

LISTA DE COTEJO REPORTE DE ACTIVIDAD (50%)

Unidad 1

Nombre asignatura: Desarrollo e Implementación de Sistemas de Información

Nombre del alumno: Alfonzo Velasco Quino

Nombre del docente: Dra. Verónica Guerrero Hernández

Criterios	Indicador máximo por criterio	Indicador de alcance total (50%)
1. Anexo se encuentra una portada	0-1	1
2. Explica el procedimiento de solución para llevar a cabo la actividad solicitada: <ul style="list-style-type: none"> • Conoce, identifica y analiza los temas correspondientes a la unidad para explicar el procedimiento utilizado para dar solución a lo solicitado. • Descripción satisfactoria al procedimiento de solución para llevar a cabo la actividad. 	0-45	45
3. Anexo de introducción	0-2	2
4. Manejo e inclusión de referencias bibliográficas	0-2	2
Total Indicador:	0-50	50



INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS

TUXTLA

DOCENTE: VERONICA GUERRERO HERNANDEZ

CARRERA: INGENIERIA EN INFORMATICA

ASIGNATURA: DSI

GRUPO: 610-A

TRABAJO: ACT3 MODELO DEL SISTEMA

ALUMNOS: ALFONSO VELASCO QUINO

CARLOS ALBERTO ALONSO VELASCO

ADRIAN SANTIAGO TORNADO MARTINEZ

FECHA: 12 DE MARZO DE 2023

DIAGRAMA DE CASO DE USO

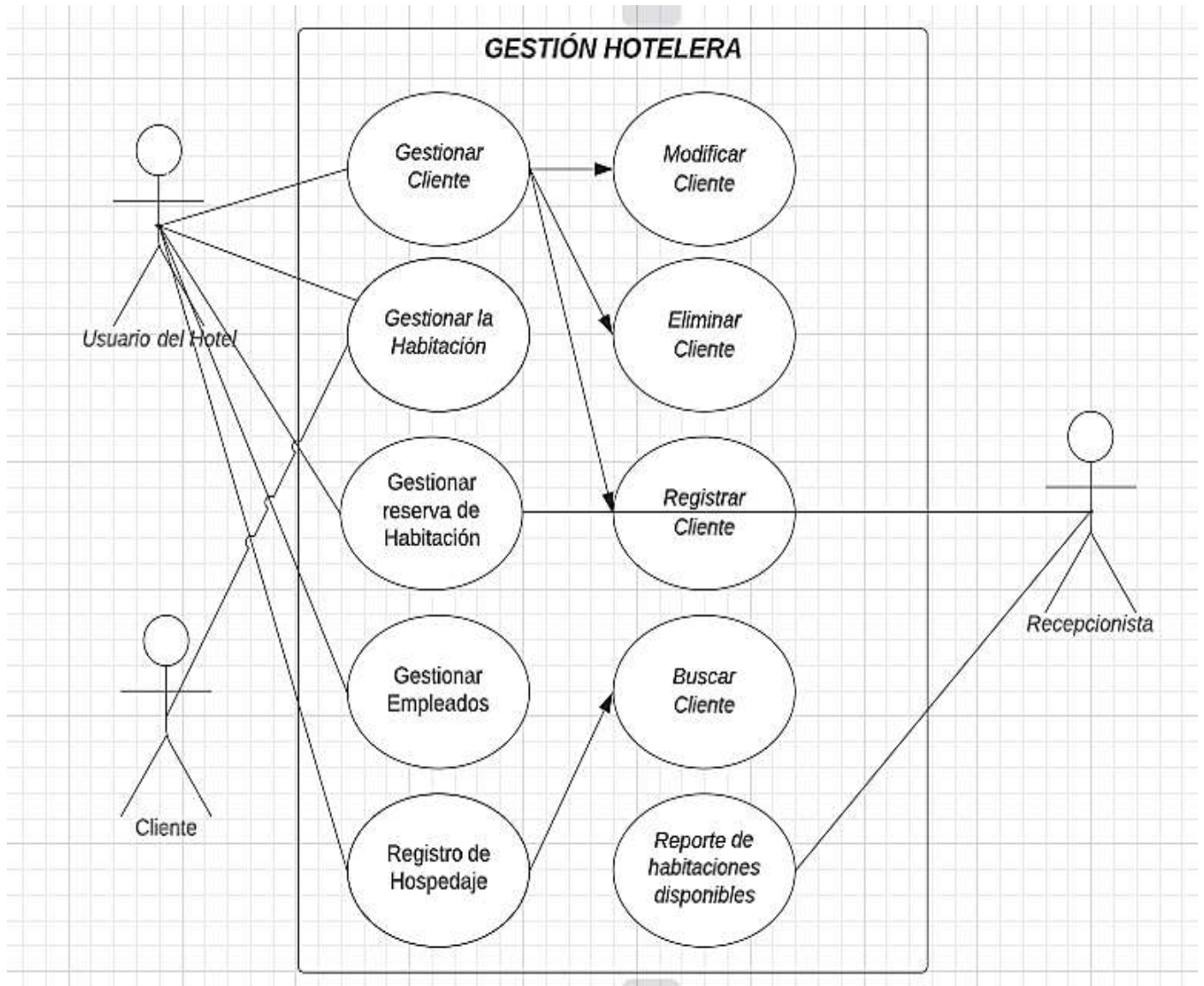


DIAGRAMA DE ACTIVIDAD

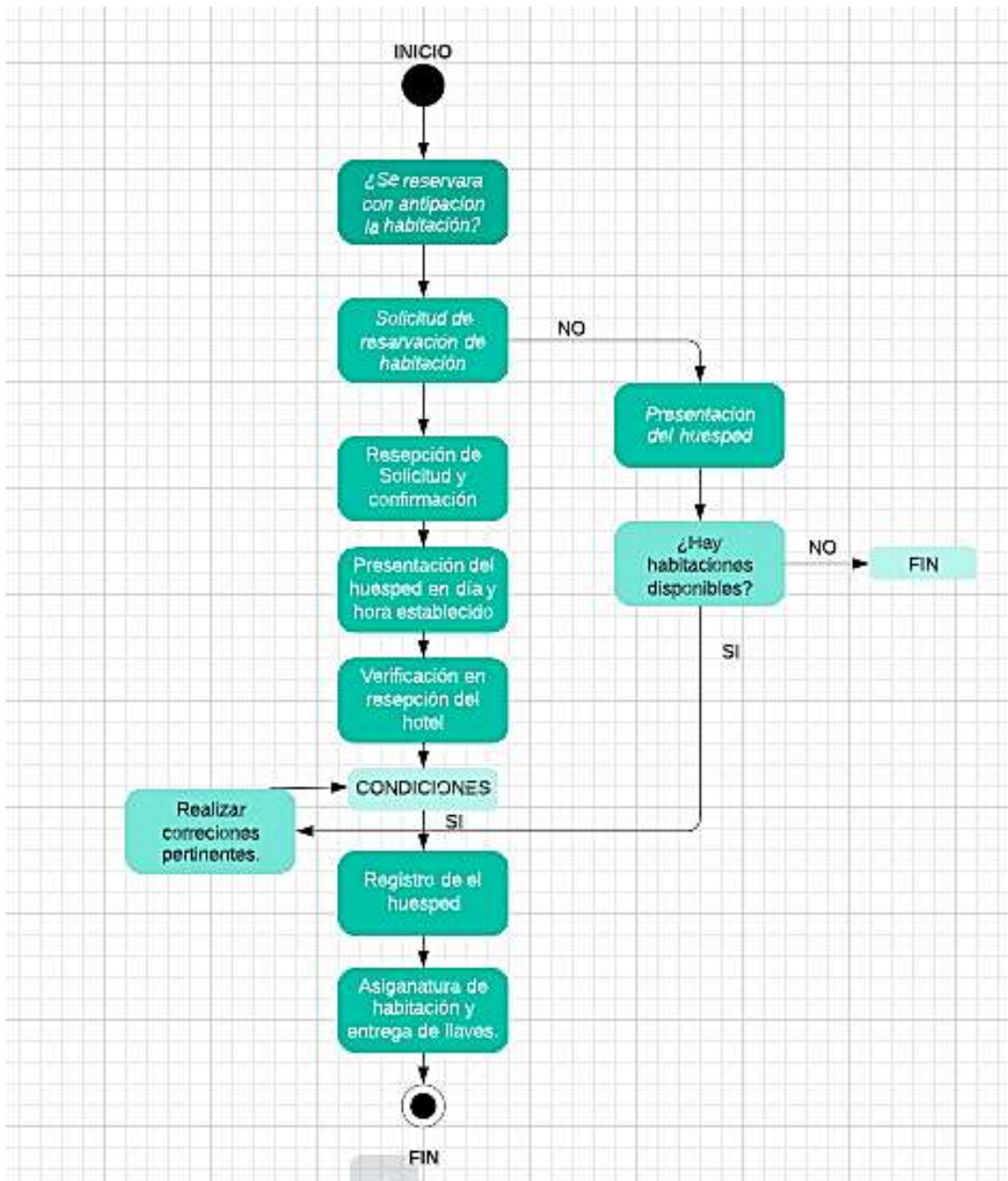


DIAGRAMA DE CLASES

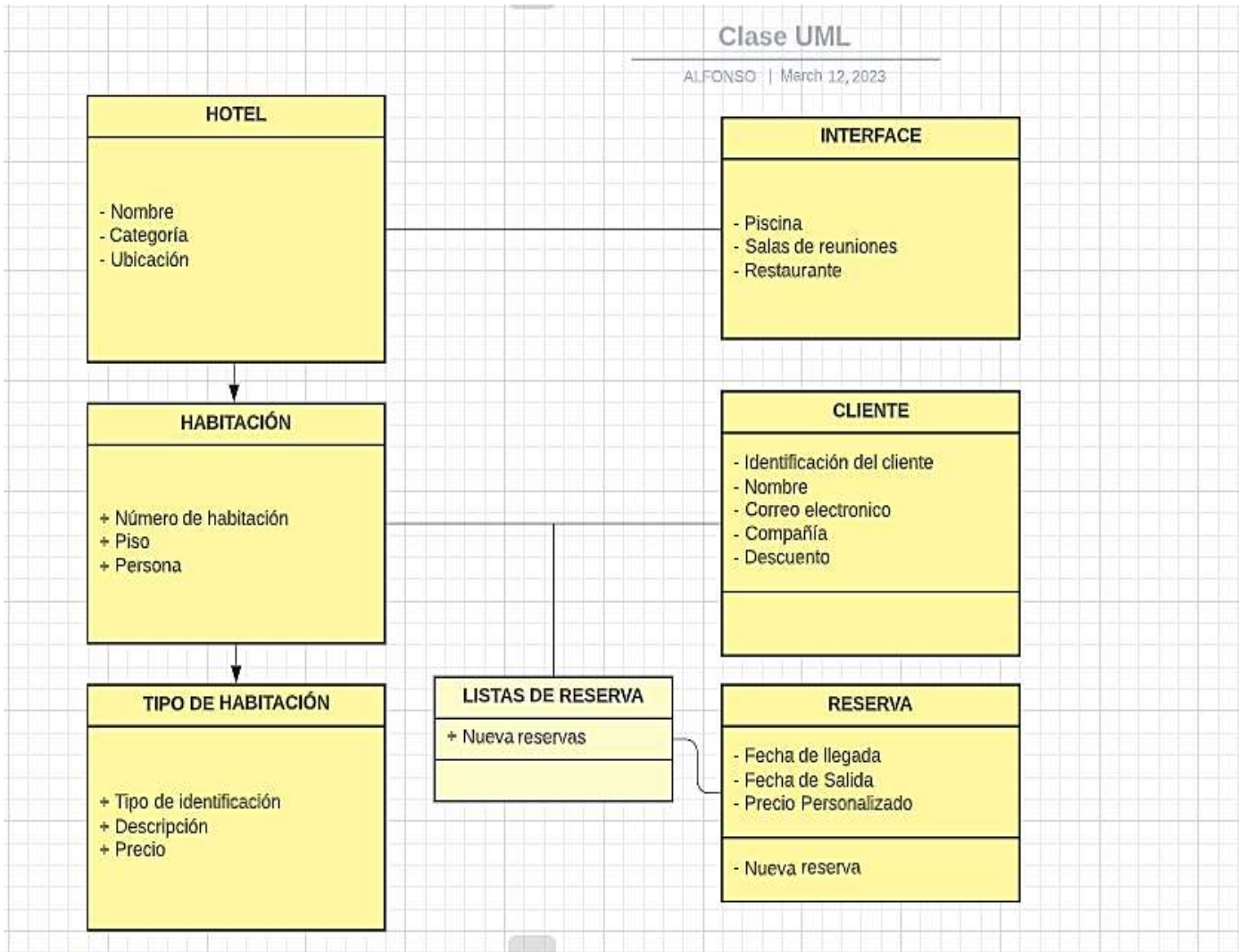


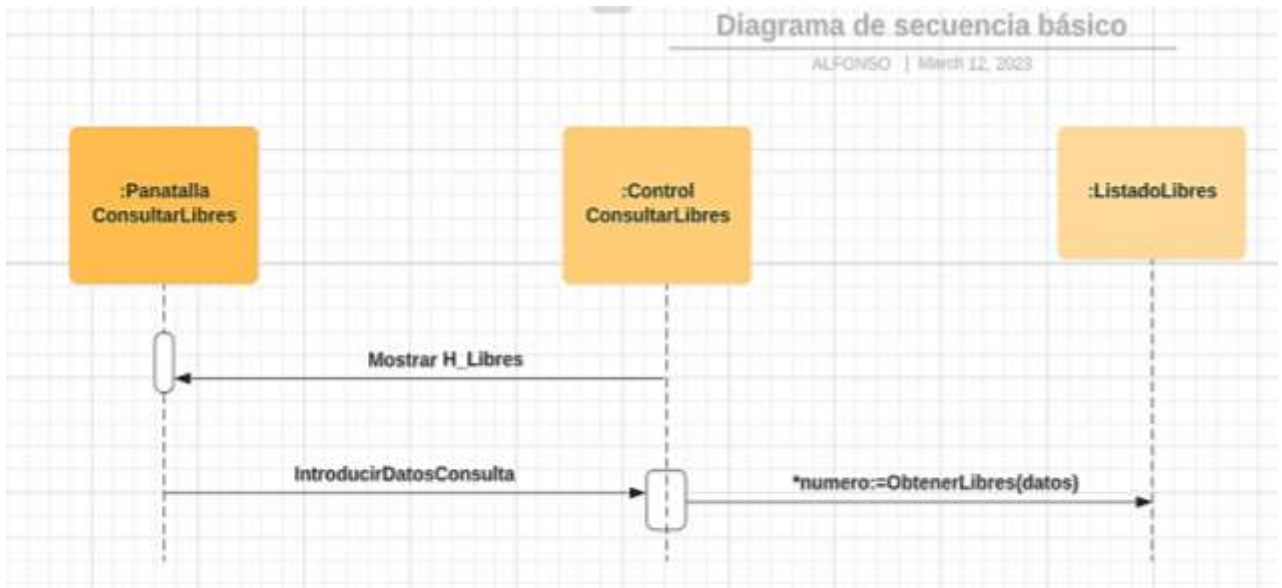
DIAGRAMA DE ESTADO

ALFONSO | March 12, 2023



DIAGRAMA DE SECUENCIA

CONSULTA DE HABITACION DISPONIBLE



Lista de cotejo INFORME TECNICO (30%)

Unidad 1

Nombre asignatura: Desarrollo e Implementación de Sistemas de Información

Nombre del alumno: Alfonzo Velasco Quino

Nombre del docente: Dra. Verónica Guerrero Hernández

Criterios	Indicador máximo por criterio	Indicador de alcance total (30%)
Adjunta una hoja de presentación y se menciona el título del proyecto	0-5	5
Se describe las generalidades de la propuesta de sistema de información.	0-5	5
Se describe el análisis de la propuesta de sistema de información	0-5	5
Se describe el modelado de la propuesta de sistema de información	0-5	5
Se describen el diseño de la propuesta de sistema de información	0-5	5
Manejo e inclusión de referencias bibliográficas	0-5	5
Total Indicador	0-30	30



**INSTITUTO TECNOLÓGICO SUPERIOR DE SAN
ANDRÉS
TUXTLA**

DOCENTE: VERONICA GUERRERO HERNANDEZ

CARRERA: INGENIERIA EN INFORMATICA

ASIGNATURA: DESARROLLO E IMPLEMTACION
DE SI

GRUPO: 610-A

TRABAJO: PROYECTO DESARROLLO DE UNA
APLICACIÓN WEB DE RESERVAS Y GESTIÓN
HOTELERA

INTEGRANTES: TORNADO MARTINEZ ADRIAN
SANTIAGO

ALFONSO VELASCO QUINO

CARLOS ALBERTO ALONSO VELASCO

RESUMEN

La popularización y auge del turismo han facilitado el surgimiento de nuevos modelos de alojamiento, tales como la vivienda de uso turístico, y de nuevas plataformas que entran en el mercado ofreciendo este y otros servicios al turista.

Dentro de este contexto, se desarrolla una aplicación web llamada WhatHostel; que permita al usuario de dicha aplicación buscar, filtrar y reservar en hoteles, tanto camas en habitaciones compartidas como habitaciones privadas.

INDICE

INTRODUCCIÓN	6
CAPÍTULO 1. GENERALIDADES	7
1.ANTECEDENTES	7
DESCRIPCIÓN DEL PROBLEMA	11
2.Planteamiento del Problema.....	11
3.Objetivo General.....	12
Objetivos Específicos	12
4.Justificación.....	12
CAPÍTULO 2. Análisis	13
Metodología.....	13
APLICACIÓN WEB. ESTRUCTURA Y FUNCIONAMIENTO	13
SERVICIOS WEB Y SERVIDOR	16
BASE DE DATOS.....	17
LENGUAJES DE PROGRAMACIÓN	18
CAPÍTULO 3.MODELADO	19
ANÁLISIS DEL SOFTWARE	19
REQUERIMIENTOS FUNCIONALES.....	19
REQUERIMIENTOS NO FUNCIONALES:	22
MODELO DEL SOFTWARE.....	24
MODELO DE NEGOCIO.....	24
MODELO DE CASO DE USO	29
Diagrama de estado	30
Diagrama de clases.....	31
DATOS DE LA APLICACIÓN	31
CAPITULO 4. Diseño.....	32
1.Diseño de la base de datos	32
Diagrama de entidad.....	32

2.Diseño Logico.....	33
3.Diseño Físico.....	34
4.Normaizacion	35
5.Programacion	35
6.Herramientas.....	43
CONCLUSIÓN.....	49
BIBLIOGRAFÍA.....	50

CONTENIDO DE TABLA

Tabla	
1.....	9
Tabla	
2.....	15
Tabla	
3.....	15
Tabla	
4.....	16
Tabla	
5.....	17
Tabla	
6.....	24
Tabla	
7.....	25

Tabla
8.....
.....26

Tabla
9.....
.....27

Tabla
10.....
.....28

INTRODUCCIÓN

Los portales de reservas existentes pertenecen a grandes empresas, enfocadas más en aspectos económicos que técnicos. Los cambios en la tecnología y estándares de diseño están siendo adoptados con lentitud por estas compañías, ofreciendo una oportunidad a nuevos portales para impresionar y atraer a los usuarios.

Una interfaz intuitiva, una navegación rápida, un fácil proceso de reserva; características que se pueden conseguir empleando las técnicas adecuadas y que ofrecen a WhatHostel una oportunidad.

CAPÍTULO 1. GENERALIDADES

1. ANTECEDENTES

La globalización y la reducción de precios en el transporte ha favorecido la creciente demanda turística en todo el mundo. Esta creciente demanda ha sido acompañada por un crecimiento de la oferta, los alojamientos dejan de ser ofrecidos únicamente por cadenas hoteleras y pequeños hostales para ser ofrecidos también por propietarios de viviendas. El precio se convierte en una parte clave de la competitividad y los albergues cobran un papel importante, ofreciendo camas individuales dentro de habitaciones compartidas.

Este modelo de alojamiento permite reducir los precios; además, el incentivo de tratar con otras personas y conocer gente de diferentes países ha propiciado también su popularización. Dentro de este mercado fructífero tiene también su sitio la tecnología. Desde antes del 2000 han ido apareciendo nuevos portales web para ofrecer alojamiento, actuando de intermediarios entre el turista y el anfitrión.

Estos portales han ganado poco a poco el terreno a las empresas físicas de turismo, dada su mayor capacidad de adaptación a los nuevos modelos; así como la expansión del comercio electrónico en general.

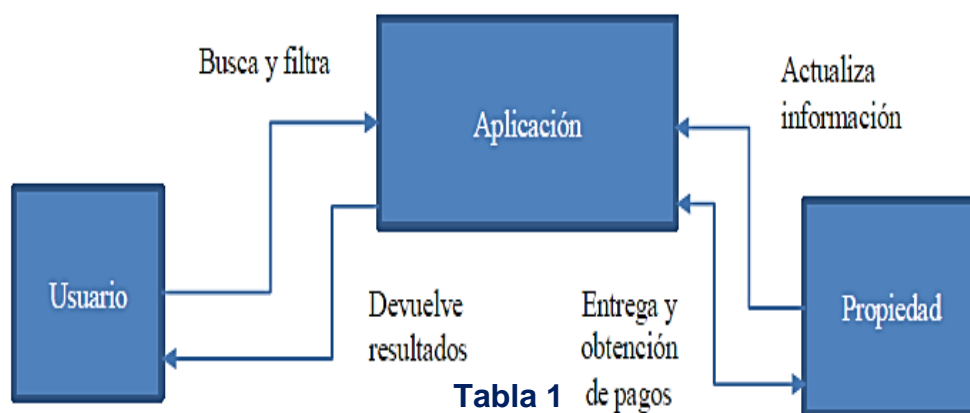
TIPO DE SOFTWARE QUE SE PROPONE: PORTAL DE RESERVAS

Un portal de reservas se caracteriza por tres actores principales: el usuario, la aplicación web y la propiedad. La aplicación web, o cualquier plataforma que permita la interacción entre el usuario y la propiedad, es el medio que emplea la empresa intermediaria para ofertar las habitaciones y camas de la propiedad al usuario.

La propiedad es dada de alta en la plataforma e introduce toda la información necesaria. La plataforma almacena esta información de forma persistente, por ejemplo, en una base de datos. Los precios y la disponibilidad son gestionados por la propiedad, quien indica un precio, del cual va a recibir el porcentaje acordado en el contrato de registro. El usuario accede a la aplicación desde la herramienta correspondiente; por un navegador en el caso de una aplicación web, o descargando una aplicación móvil desde el catálogo del dispositivo. A través de esta aplicación, el usuario puede realizar búsquedas indicando detalles como el lugar de destino, la fecha de llegada y de salida o el número de huéspedes; además, se pueden aplicar filtros sobre los servicios ofrecidos por la propiedad, localización, etc.

Los resultados que obtiene el usuario a partir de esta búsqueda son los que la aplicación proporciona a partir de los datos almacenados de las propiedades. Se muestran imágenes de la propiedad, horarios, dirección y el precio, entre otras cosas. El precio mostrado no tiene, ni suele, corresponder con el indicado por la propiedad. Éste viene dado por las correspondientes modificaciones que le aplique el propio portal, rebajando o subiendo el mismo.

Son varias las posibles formas mediante las cuales el portal y la propiedad reciben su parte correspondiente del precio mostrado. Una vez el usuario realice la reserva, este puede haber pagado una parte del precio en el momento de la reserva. Dicha parte pertenece al portal, es la comisión que cobra por el servicio ofrecido; el resto puede pagarse también en ese momento y lo recibe la propiedad, o ser pagado en el momento de llegada al alojamiento. Otra forma sería no pagar nada al realizar la reserva, pagando el precio íntegro a la llegada. En este caso la propiedad debe abonar a la empresa intermediaria la cantidad correspondiente en unos plazos acordados, normalmente, mensual.



FUNDAMENTO TEÓRICO

Modelo de negocio

Las Agencia de "Hotel Piedra Alta Classic" presentan tres modelos de negocio principales:

1. Modelo de negocio por publicidad. TripAdvisor.

En este modelo la empresa ofrece información o publicidad sobre una serie de hoteles y otras propiedades turísticas en función de las preferencias marcadas por el usuario. Un ejemplo de negocio que emplea este modelo como fuente de ingresos principal es TripAdvisor.

TripAdvisor muestra a sus usuarios los resultados de su búsqueda; en el caso de hoteles, muestra su información y valoraciones del mismo. Además, ofrece enlaces disponibles a otras páginas donde poder realizar la reserva. La empresa recibe una compensación económica por cada click que un usuario haga a un enlace, por parte de la agencia propietaria de dicho enlace. Además, puede obtener una compensación mayor en función de lo lejos que el cliente llegue en el proceso de reserva.

Silken Al-Andalus Palace Hotel

4.862 opiniones N.º 54 de 187 hoteles en Sevilla
Avenida de La Palmera s/n, Esquina C/ Parana, 41012 Sevilla, España 954 23 06 00 [Visitar sitio web del hotel](#)

25 personas están mirando este hotel

Llegada: --/--/-- Salida: --/--/--
Huéspedes: 1 habitación, 2 adultos, 0 niños

Asegura el precio más bajo de estos sitios web

Expedia 56 €


[Ver oferta](#)

Web oficial	56 €	Agoda.com	56 €
Travertia.es	56 €	Booking.com	56 €
lastminute.com	56 €	TripAdvisor	56 €

[Ver las 12 ofertas](#)

Los precios los proporcionan nuestros socios y en ellos se r...




Certificado de excelencia



[Ver todas las fotos \(2 723\)](#)

2. Modelo de negocio mercantil. Expedia.


En el caso de Expedia, sus ingresos provienen principalmente de las reservas que se realizan a través de su portal. Su modelo de acción original se basa en adquirir paquetes de habitaciones y vuelos a hoteles y compañías aéreas de todo el mundo, revendiendo dichos paquetes al usuario a través de su página. Además, ha adoptado el tercer modelo, el de agencia, debido a su mayor popularidad en Europa, Medio Oriente y Asia.

 Selecciona el hotel >
  Selecciona la habitación >
  Elige el vuelo

Para empezar, elige un hotel

¿Alguna pregunta? 912 71 81 76

Los precios de los viajes que se muestran incluyen vuelo + hotel, impuestos y tasas, pero no incluyen tasas de equipaje ni ninguna otra tasa que cobre directamente la línea aérea por el vuelo.




Ver mapa

Buscar por nombre del establecimiento

Filtrar establecimientos por


Nuestras sugerencias Precio Descuento del paquete Más ▾



Gran Meliá Palacio de los Duques-Leading Hotels of the World ★★★★★

Centro

912 71 81 76 • Tarifa de Expedia

 El precio incluye vuelos directos en clase económica/turista

4,8/5

¡Excepcional!
(267 comentarios)

¡Muy demandado!
Nos queda 1 por

~~1.249 €~~ **959 €**

por persona (incluye vuelo + hotel, tasas e impuestos)

3. Modelo de negocio como agencia. Booking.com.

El tercer modelo, representado por la agencia online perteneciente al grupo Booking Holdings, Booking.com; está basado también en la realización de reservas desde la propia página o aplicación. La diferencia respecto al modelo anterior es la no posesión de los productos que se venden. En el caso de Booking.com, no posee las habitaciones que ofrece, si no que cobra una comisión por la transacción.

La ventaja de este modelo es que no existe necesidad de preocuparse por la gestión del inventario, se basa en contratos con los dueños de este inventario (los hoteles) con comisiones que varían desde el 10% al 30%, en función de la posición que estos desean tener en el portal de reservas y de su tamaño.

Este es el modelo a adoptar por la aplicación web, *WhatHostel*; exige una menor necesidad de capital inicial, ya que no se necesita disponer de inventario, y se basa principalmente en crear una interfaz práctica y que genere confianza en el usuario, así como entablar relaciones contractuales con los diferentes *hostels* que van a formar parte del catálogo.

Booking.com

€ España Registra tu alojamiento Hazte una cuenta Inicia sesión

Alojamiento Vuelos Vuelo + Hotel Alquiler de coches Taxis al aeropuerto

Inicio > España > Andalucía > Sevilla > Resultados de la búsqueda

Buscar

Destino/Nombre del alojamiento:
Sevilla

Fecha de entrada
+ Fecha de entrada

Fecha de salida
+ Fecha de salida

2 adultos

Sin niños 1 habitación

Viajo por trabajo ?

Sevilla: 2.372 alojamientos encontrados
Las 3 razones para visitar este lugar: centro histórico, monumentos y comer tapas

Vista en mapa

Nuestros destacados Precio (más bajo primero) Estrellas Estrellas y precio Distancia desde el centro de la ciudad

Bestseller

Exe Sevilla Macarena ★★★★★

Macarena, Sevilla – Mostrar en el mapa (a 1,8 km del centro)

El Hotel Sevilla Macarena, situado frente a la basílica de la Macarena, da a la antigua muralla de Sevilla.

Reservado 7 veces en la última hora en nuestra página

Bien 7,9
11.053 comentarios

Mostrar precios

DESCRIPCIÓN DEL PROBLEMA

El proyecto pretende entrar en un sector ya poblado, ubicado en San Andrés Tuxtla. Dicho hotel lleva el nombre de “Hotel Piedra Alta Classic”, donde existen actualmente portales web y aplicaciones que facilitan servicios similares. Por tanto, el primer paso es analizar esta población, estudiar su modelo de negocio y las tecnologías que emplean, así como diferenciar a los distintos competidores en función de su posición en el mercado.

2.Planteamiento del Problema

Las causas principales que han intervenido en esta problemática son las siguientes: El administrador maneja información de forma manual, desperdicio de papel, vía telefónica inexistencia de un aplicativo lo que ha generado los siguientes efectos:

- No existe información de clientes.
- No existe recibo de pagos.
- Inexistencia de publicidad.
- No se conoce los servicios que oferta en el hotel.

- El contacto telefónico es informal y no genera compromisos.
- El contacto vía correo electrónico genera demoras en respuestas.

3.Objetivo General

Desarrollar una aplicación web y móvil mediante la metodología Scrum, para la gestión de reservas en el Hotel Piedra Alta Classic.

Objetivos Específicos

- Investigar distintas fuentes bibliográficas sobre temas relacionados acerca de las aplicaciones web, móviles y las reservas hoteleras para la construcción de la base teórica del aplicativo.
- Identificar requerimientos, mediante la metodología Scrum, para agilizar los procesos durante el desarrollo del sistema.
- Validar la aplicación a través de casos de pruebas realizadas por el usuario final para garantizar el cumplimiento de los requisitos del software.

4.Justificación

La actividad de hotelería, es una actividad mercantil de venta de servicios de alojamiento y gastronomía fundamentalmente, ésta tiene características generales y económicas financieras especiales que la diferencian de otras actividades comerciales e industriales.

Los portales de reservas existentes en muchos hoteles pertenecen a grandes empresas, enfocadas más en aspectos económicos que técnicos. Los cambios en la tecnología y estándares de diseño están siendo adoptados con lentitud por estas compañías, ofreciendo una oportunidad a nuevos portales para impresionar y atraer a los usuarios. Una interfaz intuitiva, una navegación rápida, un fácil proceso de reserva; características que se pueden conseguir empleando las técnicas adecuadas y que ofrecen a WhatHostel una oportunidad.

Este trabajo abarca la aplicación web; en concreto, desarrollar el portal de reservas. Para la creación del portal de reservas, denominado WhatHostel, se implementan los siguientes componentes:

- Servidor. El servidor de la aplicación, encargado de ofrecer servicios web al cliente y facilitarle recursos, además de conectarse y operar con la base de datos empleada.
- Cliente. Parte de la aplicación descargada por el usuario en su navegador. Hace uso del servidor para la obtención de información y recursos, y para realizar acciones sobre esta información.
- Base de datos. Contiene toda la información manejada por la aplicación, permitiendo su obtención y modificación por parte del servidor.

Además de estas tres partes principales, se hace uso de una plataforma como servicio para el despliegue de la aplicación. Esto consiste en un servicio de alojamiento y computación en la nube que externaliza la configuración del entorno del servidor.

CAPÍTULO 2. Análisis

Metodología

APLICACIÓN WEB. ESTRUCTURA Y FUNCIONAMIENTO

Se describe la estructura y funcionamiento de una aplicación, actor principal de un portal de reservas. En concreto, dado el alcance del proyecto, se trata el caso de una aplicación web, indicando los distintos componentes y las interacciones entre ellos.

- **Servicios web y servidor.**

El servidor es el encargado de realizar las tareas lógicas relevantes de la aplicación; autorización de recursos, autenticación, reservas, obtención y actualización de información de la base de datos.

Para ello, el servidor ofrece los servicios web a través de una API REST o una API SOAP. Los servicios web se basan en una comunicación cliente-servidor a través de la web usando HTTP/S. HTTP es un protocolo de aplicación de comunicación sin estado que permite las transferencias de información en la web. HTTPS es una modificación de HTTP, apoyada en protocolos de seguridad (SSL/TLS) para encriptar la comunicación.

REST es una representación de cambio de estado de los recursos basado en que Internet es una máquina de estados virtual donde el usuario avanza a través de una aplicación, atravesando estados, resultando en una página siguiente o nuevo estado de la aplicación. Este estilo de arquitectura de software presenta las siguientes restricciones:

- Cliente-Servidor: separación en dos componentes independientes.
- Sin estado: toda la información necesaria para comprender la petición está presente en cada mensaje HTTP. El cliente puede mantener una sesión a través de *cookies* o *tokens*.
- Orientación a recursos: utiliza un identificador para acceder a los recursos, URI. Asocia los comandos HTTP con las funciones CRUD: POST-Crear, GET-Leer, PUT-Actualizar, DELETE-Borrar.

SOAP es un protocolo estándar de transporte basado en XML que permite el intercambio de mensajes a través de Internet, mediante el protocolo HTTP. Está orientado a métodos, en lugar de a recursos como REST. Una petición SOAP tiene 3 partes principales:

- Envoltorio: marco que define el contenido del mensaje y su procesamiento.
- Reglas de codificación: indican como expresar las instancias de los tipos de datos definidos en la aplicación.
- Convención: usado para representar las llamadas a procedimientos remotos y sus respuestas.

El servidor como aplicación software, debe estar alojado en una máquina con conexión a Internet para poder recibir y procesar las peticiones de los clientes. Entre las opciones disponibles, podemos desplegarlo en una maquina propia, que debe ser configurada para el correcto funcionamiento del servidor; o emplear los servicios de *cloud computing* que ofrecen algunas empresas. Estos servicios de *cloud computing* o computación en la nube ofrecen alternativas de alojamiento en

plataformas e infraestructuras, dependiendo del tipo de servicio (PaaS, IaaS), en máquinas externas ya configuradas.

- **Cliente.**

El cliente es la parte de la aplicación que se ejecuta en la máquina del usuario, en concreto, en un navegador web. El cliente se encarga de utilizar la API que ofrece el servidor para obtener información y mostrársela al usuario. Además, el cliente atiende las interacciones del usuario con la página y realiza acciones en consecuencia, tales como enviar formularios al servidor para obtener un resultado o mostrar nueva información en la página.

- **Base de datos.**

En la mayoría de aplicaciones web es necesario almacenar información de manera persistente. Una de las opciones más utilizadas es el uso de una base de datos. Los sistemas gestores de bases de datos son herramientas que permiten la creación y gestión de una base de datos. Existen varios modelos de bases de datos dependiendo de su función y administración de los datos.

- Jerárquicas: los datos se organizan en forma de árbol invertido, el nodo raíz es el nodo padre y puede tener hijos, un hijo sin hijos se conoce como hoja.
- De red: se diferencia del anterior en que un mismo nodo puede tener varios nodos padres.
- Orientada a objetos: se organiza en torno a objetos enfocándose en la integración con lenguajes de programación orientados a objetos como Java, C#, C++, etc.
- Relacionales: basada en relaciones concebidas como tablas compuestas por registros, que representan las tuplas y campos (información) de la base de datos. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL.

- Documentales: almacenan los datos en forma de documentos. Cada documento representa información y su relación con otros elementos de datos, en forma de clave-valor.
- Gráfica: utiliza la teoría de grafos para almacenar, mapear y consultar relaciones. Se basa en colecciones de nodos y bordes interconectados.

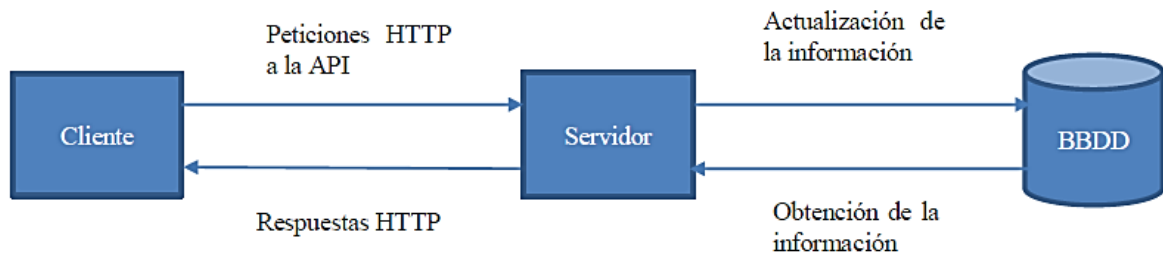


Tabla 2

SERVICIOS WEB Y SERVIDOR

En primer lugar, se selecciona el tipo de servicio web o API a utilizar: SOAP o REST. A continuación, se barajan las alternativas para el servidor, tratando los lenguajes de programación disponibles y, en el caso correspondiente, el servidor web o entorno de ejecución complementario.

Tabla 3

Tipo API	Formato Mensaje	Enfoque	Caché peticiones	Rendimiento
SOAP	XML	Basado en funciones	No	Más ancho de banda y recursos de computación
REST	Texto plano, JSON, HTML, XML, YAML, etc	Basado en datos	Sí	Menos recursos

flexibilidad, la prioridad es la obtención de datos con una alta velocidad de respuesta. Por otro lado, se tiene una base de conocimientos personal previa mayor de JSON y HTML que de XML. Dado que REST permite que las peticiones se almacenen en caché y que SOAP consume un mayor ancho de banda; el primero resulta más favorable respecto a la escalabilidad y velocidad de respuesta. Además, permite el uso de JSON y HTML. Se elige REST como API para los servicios web de la aplicación.

En cuanto a infraestructura necesaria para almacenar la parte del servidor de la aplicación web, para la atención de peticiones y su procesamiento; se comparan las distintas alternativas prestando especial atención a los costes, tiempo de desarrollo y escalabilidad.

Según los costes, se pueden diferenciar dos tipos: CAPEX y OPEX:

- Inversiones en bienes de capital (CAPEX). Inversión para la compra de un activo del que se espera que otorgue beneficios a lo largo de su vida útil.
- Coste de funcionamiento (OPEX). Gasto recurrente o continuo, asociado generalmente a los costes de mantenimiento.

A continuación, se exponen las características de las dos alternativas consideradas: servidor propio y *cloud*.

Tabla 4

Infraestructura	Costes CAPEX	Costes OPEX	Tiempo	Escalabilidad
Servidor propio	Servidores físicos Router, switches, cables, etc. Espacio físico ocupado Seguridad, extinción	Electricidad Conectividad Mantenimiento <i>hardware</i> Monitorización	Configuración de un entorno capaz de ejecutar la aplicación web, así como de tener una red segura y eficiente.	Se deben ampliar y configurar los nuevos recursos necesarios para atender la demanda
<i>Cloud Computing</i>	Ninguno	Recursos consumidos Servicios opcionales	Aprendizaje de la plataforma que ofrece el servicio <i>cloud</i>	Automático (asumiendo un posible aumento de los costes)

BASE DE DATOS

Las dos alternativas a considerar para la creación de la base de datos son MySQL y MongoDB. MySQL es un sistema de gestión de bases de datos relacional que permite el almacenamiento de datos estructurados. Ofrece una gran consistencia y la posibilidad de realizar transacciones. Esto lo hace menos flexible en un sistema con esquemas de datos no necesariamente estructurados y que pueden cambiar con frecuencia. MongoDB es una base de datos no relacional documental, con una estructura de datos similar a JSON (BSON), lo cual permite una integración de los

datos más rápida que en los modelos relacionales, pero pierde en cuanto a propiedades ACID.

LENGUAJES DE PROGRAMACIÓN

Una vez seleccionado el tipo de API y la infraestructura que almacenará nuestra aplicación web, comparamos los lenguajes de programación más utilizados para el desarrollo del lado del servidor y del cliente.

Lenguaje de programación en el lado del servidor

Dado que Heroku proporciona un servicio ya configurado, conocido como *Platform as a Service (PaaS)*, según la tecnología empleada para el desarrollo del servidor, se comparan los lenguajes de programación del lado del servidor en función de su rendimiento y escalabilidad, además de la documentación existente y la comunidad.

Lenguaje de programación	Año de lanzamiento	Comunidad	Sistema o <i>stack</i> usual	Rendimiento y escalabilidad
PHP	1995	Preguntas en StackOverflow: 1.112.866 Estrellas en GitHub: 12.670 Contribuidores: 485	LAMP: Linux Apache MySQL PHP	Mayor velocidad computacional
NodeJS (JavaScript)	2009	Preguntas en StackOverflow: 187.917 Estrellas en GitHub: 38.398 Contribuidores: 1488	MEAN: MongoDB ExpressJS Angular NodeJS	Mayor escalabilidad: capaz de atender más peticiones simultáneamente con menos recursos

Tabla 5

Lenguaje de programación en el lado del cliente

La parte del cliente de la aplicación web, corresponde a las páginas web y recursos que el usuario obtiene a través del navegador. Se emplean los siguientes lenguajes para la creación de estas páginas:

- 1) HTML.

HyperText Markup Language es un lenguaje de etiquetas empleado para la estructuración de páginas web, permitiendo definir los distintos componentes de manera estática.

2) CSS.

Cascading Style Sheets es un lenguaje de diseño gráfico para establecer el diseño visual de un documento estructurado con un lenguaje de etiquetas como HTML. Permite aplicar estilos a las páginas web.

3) JavaScript.

JavaScript es un lenguaje de programación interpretado, original del estándar ECMAScript. Es débilmente tipado y orientado a objetos. Su uso principal es el desarrollo de páginas web dinámicas, insertando lógica en el lado del cliente.

Existen *frameworks* que cambian el paradigma de desarrollo del lado del cliente de una aplicación web, ofreciendo patrones de software con los que han sido diseñados. Uno de estos *frameworks* es Angular.

CAPÍTULO 3.MODELADO

ANÁLISIS DEL SOFTWARE

REQUERIMIENTOS FUNCIONALES

Proceso de Negocio	Actividad del Negocio	Responsable del Negocio	Requisito Funcional
			RF-01 Debe administrar las reservaciones de habitaciones o lo que es lo mismo será capaz de manejar de manera fluida la reservación de alguna habitación, esto clasificara entre las habitaciones que

Generales	Registro de datos	Administrador de datos		están disponibles y las que no al mismo tiempo que los datos del cliente.
			RF-02	Debe imprimir una lista con los clientes que tenga una reservación realizada.
			RF-03	Aplicación capaz de funcionar en un entorno web, esto porque se podrá reservar una habitación por medio de internet para cualquier cliente que desee una habitación.
Reservación	Registro del cliente	Recepcionista	RF-04	Reservar habitaciones que estén disponibles. La reserva se puede realizar a través de la aplicación web. El cliente puede observar las habitaciones que están disponibles para así poder hacer su reservación como le convenga.
			RF-05	Cancelar una reservación estará disponible para el cliente para el momento en que así lo decida. La reservación tiene una cancelación automática si el cliente después de un tiempo no se reporta al hotel.
			RF-06	Reportes del registro diario del uso de las habitaciones.
			RF-07	Registro tipo de cliente, el cliente se registrará si es por empresa o es turista.
			RF-08	Se debe registrar los datos personales del cliente, esta información debe llevar los siguientes datos, nombre, apellidos, edad.

			<p>RF-09 Identificar clientes que reciben preferencia, estos clientes son referenciados según la edad, donde la edad sea menor de 9 años.</p> <p>RF-10 Se tendrá otra interfaz donde la recepcionista puede observar las habitaciones que se reservaron por medio de la web, esta interfaz mostrara la información del cliente registrado en la base d datos y comunicara en tal las reservaciones.</p>
Habitaciones	Información de Habitaciones	Personal informativo	<p>RF-11 Mostrar una lista con las habitaciones y sus tipos dobles o sencilla cada una con una descripción detallada con lo que cuenta (precio, capacidad, etc.).</p> <p>RF-12 Debe contar con una base de datos de habitaciones.</p> <p>RF-13 Debe contar con fecha de inicio y de fin de la reservación de la habitación.</p> <p>RF-14 Debe mostrar disponibilidad de habitaciones.</p>
Web	Acceso web	Página web	<p>RF-15 Para reservar una habitación necesita acceso de internet por parte del usuario o el cliente.</p>
			<p>RF-16 El cliente puede darse de alta en cualquier momento una vez hecha la reservación luego de esto la</p>

Clientes	Reservación	Recepcionista		habitación que tenía reservada quedará disponible.
			RF-17	El cliente no podrá reservar una habitación que ya está reservada.
			RF-18	El cliente sino se reporta después de la hora establecida la reservación la misma será cancela y puesta a disposición para otro cliente.
			RF-19	Calcular número de habitaciones que estén disponibles en ese momento y las no disponibles.
			RF-20	Guardar datos de usuarios registrados por cada habitación para ubicar las habitaciones disponibles, es porque el usuario escogerá la habitación que desee mientras esté disponible.

REQUERIMIENTOS NO FUNCIONALES:

Proceso de Negocio	Actividad del Negocio	Responsable del Negocio	Requisito No Funcional	
Usabilidad	Mantenimiento de aplicación	orden de trabajo de mantenimiento	RF-01	Contar con una persona encargada para el mantenimiento de la aplicación
			RF-02	Equipo idóneo (computadoras, impresoras, teclado, etc.) para el uso de la aplicación correctamente

			RF-03	Interfaz Gráfica Agradable
Confiabilidad y seguridad.	Privacidad	Administrador	RF-04	Solo el usuario administrador tendrá permisos para realizar modificaciones a los registros, como eliminar, agregar etc.
			RF-05	Usuario administrador tendrá acceso al sistema por medio de usuario con contraseña.
Requerimientos de espacio	Servidor	Empleado	RF-06	Los servidores estarán ubicados en el Centro de Cómputo
Robustez	Sistemas	Sistema web	RF-07	El sistema esté disponible reservación de alguna habitación. cuando el usuario intente hacer una
Eficiencia y rendimiento.	Reservación	Aplicación	RF-08	Consultas y entradas sean realizadas de manera ágil y en un tiempo adecuado para el cliente.
			RF-09	El sistema de funcionar 99 por ciento del tiempo.
			RF-10	Se debe contar con un tiempo de respuesta razonable.
			RF-11	Se debe contar con una velocidad de conexión a internet.
Mantenimiento	Mantenimiento	orden de trabajo de mantenimiento	RF-12	Se tendrá un manual para dar soporte al sistema realizado.
			RF-13	La documentación técnica se entregará en formato digital.

Requerimientos de entrega	Registro de datos	Aplicación	RF-14	El manual de usuario se entregará en formato digital.
			RF-15	Se deberá utilizar sistemas operativos de Microsoft.
			RF-16	Se deberá incluir un servidor.
Propiedades del diseño	Desarrollador web	Programador web	RF-17	Desarrollado con Microsoft Visual Studio.NET
			RF-18	Desarrollado con una arquitectura de 3 capas.
			RF-19	Implementado para que trabaje con Sql Server.
			RF-20	Utilización de herramientas Css para desarrollar ayudas en línea en formato html.
			RF-21	Implementación del modelo RUB.
Interfaces	Conexión funcional	Aplicación	RF-22	La interfaz se podrá observar en cualquier navegador.
Licenciamiento	Requerimientos de Licenciamiento	Encargado del puesto	RF-23	Se debe contar con una licencia Microsoft

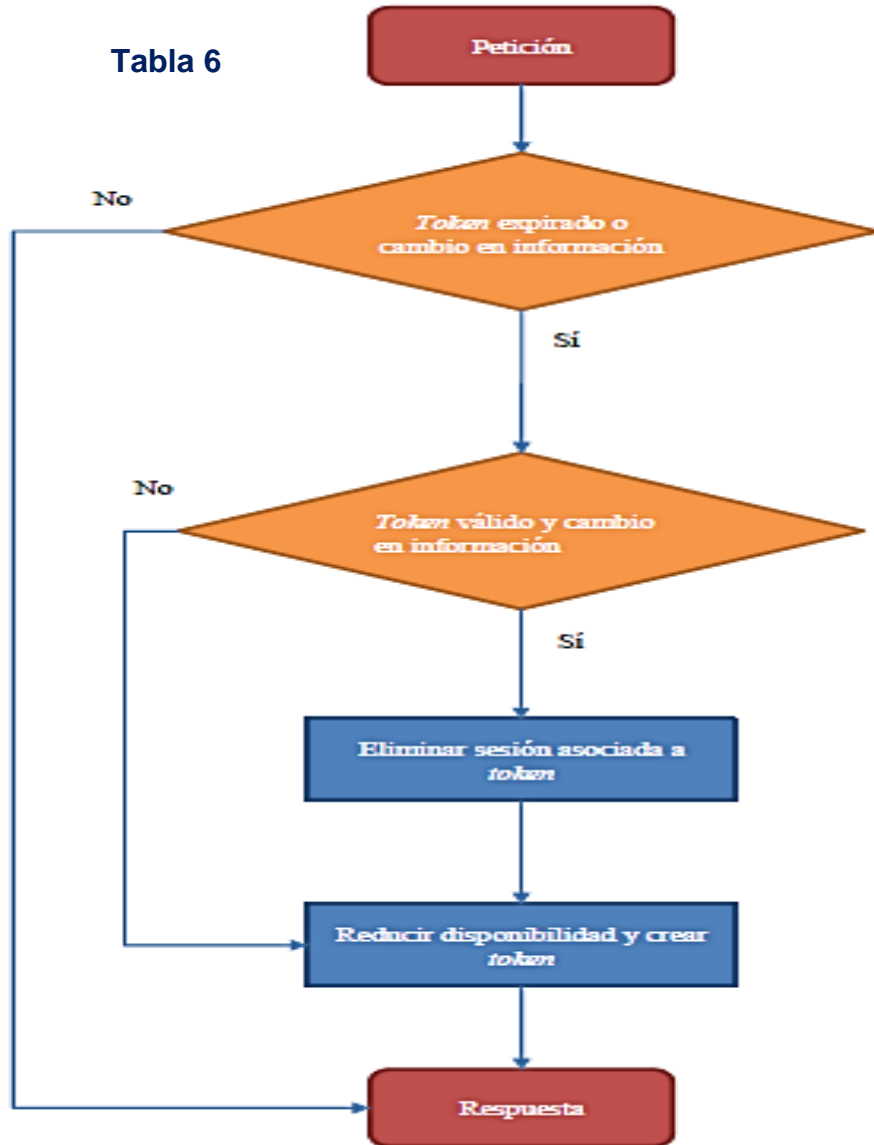
MODELO DEL SOFTWARE

MODELO DE NEGOCIO

❖ Reserva temporal de disponibilidad. `newCheckoutSession()`

Este token contiene la información de la reserva necesaria para que, en próximos accesos al servicio, se pueda comprobar el estado de la reserva, pudiendo realizar una nueva descartando la anterior en el caso de que cambiara su estado.

Tabla 6



Reserva definitiva. checkCheckoutSession()

Se comprueba nuevamente la disponibilidad y el precio para evitar posibles fallos. Se anula la reserva temporal y se reserva nuevamente de manera definitiva. Se pasa la petición a la función del pago.

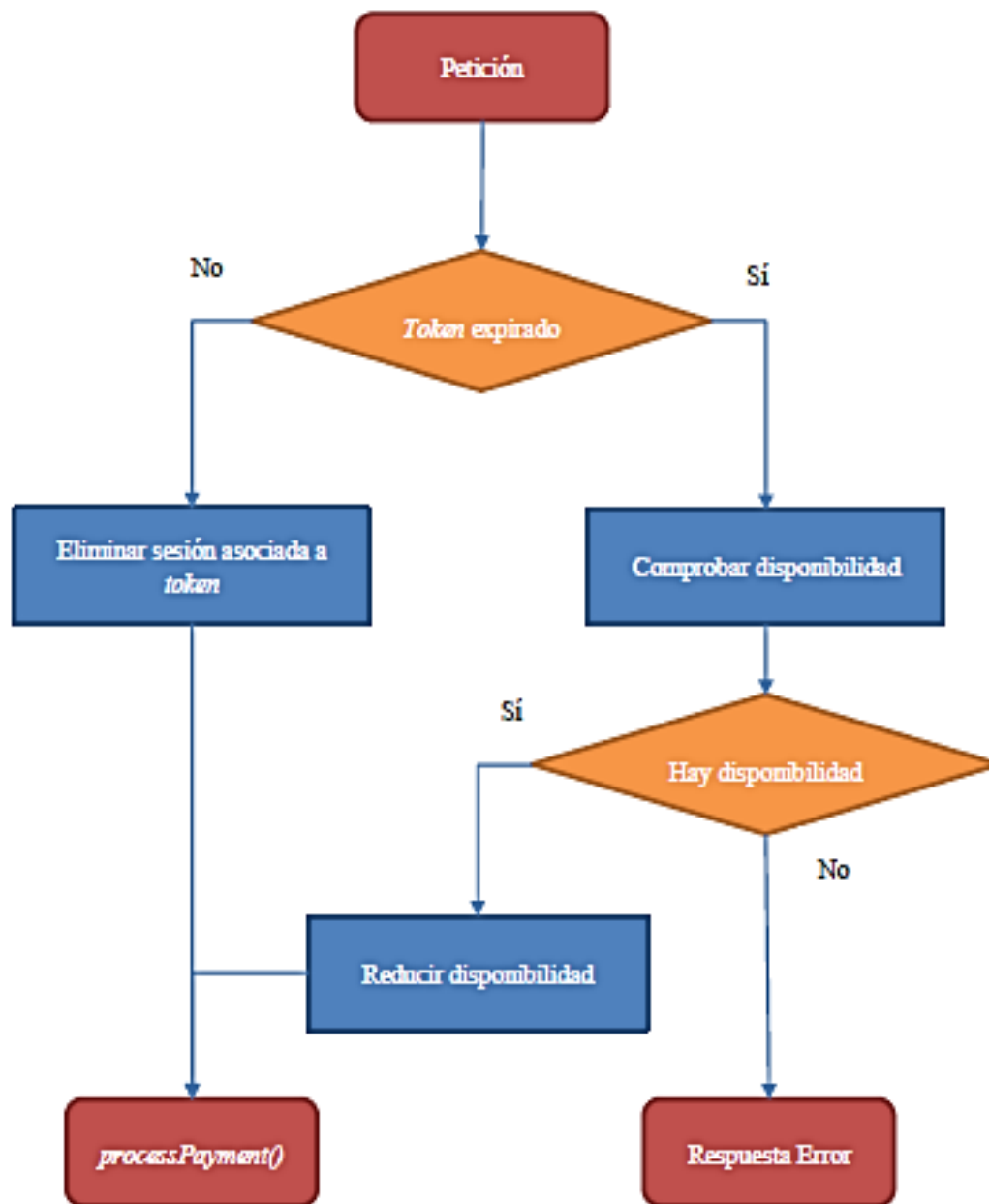


Tabla 7

❖ **Pago. processPayment()**

Se hace uso de un servicio de pago externo, el cual procesa la información de la tarjeta contenida en la petición para el cobro indicado. Se pasa la petición a la

función de envío de email de confirmación en el caso de que se haya realizado con éxito; se devuelve error en la respuesta en caso contrario.

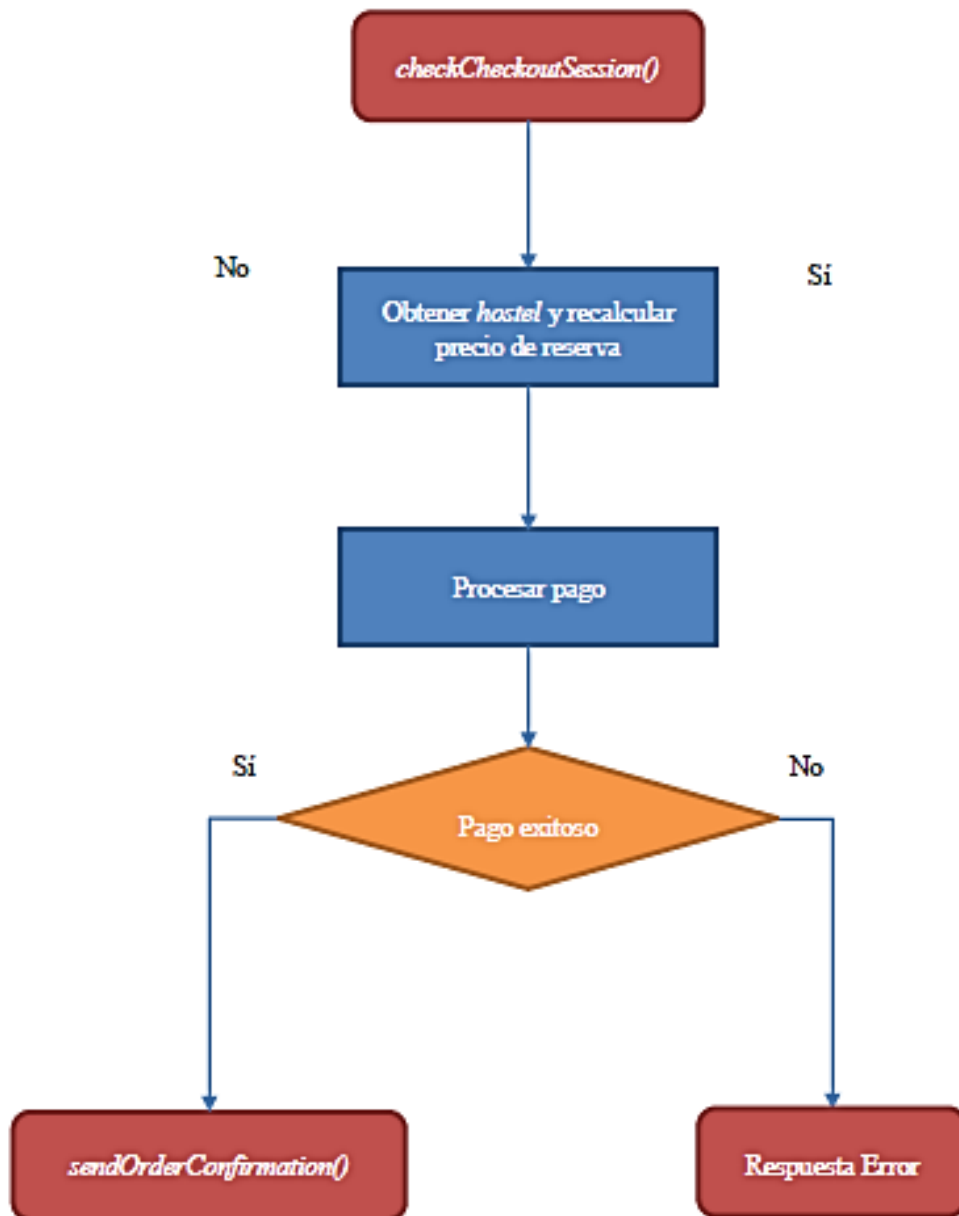


Tabla 8

❖ Envío de email de confirmación. *sendOrderConfirmation()*

Se emplea un módulo denominado *sendinblue* para el envío de una plantilla de email generada dinámicamente a partir de los datos de la reserva y del cliente. Se devuelve éxito en la respuesta.

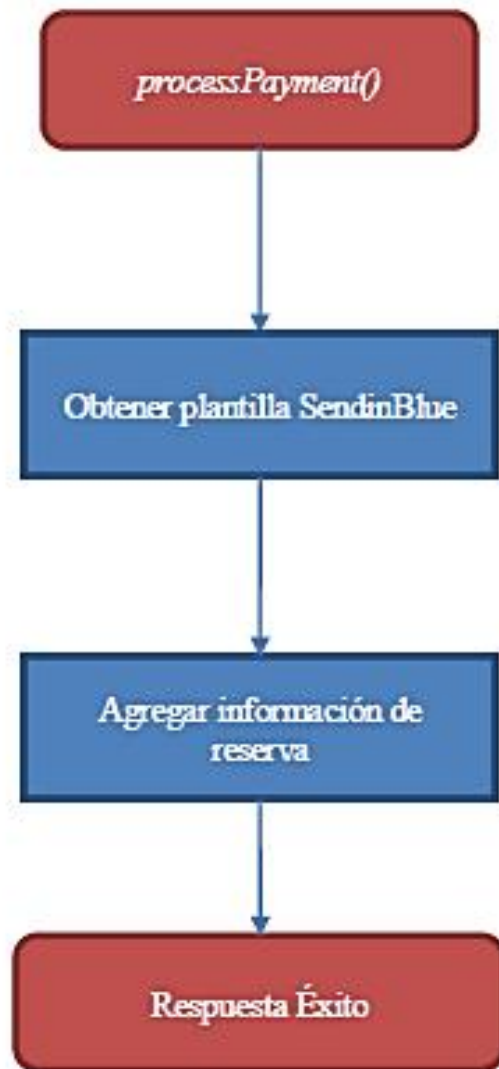


Tabla 9

MODELO DE CASO DE USO

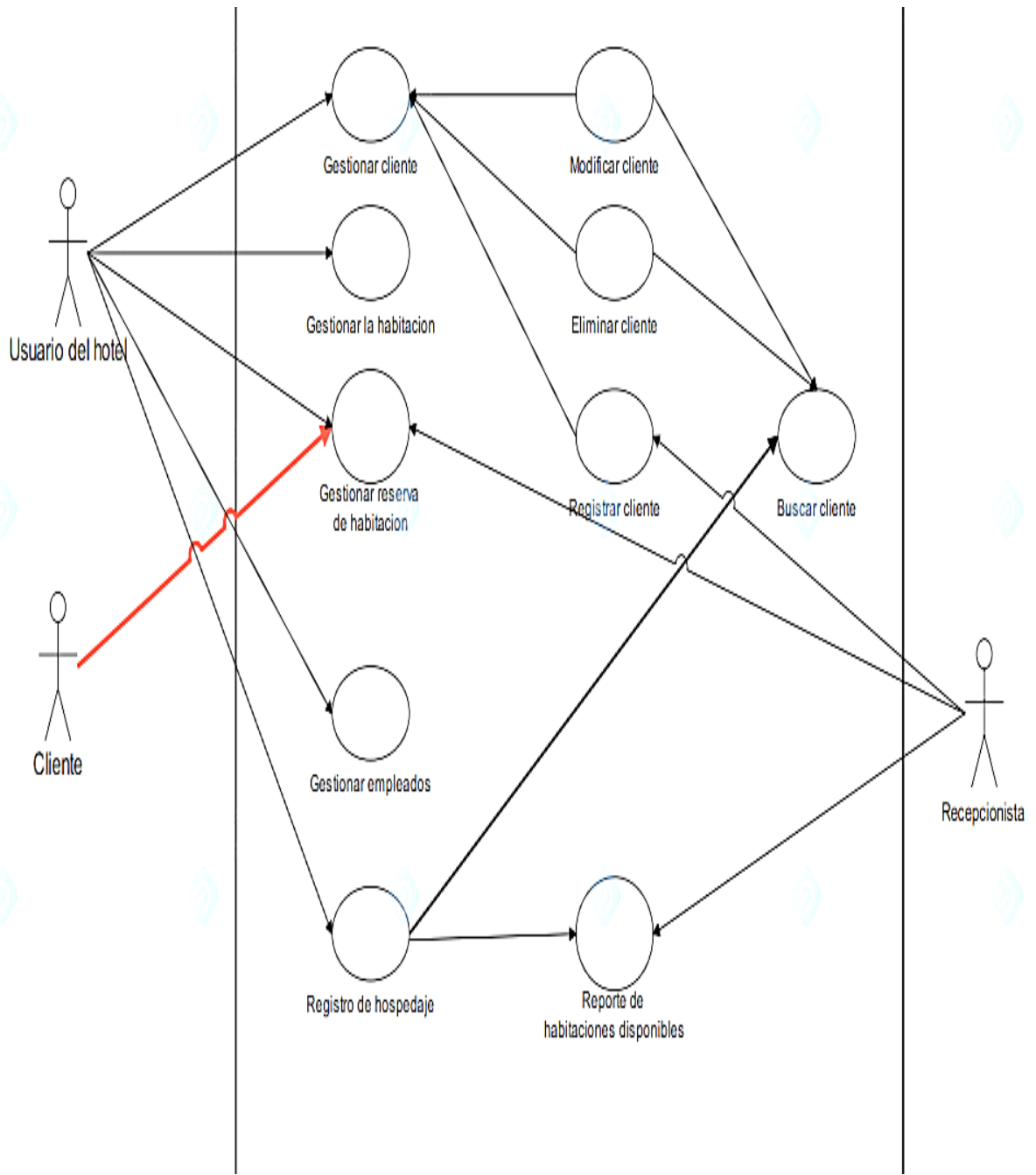


Tabla 10

Diagrama de estado

Diagrama de estado

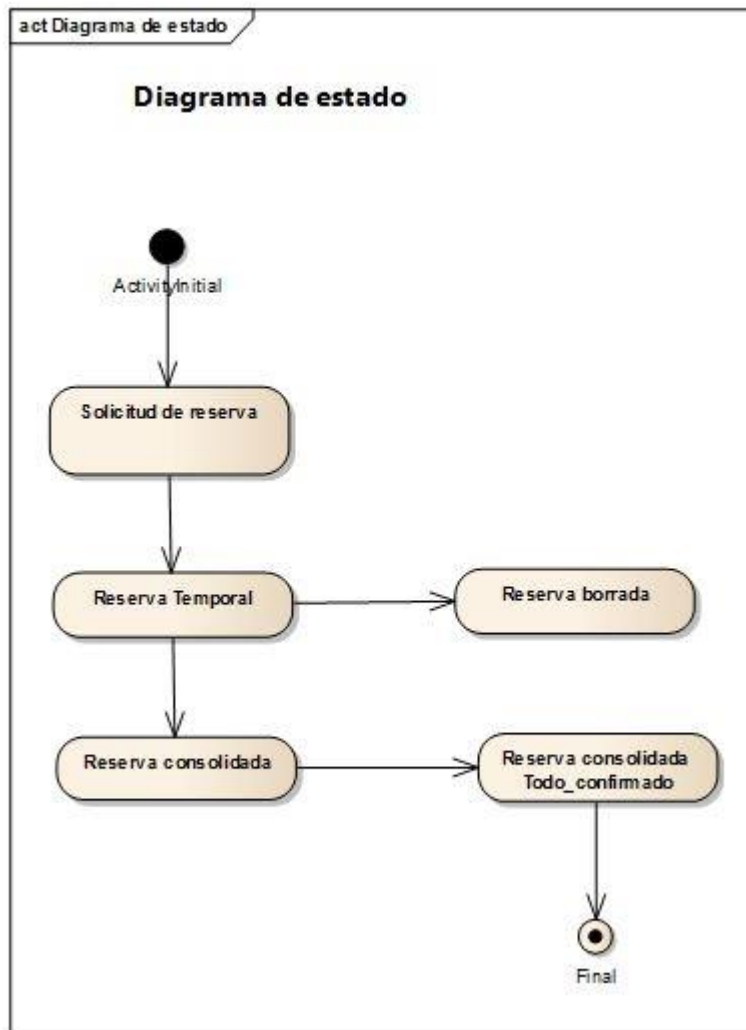
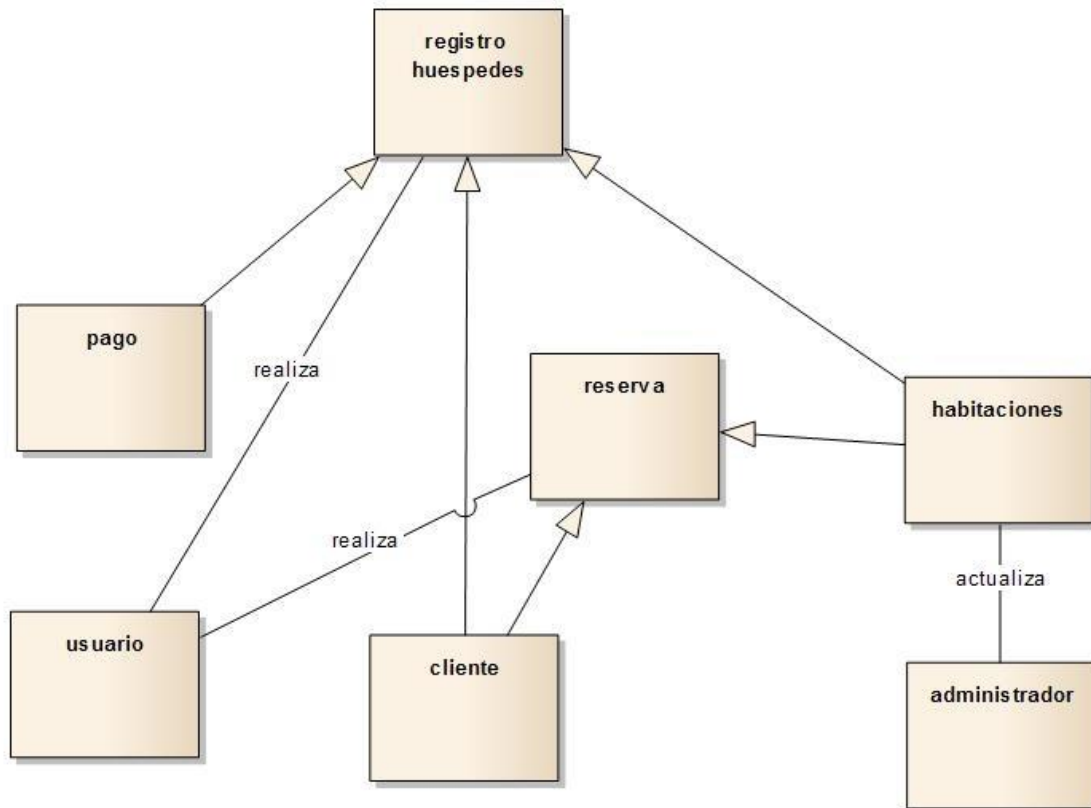


Diagrama de clases



DATOS DE LA APLICACIÓN

Una vez analizados los servicios y funcionalidades de la aplicación, resulta sencillo identificar la clase de información que se emplea. Cada tipo de información se define en un esquema, este esquema representa sus atributos y las relaciones con otros componentes del modelo de datos.

Elementos del cliente

El lado del cliente está compuesto por una serie de componentes y servicios, cuya misión es ofrecer al usuario las funcionalidades de la aplicación, haciendo uso de la API desarrollada. A continuación, se explica el flujo de la página web; se detallan los distintos componentes y servicios, indicando su función, la comunicación entre los mismos y las interacciones con el usuario.

Servicios

Ya se han mencionado los servicios utilizados, ahora se expone en detalle su funcionamiento.

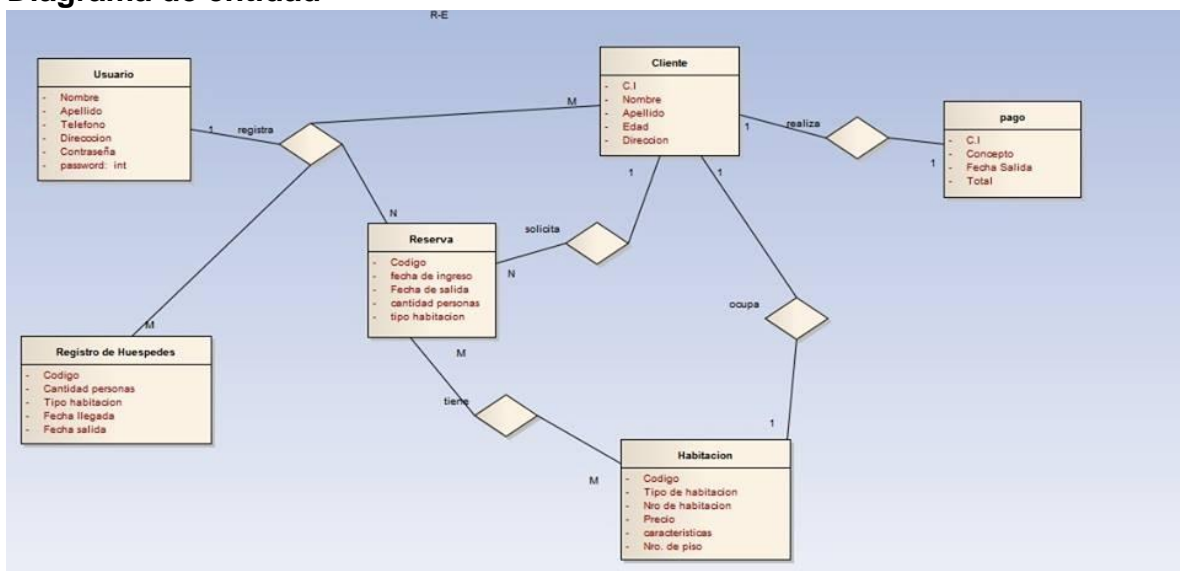
- City Service :Define métodos para realizar las peticiones a la API del servidor para obtener la lista de ciudades.
- Search Service :Define métodos para realizar las peticiones a la API del servidor para obtener los *hostels* de una ciudad. Tiene un método que permite filtrar los *hostels* según su disponibilidad para las fechas y huéspedes seleccionados.
- Hostel Service : Define métodos para realizar las peticiones a la API del servidor para obtener la información de un *hostel* concreto.

Permite la comunicación entre el componente *search* y *hostel*; usa el patrón Observable para que el primero pase la información que ya posee sobre el *hostel* seleccionado al segundo, para que la muestra sin tener que realizar una petición HTTP.

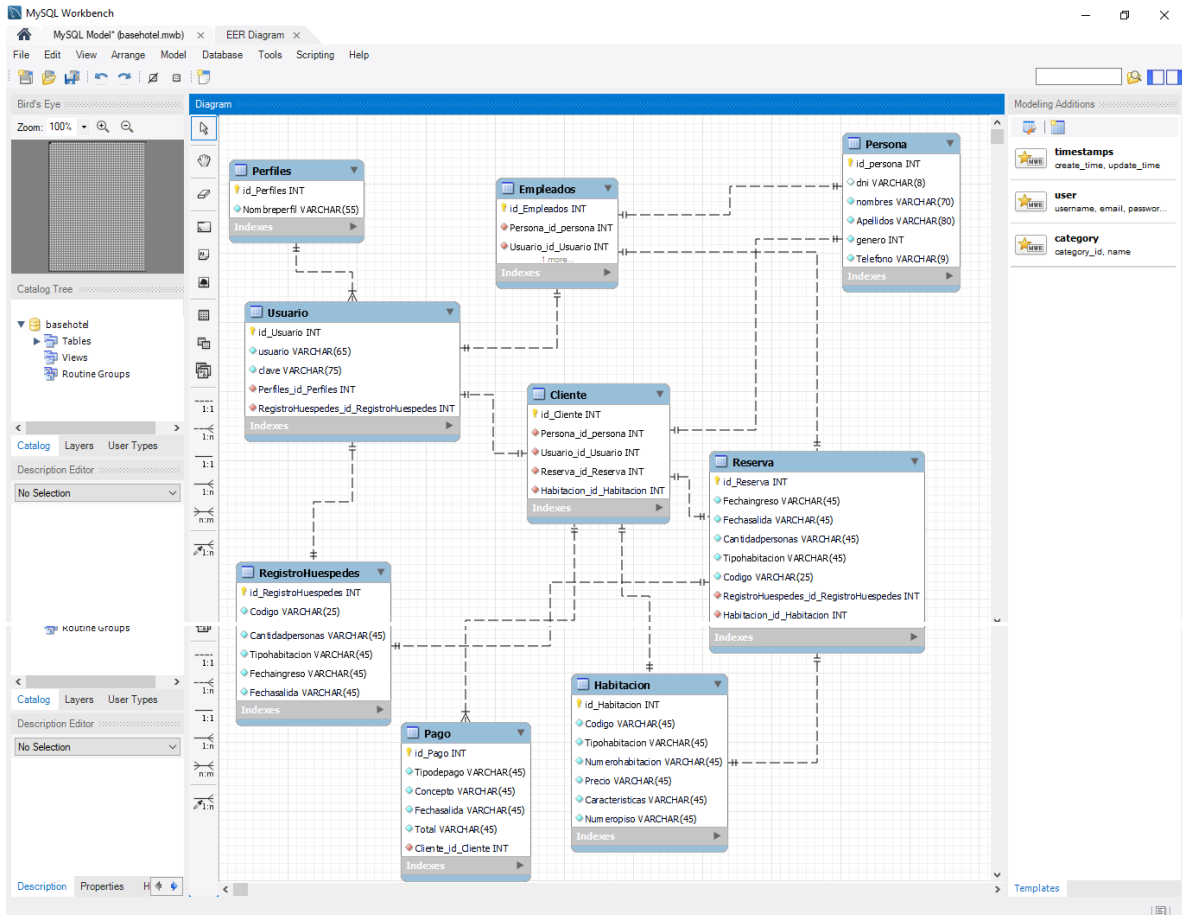
CAPITULO 4. Diseño

1.Diseño de la base de datos

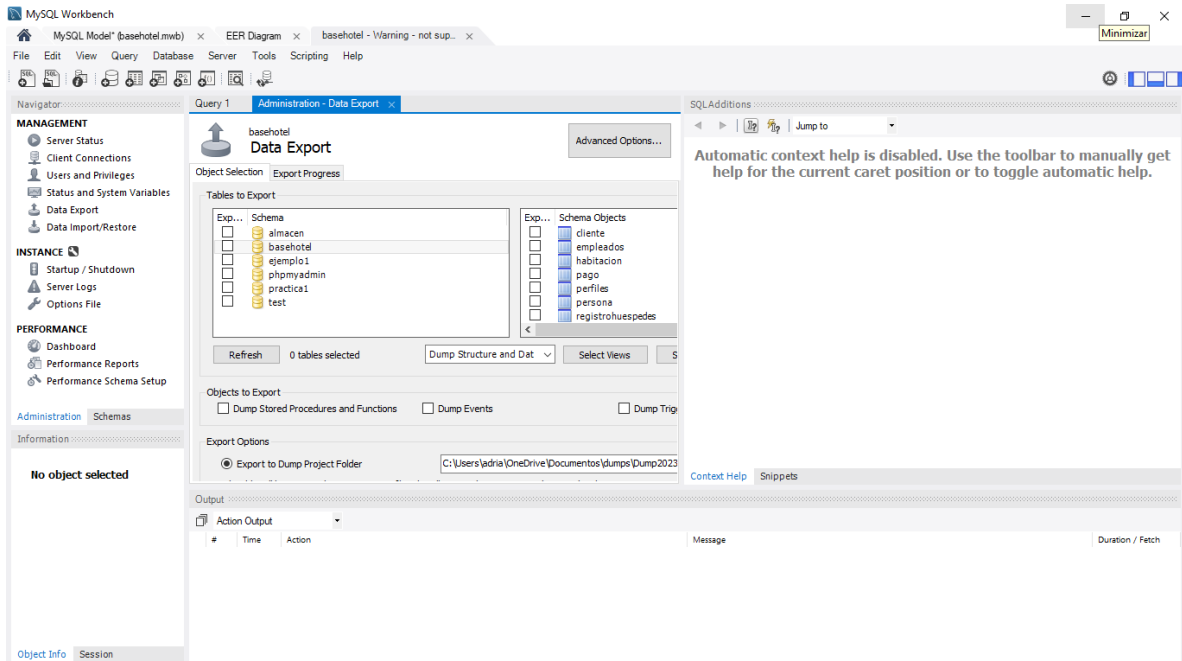
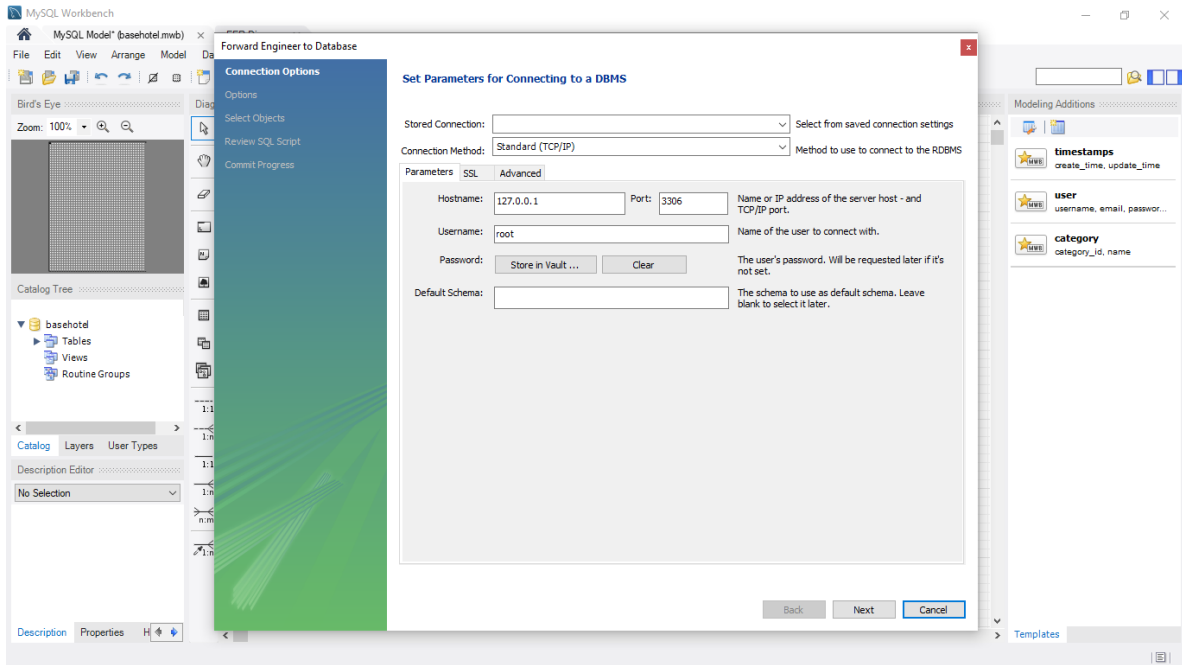
Diagrama de entidad



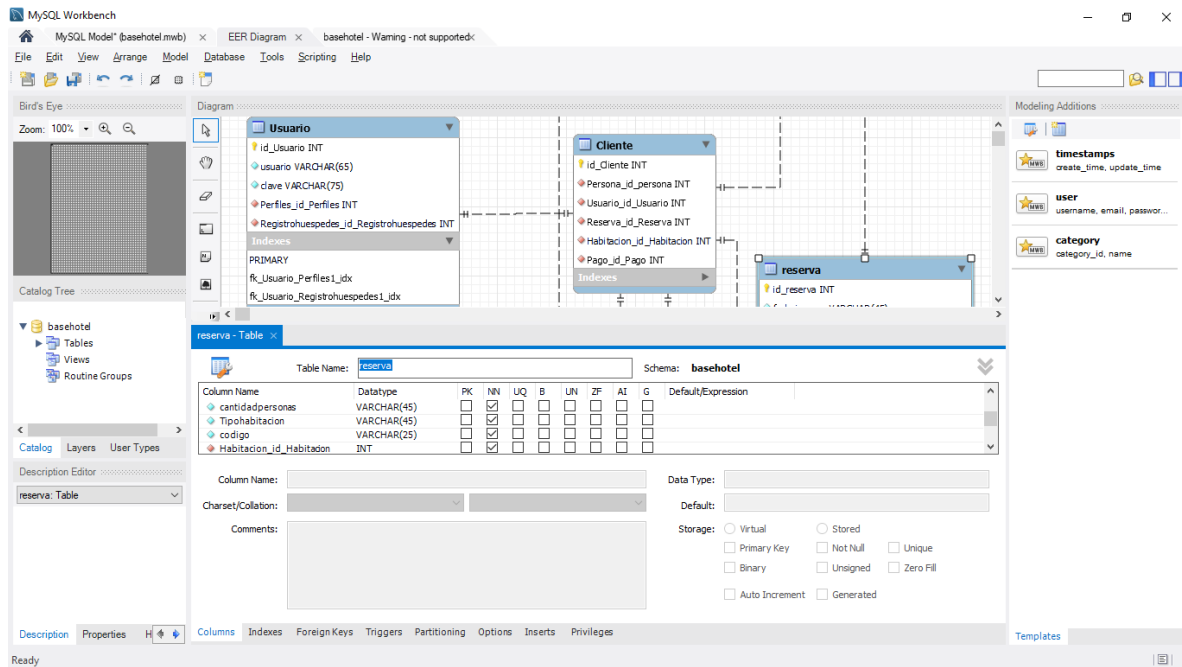
2. Diseño Logico



3. Diseño Físico



4. Normaizacion



5. Programacion

-- 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 basehotel

-- Schema basehotel

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

```
-----  
-- Table `basehotel`.`Persona`  
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Persona` (  
  `id_persona` INT NOT NULL AUTO_INCREMENT,  
  `dni` VARCHAR(8) NULL,  
  `nombres` VARCHAR(70) NOT NULL,  
  `Apellidos` VARCHAR(80) NOT NULL,  
  `genero` INT NOT NULL,  
  `Telefono` VARCHAR(9) NOT NULL,  
  PRIMARY KEY (`id_persona`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `basehotel`.`Perfiles`  
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Perfiles` (  
  `id_Perfiles` INT NOT NULL AUTO_INCREMENT,  
  `Nombreperfil` VARCHAR(55) NOT NULL,  
  PRIMARY KEY (`id_Perfiles`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `basehotel`.`Registrohuespedes`  
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Registrohuespedes` (  
  `id_Registrohuespedes` INT NOT NULL AUTO_INCREMENT,  
  `Codigo` VARCHAR(45) NOT NULL,  
  `Cantidadpersonas` VARCHAR(45) NOT NULL,  
  `Tipohabitacion` VARCHAR(45) NOT NULL,  
  `Fechaingreso` VARCHAR(45) NOT NULL,  
  `Fechasalida` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_Registrohuespedes`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `basehotel`.`Usuario`  
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Usuario` (  
  `id_Usuario` INT NOT NULL AUTO_INCREMENT,  
  `usuario` VARCHAR(65) NOT NULL,  
  `clave` VARCHAR(75) NOT NULL,  
  `Perfiles_id_Perfiles` INT NOT NULL,  
  `Registrohuespedes_id_Registrohuespedes` INT NOT NULL,  
  PRIMARY KEY (`id_Usuario`),  
  INDEX `fk_Usuario_Perfiles1_idx` (`Perfiles_id_Perfiles` ASC),  
  INDEX `fk_Usuario_Registrohuespedes1_idx`  
  (`Registrohuespedes_id_Registrohuespedes` ASC),  
  CONSTRAINT `fk_Usuario_Perfiles1`  
  FOREIGN KEY (`Perfiles_id_Perfiles`)
```

```
REFERENCES `basehotel`.`Perfiles` (`id_Perfiles`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario_Registrohuespedes1`
FOREIGN KEY (`Registrohuespedes_id_Registrohuespedes`)
REFERENCES `basehotel`.`Registrohuespedes` (`id_Registrohuespedes`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `basehotel`.`Habitacion`
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Habitacion` (
  `id_Habitacion` INT NOT NULL AUTO_INCREMENT,
  `Codigo` VARCHAR(45) NOT NULL,
  `Tipohabitacion` VARCHAR(45) NOT NULL,
  `Numerohabitacion` VARCHAR(45) NOT NULL,
  `Precio` VARCHAR(45) NOT NULL,
  `Caracteristicas` VARCHAR(45) NOT NULL,
  `Numeropiso` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id_Habitacion`))
ENGINE = InnoDB;
```

```
-----
-- Table `basehotel`.`reserva`
```

```

-----
CREATE TABLE IF NOT EXISTS `basehotel`.`reserva` (
  `id_reserva` INT NOT NULL AUTO_INCREMENT,
  `fechaingreso` VARCHAR(45) NOT NULL,
  `fechasalida` VARCHAR(45) NOT NULL,
  `cantidadpersonas` VARCHAR(45) NOT NULL,
  `Tipohabitacion` VARCHAR(45) NOT NULL,
  `codigo` VARCHAR(25) NOT NULL,
  `Habitacion_id_Habitacion` INT NOT NULL,
  `Registrohuespedes_id_Registrohuespedes` INT NOT NULL,
  PRIMARY KEY (`id_reserva`),
  INDEX `fk_Reserva_Habitacion1_idx` (`Habitacion_id_Habitacion` ASC),
  INDEX `fk_Reserva_Registrohuespedes1_idx`
  (`Registrohuespedes_id_Registrohuespedes` ASC),
  CONSTRAINT `fk_Reserva_Habitacion1`
    FOREIGN KEY (`Habitacion_id_Habitacion`)
    REFERENCES `basehotel`.`Habitacion` (`id_Habitacion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Reserva_Registrohuespedes1`
    FOREIGN KEY (`Registrohuespedes_id_Registrohuespedes`)
    REFERENCES `basehotel`.`Registrohuespedes` (`id_Registrohuespedes`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `basehotel`.`Pago`

```

```
-----  
CREATE TABLE IF NOT EXISTS `basehotel`.`Pago` (  
  `id_Pago` INT NOT NULL AUTO_INCREMENT,  
  `Concepto` VARCHAR(45) NOT NULL,  
  `Fechasalida` VARCHAR(45) NOT NULL,  
  `Total` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_Pago`))  
ENGINE = InnoDB;
```

```
-----  
-- Table `basehotel`.`Cliente`  
-----
```

```
CREATE TABLE IF NOT EXISTS `basehotel`.`Cliente` (  
  `id_Cliente` INT NOT NULL AUTO_INCREMENT,  
  `Persona_id_persona` INT NOT NULL,  
  `Usuario_id_Usuario` INT NOT NULL,  
  `Reserva_id_Reserva` INT NOT NULL,  
  `Habitacion_id_Habitacion` INT NOT NULL,  
  `Pago_id_Pago` INT NOT NULL,  
  PRIMARY KEY (`id_Cliente`),  
  INDEX `fk_Cliente_Persona1_idx` (`Persona_id_persona` ASC),  
  INDEX `fk_Cliente_Usuario1_idx` (`Usuario_id_Usuario` ASC),  
  INDEX `fk_Cliente_Reserva1_idx` (`Reserva_id_Reserva` ASC),  
  INDEX `fk_Cliente_Habitacion1_idx` (`Habitacion_id_Habitacion` ASC),  
  INDEX `fk_Cliente_Pago1_idx` (`Pago_id_Pago` ASC),  
  CONSTRAINT `fk_Cliente_Persona1`  
    FOREIGN KEY (`Persona_id_persona`)
```



```
REFERENCES `basehotel`.`Persona` (`id_persona`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Cliente_Usuario1`
FOREIGN KEY (`Usuario_id_Usuario`)
REFERENCES `basehotel`.`Usuario` (`id_Usuario`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Cliente_Reserva1`
FOREIGN KEY (`Reserva_id_Reserva`)
REFERENCES `basehotel`.`reserva` (`id_reserva`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Cliente_Habitacion1`
FOREIGN KEY (`Habitacion_id_Habitacion`)
REFERENCES `basehotel`.`Habitacion` (`id_Habitacion`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_Cliente_Pago1`
FOREIGN KEY (`Pago_id_Pago`)
REFERENCES `basehotel`.`Pago` (`id_Pago`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

-- Table `basehotel`.`Empleados`

```

-----
CREATE TABLE IF NOT EXISTS `basehotel`.`Empleados` (
  `id_Empleados` INT NOT NULL AUTO_INCREMENT,
  `Persona_id_persona` INT NOT NULL,
  `Usuario_id_Usuario` INT NOT NULL,
  `Reserva_id_Reserva` INT NOT NULL,
  `Habitacion_id_Habitacion` INT NOT NULL,
  PRIMARY KEY (`id_Empleados`),
  INDEX `fk_Empleados_Persona1_idx` (`Persona_id_persona` ASC),
  INDEX `fk_Empleados_Usuario1_idx` (`Usuario_id_Usuario` ASC),
  INDEX `fk_Empleados_Reserva1_idx` (`Reserva_id_Reserva` ASC),
  INDEX `fk_Empleados_Habitacion1_idx` (`Habitacion_id_Habitacion` ASC),
  CONSTRAINT `fk_Empleados_Persona1`
    FOREIGN KEY (`Persona_id_persona`)
    REFERENCES `basehotel`.`Persona` (`id_persona`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Empleados_Usuario1`
    FOREIGN KEY (`Usuario_id_Usuario`)
    REFERENCES `basehotel`.`Usuario` (`id_Usuario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Empleados_Reserva1`
    FOREIGN KEY (`Reserva_id_Reserva`)
    REFERENCES `basehotel`.`reserva` (`id_reserva`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Empleados_Habitacion1`

```

```
FOREIGN KEY (`Habitacion_id_Habitacion`)  
REFERENCES `basehotel`.`Habitacion` (`id_Habitacion`)  
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;
```

6.Herramientas

MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL

Paso 1: Ve a la página de MySQL Workbench en AcademicSoftware y haz clic en el botón 'Descargar MySQL Workbench' para descargar el archivo de instalación.

Paso 2: Abre el archivo de instalación MSI de tu carpeta de descargas y haz clic en Next para iniciar la instalación.



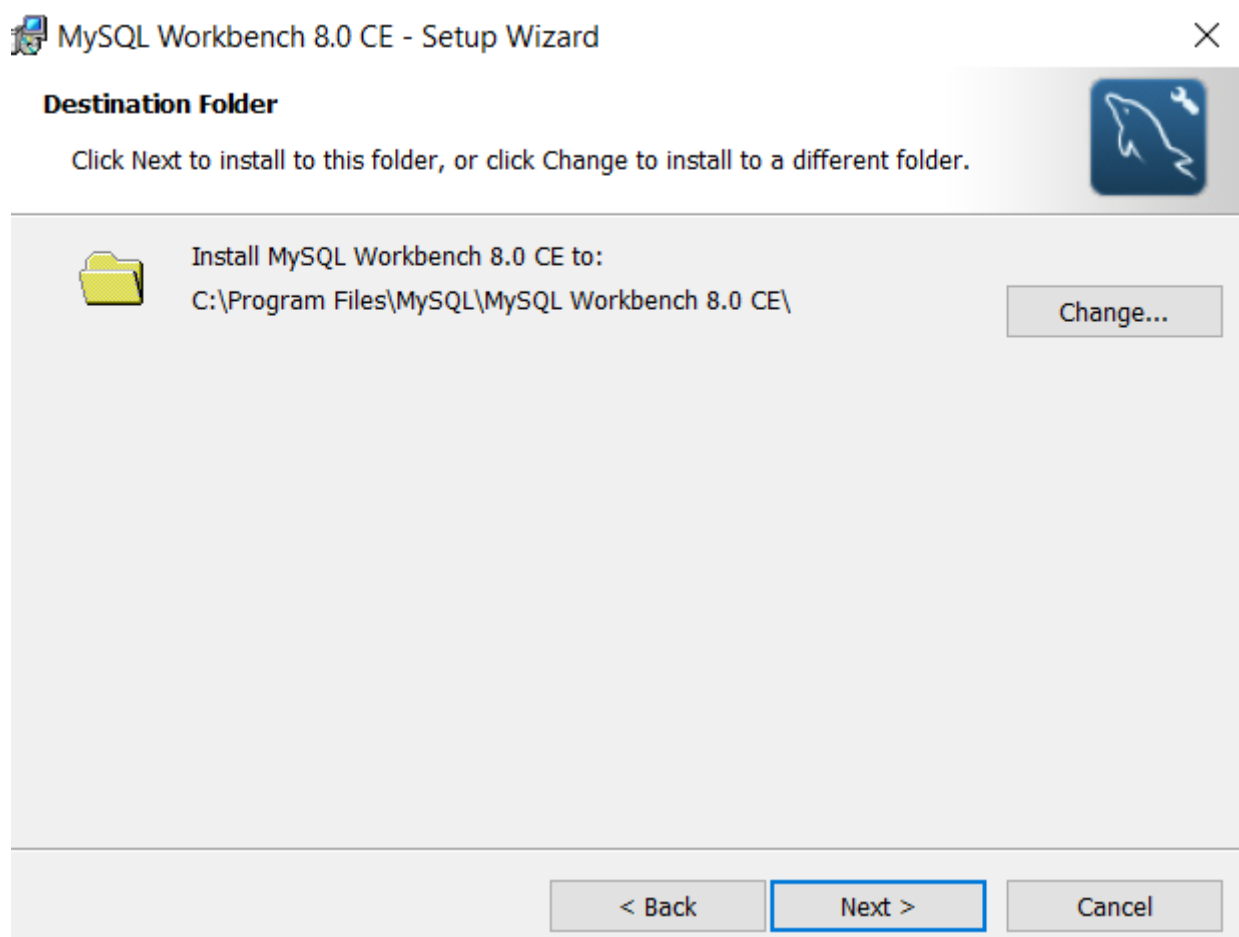
Welcome to the Setup Wizard for MySQL Workbench 8.0 CE

The Setup Wizard will install version 8.0.23 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

< Back Next > Cancel

Paso 3: Haz clic en Next para instalar el software en la ubicación predeterminada. También puedes cambiar la carpeta de destino.



Paso 4: Haz clic en Next para instalar todas las funcionalidades del programa o elige Custom para los usuarios avanzados.

MySQL Workbench 8.0 CE - Setup Wizard



Ready to Install the Program



The wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Current Settings:

Setup Type:

Complete

Destination Folder:

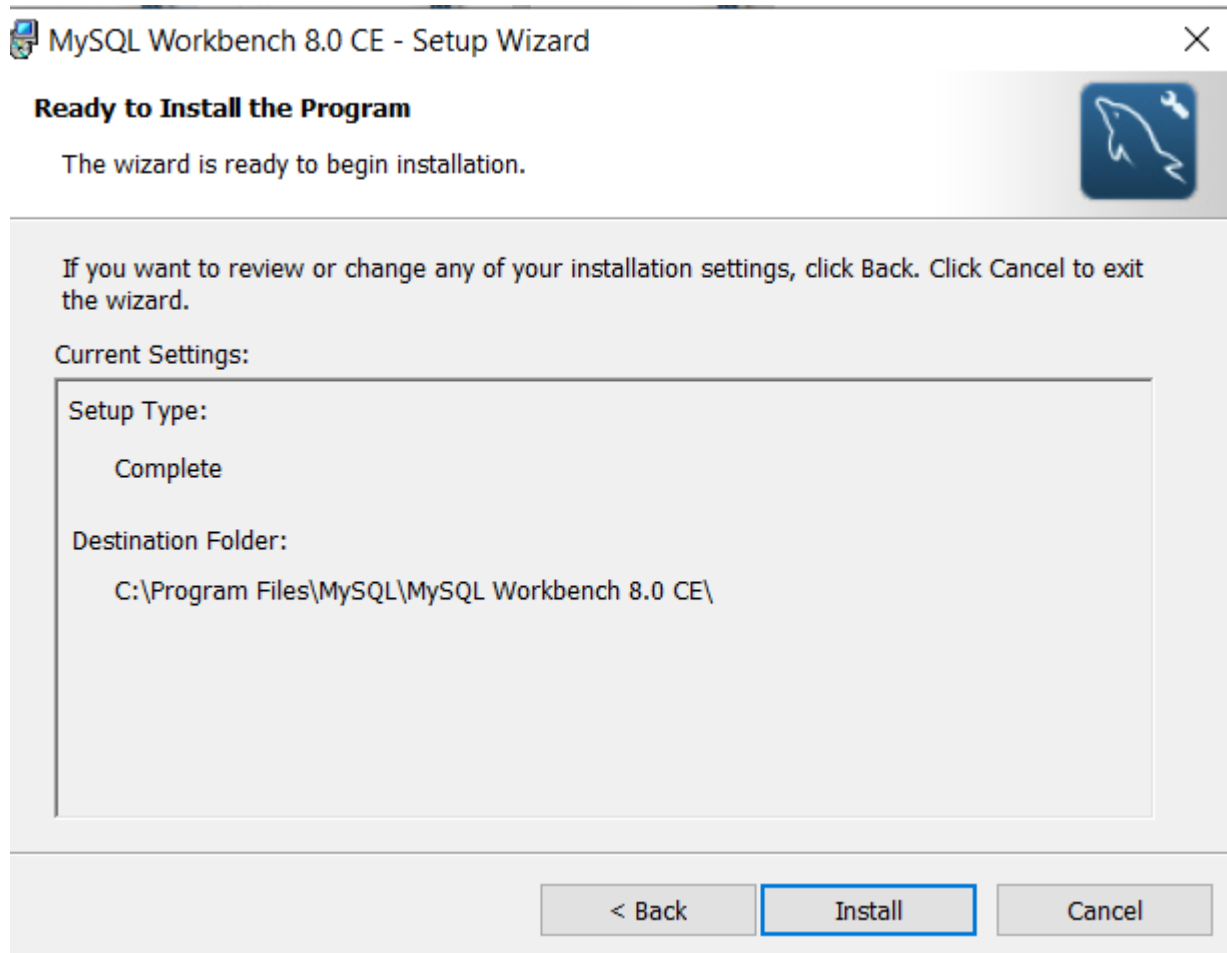
C:\Program Files\MySQL\MySQL Workbench 8.0 CE\

< Back

Install

Cancel

Paso 5: Haz clic en Install para iniciar la instalación.



Paso 6: Una vez finalizada la instalación, marca la casilla para lanzar el programa y haz clic en Finish.
El programa está instalado y listo para usar.

MySQL Workbench 8.0 CE - Setup Wizard



Installing MySQL Workbench 8.0 CE



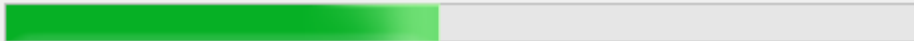
The program features you selected are being installed.



Please wait while the Setup Wizard installs MySQL Workbench 8.0 CE. This may take several minutes.

Status:

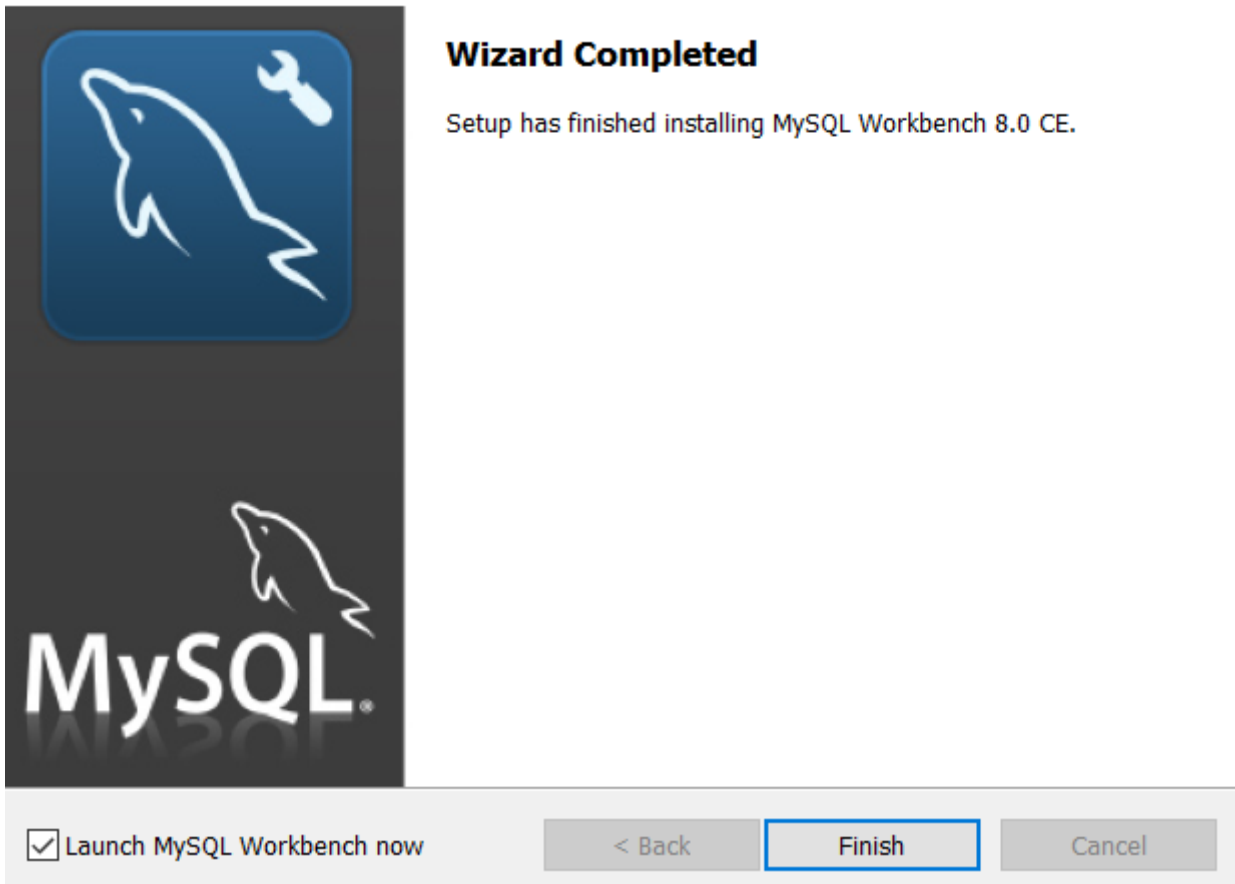
Copying new files



< Back

Next >

Cancel



CONCLUSIÓN

La globalización y la reducción de precios en el transporte ha favorecido la creciente demanda turística en todo el mundo. Este modelo de alojamiento permite reducir los precios; además, el incentivo de tratar con otras personas y conocer gente de diferentes países ha propiciado también su popularización.

Dentro de este mercado fructífero tiene también su sitio la tecnología. Desde antes del 2000 han ido apareciendo nuevos portales web para ofrecer alojamiento, actuando de intermediarios entre el turista y el anfitrión. Estos portales han ganado poco a poco el terreno a las empresas físicas de turismo, dada su mayor capacidad

de adaptación a los nuevos modelos; así como la expansión del comercio electrónico en general.

BIBLIOGRAFÍA

[1] S. Guisasola Benítez. <https://idus.us.es/bitstream/handle/11441/93427/TFG-2387-GUISASOLA.pdf?sequence=1&isAllowed=y> (accedido el 20 de noviembre de 2022).

[2] M. Ramos Otero. https://oa.upm.es/39946/10/TFG_Melanie_Ramos_Otero.pdf (accedido el 20 de noviembre de 2022)

[3] O. O. Díaz Ochoa. "IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE RESERVAS Y DE ESPACIOS PARA LA DIRECCIÓN TÉCNICA DE ADMINISTRACIÓN E INVENTARIOS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE GUAYAQUIL". <https://dspace.ups.edu.ec/bitstream/123456789/20569/1/UPS-GT0023300.pdf> (accedido el 20 de noviembre de 2022).

Evaluación formativa

Evaluar los conocimientos adquiridos sobre el contenido de la unidad 1

* Indica que la pregunta es obligatoria

1. Nombre completo *

Conceptos generales

2. Primera fase del ciclo de vida del desarrollo de sistemas *

1 punto

Marca solo un óvalo.

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

3. El analista trabaja con los programadores para desarrollar el software original

* 1 punto

Selecciona todos los que correspondan.

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

4. Identificar las necesidades de los usuarios involucrados, mediante instrumentos de recolección de datos para conocer su contexto

* 1 punto

Marca solo un óvalo.

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

5. Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.

* 1 punto

Marca solo un óvalo.

- Software
- Ingeniería de software
- Sistema de información
- Sistema

6. Instrucciones que cuando se ejecutan proporcionan las características, función y desempeños buscados

* 1 punto

Marca solo un óvalo.

- Software
- Ingeniería de sistemas
- Sistemas de información
- Sistemas

7. Herramientas, métodos, proceso y compromiso con la calidad son capas de ^{*} 1 punto
la

Marca solo un óvalo.

- Sosftware
- Ingeniería de software
- Sistemas de información
- Sistemas

8. Modelo evolutivo con naturaleza iterativa de prototipos, generador de ^{*} 1 punto
modelo de proceso impulsado por el riesgo

Marca solo un óvalo.

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

9. Sugiere un enfoque sistemático y secuencial para el desarrollo del software ^{*} 1 punto

Marca solo un óvalo.

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

10. Ejecuta una serie de avances llamados incrementos, que en forma progresiva da mas funcionalidad al cliente 1 punto

Marca solo un óvalo.

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

11. Modelo que sirve para generar el "primer sistema" , además de servir como mecanismo para identificar los requerimientos del software. * 1 punto

Marca solo un óvalo.

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

MODELADO

12. Comparten una estructura y comportamiento común, además representa una abstracción. * 1 punto

Marca solo un óvalo.

- Caso de uso
- Clase
- Actor
- Objeto

13. Es la instancia de una clase *

1 punto

Marca solo un óvalo.

- Caso de uso
- Clase
- Actor
- Objeto

14. Mecanismo para describir y registrar los requerimientos funcionales. *

1 punto

Marca solo un óvalo.

- Caso de uso
- Clase
- Actor
- Objeto

15. Representa el papel desempeñado en el sistema. *

1 punto

Marca solo un óvalo.

- Caso de uso
- Clase
- Actor
- Objeto

16. Secuencia específica de acciones e interacciones entre los actores y el sistema objeto de estudio * 1 punto

Marca solo un óvalo.

- Caso de uso
- Clase
- Actor
- Escenario

17. Es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir. * 1 punto

Marca solo un óvalo.

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

18. Conjunto complejo de rutinas interdependientes que se descubren, ajustan y matizan mediante la acción * 1 punto

Marca solo un óvalo.

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

19. Conjunto de casos de uso con uno o más actores *

1 punto

Marca solo un óvalo.

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

20. Modelo que se desarrolla a partir de los requerimientos funcionales *

1 punto

Marca solo un óvalo.

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

21. Modelo que es una especie de anteproyecto de una estrategia, ya que inicia con el listado de clientes y trabajadores del negocio y sus necesidades.

* 1 punto

Marca solo un óvalo.

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

ANALISIS DE REQUERIMIENTOS

22. Declaración de los servicios que se espera que el sistema proporcione y restricciones bajo las cuales debe funcionar * 1 punto

Marca solo un óvalo.

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

23. Funciones, servicios y restricciones operativas del sistema. Define que es lo que se va a implementar. * 1 punto

Marca solo un óvalo.

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

24. Son declaraciones de los servicios que debe proporcionar el sistema. * 1 punto

Marca solo un óvalo.

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

25. Son restricciones de los servicios o funciones ofrecidos por el sistema. * 1 punto

Marca solo un óvalo.

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

26. Se clasifican en requerimientos del producto, organizacionales y externos * 1 punto

Marca solo un óvalo.

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Base de datos

Sistema de Gestión de Base de Datos



[http://youtube.com/watch?](http://youtube.com/watch?v=L4WB2Tt55hs)

[v=L4WB2Tt55hs](http://youtube.com/watch?v=L4WB2Tt55hs)

27. ¿Qué es una base de datos? *

28. ¿Qué es un sistema manejador de base de datos? *

29. Selecciona los mejores motores de base de datos *

1 punto

Selecciona todos los que correspondan.

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

30. SGBD que incluye un motor mediante una interfaz gráfica, es usado por grandes compañías, además de ser un sistema muy robusto y seguro * 1 punto

Marca solo un óvalo.

- ORACLE
 ACCESS
 Microsoft SQLServer
 PostgreSQL
 MySQL

31. SGBD que permite soportar transacciones y gestiona procedimientos almacenados cliente servidor, fue realizado por Microsoft * 1 punto

Marca solo un óvalo.

- ORACLE
 ACCESS
 Microsoft SQLServer
 PostgreSQL
 MySQL

32. SGBD de código libre, es un sistema de acceso múltiple de datos en sus consultas y utiliza grandes extensiones de texto. * 1 punto

Marca solo un óvalo.

- ORACLE
 ACCESS
 Microsoft SQLServer
 PostgreSQL
 MySQL

33. SGBD de código libre, a deferencia que maneja una licencia. Es el sistema * 1 punto más utilizado debido a que se integra con lenguales de programación php y HTML

Marca solo un óvalo.

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Evaluación formativa

Evaluar los conocimientos adquiridos sobre el contenido de la unidad 1

Nombre completo *

ALFONSO VELASCO QUINO

Conceptos generales

Primera fase del ciclo de vida del desarrollo de sistemas *

1 punto

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

El analista trabaja con los programadores para desarrollar el software original *

1 punto

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

Identificar las necesidades de los usuarios involucrados, mediante instrumentos de recolección de datos para conocer su contexto

* 1 punto

- Determinación de los requerimientos
- Diseño del sistema
- Identificación de los problemas, oportunidades y objetivos
- Desarrollo y documentación del software

Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software.

* 1 punto

- Software
- Ingeniería de software
- Sistema de información
- Sistema

Instrucciones que cuando se ejecutan proporcionan las características, función y desempeños buscados

* 1 punto

- Software
- Ingeniería de sistemas
- Sistemas de información
- Sistemas

Herramientas, métodos, proceso y compromiso con la calidad son capas de la *

1 punto

- Sosftware
- Ingeniería de software
- Sistemas de información
- Sistemas

Modelo evolutivo con naturaleza iterativa de prototipos, generador de modelo de proceso * 1 punto
impulsado por el riesgo

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

Sugiere un enfoque sistemático y secuencial para el desarrollo del software *

1 punto

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

Ejecuta una serie de avances llamados incrementos, que en forma progresiva da mas funcionalidad al cliente

1 punto

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

Modelo que sirve para generar el "primer sistema" , además de servir como mecanismo para identificar los requerimientos del software. * 1 punto

- Modelo en cascada
- Modelo incremental
- Modelo espiral
- Prototipo

MODELADO

Comparten una estructura y comportamiento común, además representa una abstracción.

* 1 punto

- Caso de uso
- Clase
- Actor
- Objeto

Es la instancia de una clase *

1 punto

- Caso de uso
- Clase
- Actor
- Objeto

Mecanismo para describir y registrar los requerimientos funcionales. *

1 punto

- Caso de uso
- Clase
- Actor
- Objeto

Representa el papel desempeñado en el sistema. *

1 punto

- Caso de uso
- Clase
- Actor
- Objeto

Secuencia específica de acciones e interacciones entre los actores y el sistema objeto de estudio * 1 punto

- Caso de uso
- Clase
- Actor
- Escenario

Es utilizado por el analista como un medio para comprender el sector de negocios al cual el sistema va a servir. * 1 punto

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

Conjunto complejo de rutinas interdependientes que se descubren, ajustan y matizan mediante la acción * 1 punto

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

Conjunto de casos de uso con uno o más actores *

1 punto

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

Modelo que se desarrolla a partir de los requerimientos funcionales *

1 punto

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

Modelo que es una especie de anteproyecto de una estrategia, ya que inicia con el listado de clientes y trabajadores del negocio y sus necesidades.

* 1 punto

- Modelo de caso de uso
- Modelo de dominio
- Modelo de análisis
- Modelo de negocio

ANALISIS DE REQUERIMIENTOS

Declaración de los servicios que se espera que el sistema proporcione y restricciones bajo las cuales debe funcionar * 1 punto

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Funciones, servicios y restricciones operativas del sistema. Define que es lo que se va a implementar. * 1 punto

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Son declaraciones de los servicios que debe proporcionar el sistema. * 1 punto

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Son restricciones de los servicios o funciones ofrecidos por el sistema. *

1 punto

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Se clasifican en requerimientos del producto, organizacionales y externos *

1 punto

- Requerimientos funcionales
- Requerimientos del usuario
- Requerimientos no funcionales
- Requerimientos del sistema

Base de datos

Sistema de Gestión de Base de Datos

Sistema de Gestión de Base ...



¿Qué es una base de datos? *

Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.

¿Qué es un sistema manejador de base de datos? *

Es una colección de software muy específico, orientado al manejo de base de datos, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.

Selecciona los mejores motores de base de datos *

1 punto

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

SGBD que incluye un motor mediante una interfaz gráfica, es usado por grandes compañías, además de ser un sistema muy robusto y seguro

* 1 punto

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

SGBD que permite soportar transacciones y gestiona procedimientos almacenados cliente servidor, fue realizado por Microsoft

* 1 punto

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

SGBD de código libre, es un sistema de acceso múltiple de datos en sus consultas y utiliza grandes extensiones de texto.

* 1 punto

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

SGBD de código libre, a diferencia que maneja una licencia. Es el sistema más utilizado * 1 punto debido a que se integra con lenguajes de programación php y HTML

- ORACLE
- ACCESS
- Microsoft SQLServer
- PostgreSQL
- MySQL

Este formulario se creó en INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA.

Google Formulario