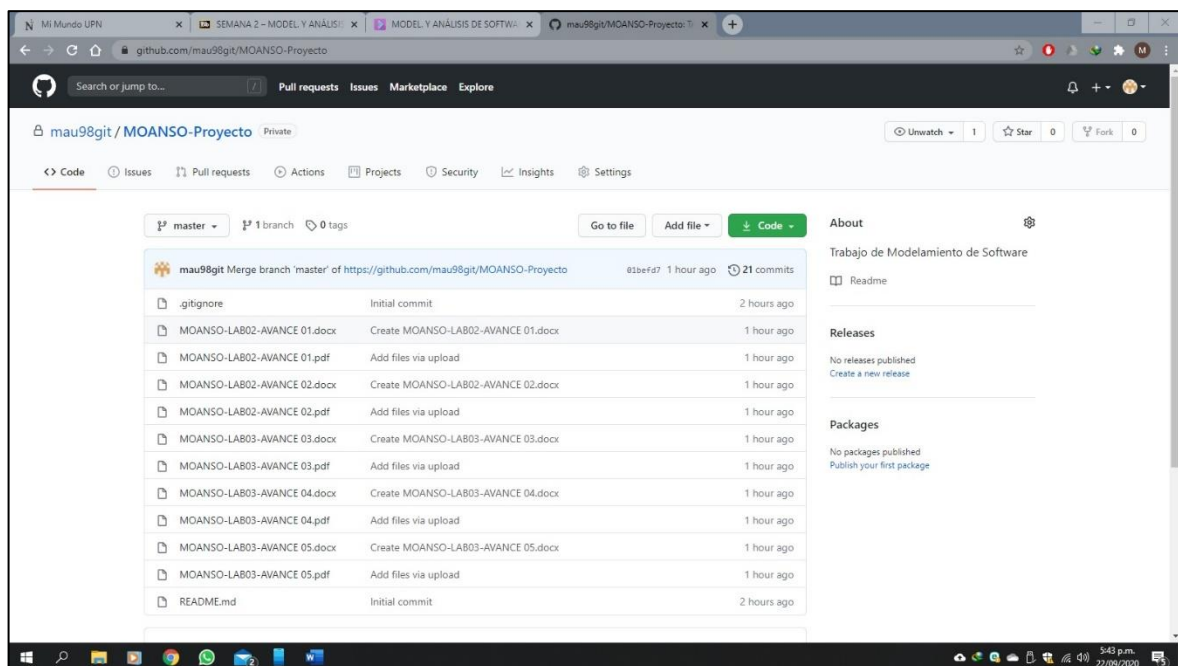
Computadora Olenka Lazo







Repositorio GitHub



Script de MySQL

```
-- MySQL Script generated by MySQL Workbench
-- Tue Sep 29 18:03:35 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_IN_DATE,NO_ZE
RO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-----

-- Schema Proyecto
-----

-----

-- Schema Proyecto
-----

CREATE SCHEMA IF NOT EXISTS `Proyecto` DEFAULT CHARACTER SET utf8 ;
USE `Proyecto` ;

-----

-- 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 NOT NULL,
  `estado` VARCHAR(25) NOT NULL,
  `tipo_habitacion` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`idHabitacion`))
```

ENGINE = InnoDB;

-- Table `Proyecto`.`Persona`

```
CREATE TABLE IF NOT EXISTS `Proyecto`.`Persona` (  
  `idPersona` INT NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(45) NOT NULL,  
  `apellido_paterno` VARCHAR(45) NOT NULL,  
  `apellido_materno` VARCHAR(45) NOT NULL,  
  `tipo_documento` VARCHAR(15) NOT NULL,  
  `num_documento` VARCHAR(15) NOT NULL,  
  `celular` VARCHAR(12) NOT NULL,  
  `email` VARCHAR(45) NULL,  
  PRIMARY KEY (`idPersona`),  
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,  
  UNIQUE INDEX `celular_UNIQUE` (`celular` ASC) VISIBLE)  
ENGINE = InnoDB;
```

-- Table `Proyecto`.`Trabajador`

```
CREATE TABLE IF NOT EXISTS `Proyecto`.`Trabajador` (  
  `idPersona` INT NOT NULL,  
  `sueldo` DECIMAL NOT NULL,  
  `acceso` VARCHAR(20) NOT NULL,  
  `login` VARCHAR(20) NOT NULL,  
  `password` VARCHAR(20) NOT NULL,  
  `estado` VARCHAR(1) NOT NULL,  
  PRIMARY KEY (`idPersona`),  
  UNIQUE INDEX `login_UNIQUE` (`login` ASC) VISIBLE,
```

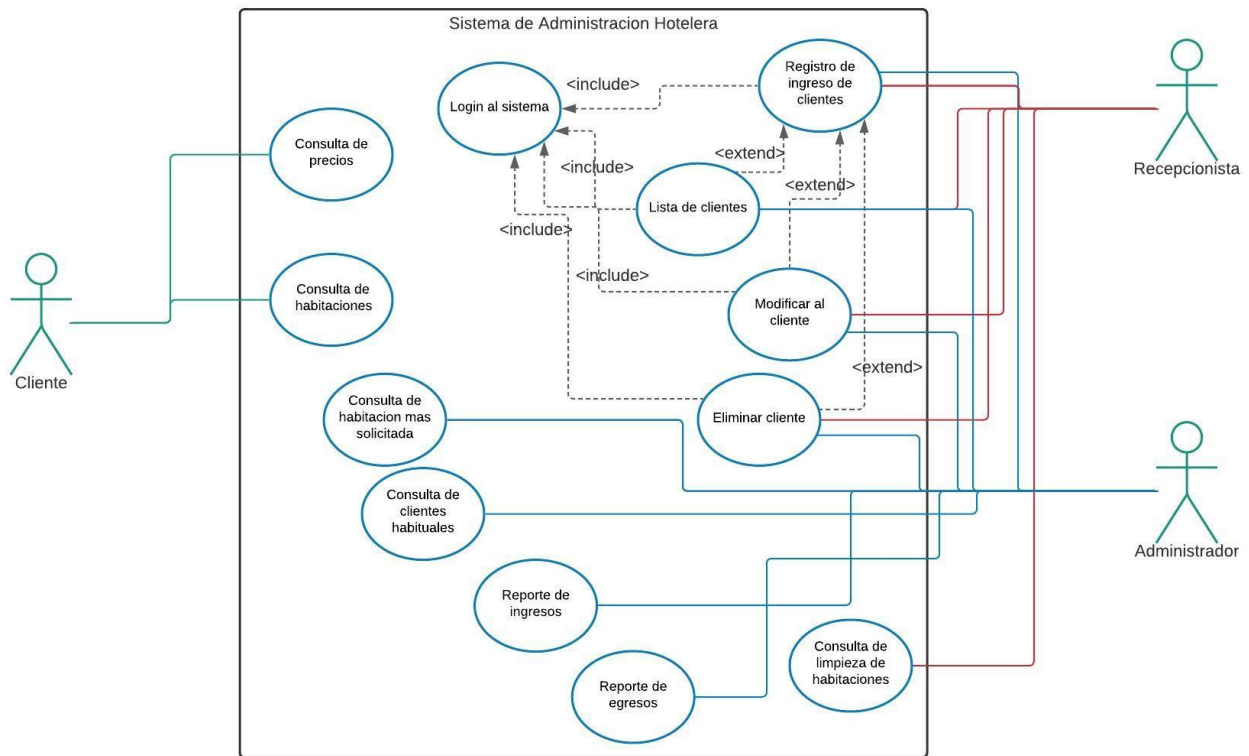
```
CONSTRAINT `fk_persona_trabajador`  
    FOREIGN KEY (`idPersona`)  
    REFERENCES `Proyecto`.`Persona` (`idPersona`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
  
-- Table `Proyecto`.`Cliente`  
  
-----  
  
CREATE TABLE IF NOT EXISTS `Proyecto`.`Cliente` (  
    `idPersona` INT NOT NULL,  
    `codigo_cliente` VARCHAR(10) NOT NULL,  
    PRIMARY KEY (`idPersona`),  
    UNIQUE INDEX `codigo_cliente_UNIQUE` (`codigo_cliente` ASC) VISIBLE,  
    CONSTRAINT `fk_persona_cliente`  
        FOREIGN KEY (`idPersona`)  
        REFERENCES `Proyecto`.`Persona` (`idPersona`)  
        ON DELETE NO ACTION  
        ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
  
-- Table `Proyecto`.`Reserva`  
  
-----  
  
CREATE TABLE IF NOT EXISTS `Proyecto`.`Reserva` (  
    `idReserva` INT NOT NULL AUTO_INCREMENT,  
    `idHabitacion` INT NOT NULL,  
    `idCliente` INT NOT NULL,  
    `idTrabajador` 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 NOT NULL,  
`estado` VARCHAR(15) NOT NULL,  
PRIMARY KEY (`idReserva`),  
UNIQUE INDEX `tipo_reserva_UNIQUE` (`tipo_reserva` ASC) VISIBLE,  
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_habitacion`  
  FOREIGN KEY (`idHabitacion`)  
    REFERENCES `Proyecto`.`Habitacion` (`idHabitacion`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_reserva_cliente`  
  FOREIGN KEY (`idCliente`)  
    REFERENCES `Proyecto`.`Cliente` (`idPersona`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_reserva_trabajador`  
  FOREIGN KEY (`idTrabajador`)  
    REFERENCES `Proyecto`.`Trabajador` (`idPersona`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
-----  
-- 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 NOT NULL,  
  `total_pago` DECIMAL 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`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Diagrama de Casos de Uso Relacionado



CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

12. Referencias Bibliográficas

- Alonso, F., Martínez, L., & Segovia, J. (2005). *Introducción a la ingeniería 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>
- Sommerville, I. (2011). *Ingeniería de software (9a. ed.)*. Pearson Educación.

13. Anexos

