Universidad de los Andes

Departamento de Ingeniería de Sistemas y Computación

Sistemas transaccionales

Documento de diseño iteración 1 - Proyecto

Profesor

Wilfredy Santamaria Ruiz

Grupo B6

Laura Sofia Murcia – 202123099

Sara Benavides Mora – 202022464

María Camila Martínez Martínez– 202116508

Bogotá - Colombia

Octubre del 2023

Documento de Análisis y Diseño Entrega 1

Para el desarrollo del proyecto se utilizó el caso Hotel de los Andes, con el objetivo de aprender a modelar correctamente las relaciones que se dan entre los clientes, servicios y otros autores que comprende el mundo del problema. Todo esto con el propósito de afianzar habilidades de modelado de datos relacional a partir del modelo conceptual propuesto. Adicionalmente, se afianzaron habilidades dentro del diseño e implementación de bases de datos.

Introducción

En este informe se encuentra toda la información pertinente para entender el funcionamiento de la aplicación realizada y sus funciones. El propósito principal de la aplicación construida es facilitar el procesamiento de los datos recibidos por el hotel para hacer más efectivo los procesos de recolección de información. Esto se hace por medio de la implementación y diseño de una base de datos y aplicación por capas.

Arquitectura de la Aplicación

La aplicación construida se realizó mediante la base de datos Oracle SQL Developer versión 23.1.0. Se utilizó Java 17 y Maven para el desarrollo de la aplicación.

El diseño de esta aplicación se centra en que el usuario pueda interactuar con la base de datos por medio de una interfaz html. Esto le permitirá al usuario un manejo fácil y práctico de los datos.

Diagrama de Secuencia

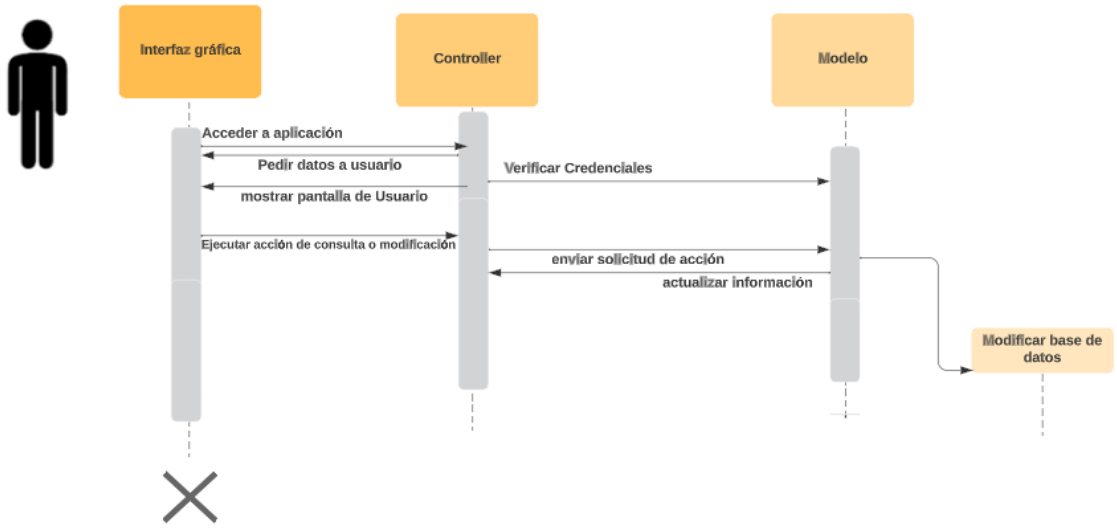


Figura 1. Diagrama de Secuencia Aplicación

En la figura 1 se puede ver el diagrama de secuencia que se tiene para esta aplicación. Como se muestra en este, se basa en interactuar con el usuario por medio de una interfaz. Esta interfaz dependiendo del tipo de usuario, le muestra las acciones que puede realizar el usuario y por medio de un controller se encarga de llevar la solicitud al modelo. Dentro del modelo, se generan los datos correspondientes dependiendo de la solicitud (sea esta de consulta o de

modificación). Finalmente, esta información se devuelve al controlador, y se actualiza la interfaz con la respuesta del sistema a la operación.

Restricciones del proyecto/Reglas de dominio

1. Los tipos de usuario posibles son: administrador, recepcionista, empleado y cliente.
2. El costo de alojamiento se define por noche.
3. Cada hotel cuenta con servicios en los que se pueden incluir: piscina, gimnasio, internet, bares, restaurantes, supermercado, tiendas, SPA, lavandería, préstamo de utensilios, salones de reuniones y salones de conferencia.
4. Para reservar un salón de conferencia se debe contar con un espacio de disponibilidad de 12 horas adicionales para efectos de limpieza.
5. Hay planes de consumo que incluyen: larga estadía, tiempo compartido, todo incluido y promociones particulares.
6. La entrada y salida de un cliente es gestionada por el recepcionista del hotel.
7. Los tipos de habitaciones incluyen: suite presidencial, suite, familiar, doble y sencilla.

Diseño de Base de Datos

Para el diseño de la base de datos de la aplicación, se generó un diagrama UML que permitiera generar una abstracción del problema general. Luego, a partir de esto se generó la conversión a un diagrama E/R. Esta conversión, se modeló conceptualmente haciendo uso de tablas de Excel en las que se pueda ver las restricciones, los atributos y un ejemplo de tupla para cada una de las relaciones creadas. Después de este proceso, se usó Oracle Data Modeler para generar un diagrama más concreto y poder obtener el diagrama final. A continuación, se pueden ver los diagramas mencionados.

Diagrama UML (desarrollado en genmymodel)

En este diagrama se encuentran todas las entidades identificadas junto con sus respectivas relaciones. En total, se identificaron 24 entidades con las cuales se modeló el problema.

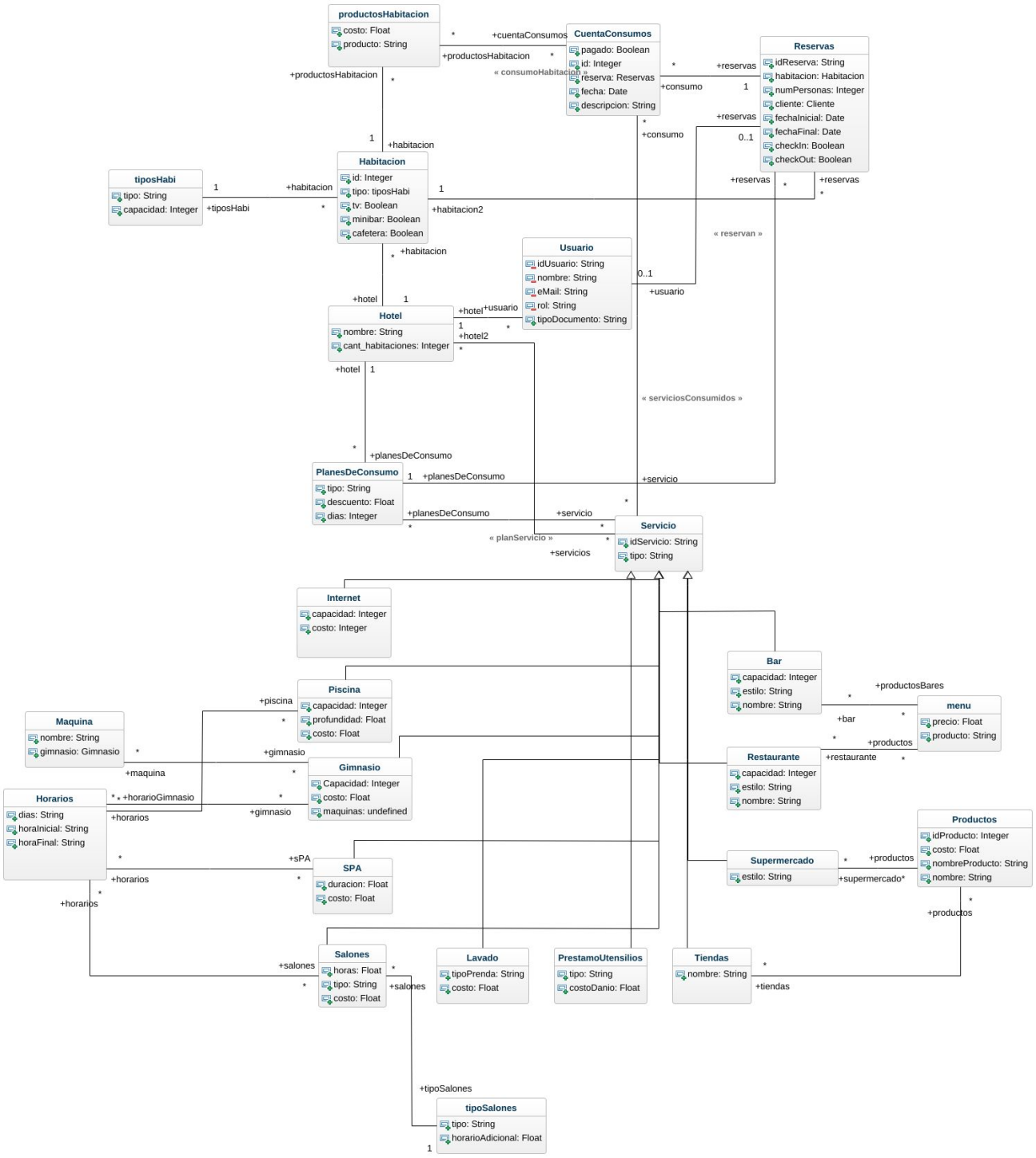
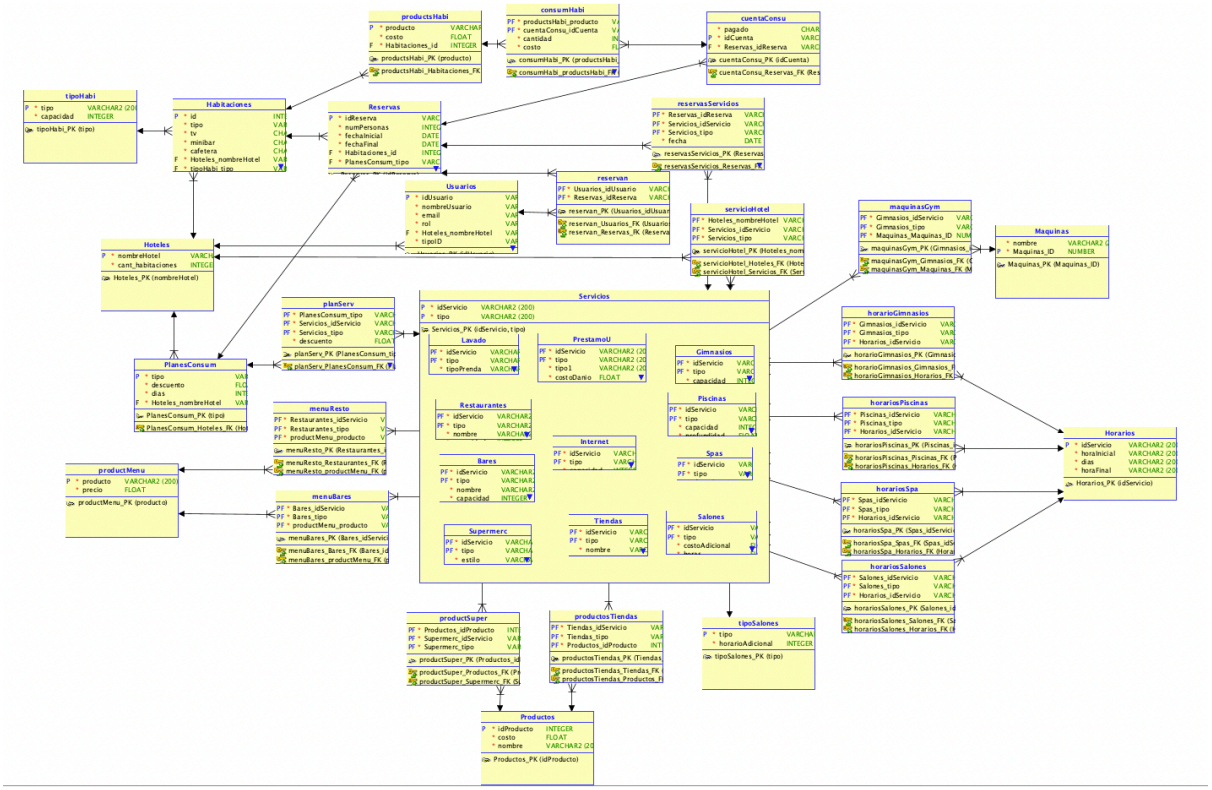
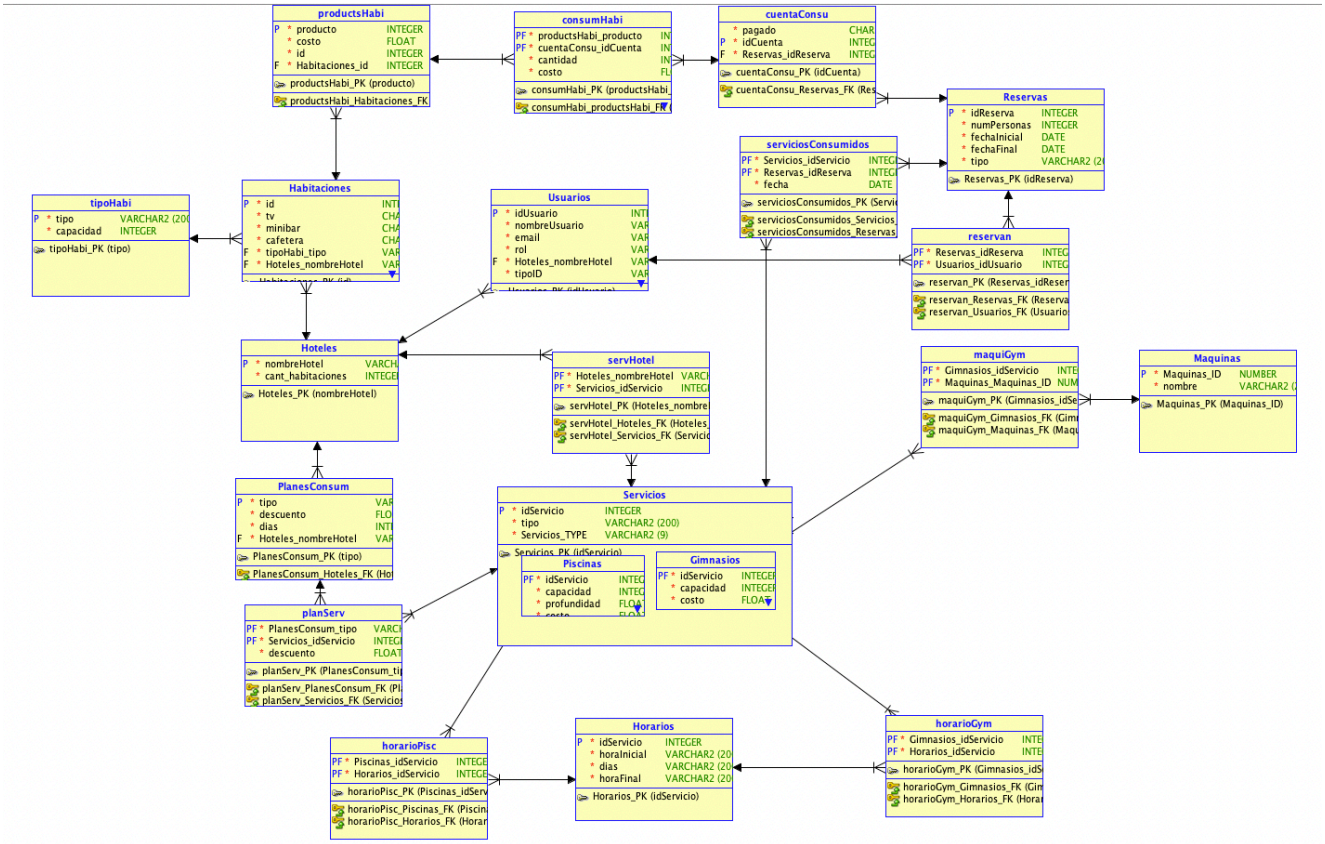


Diagrama E/R (diseñado en Oracle Modeler)

Se realizaron dos versiones para el E/R. La primera que se puede ver muestra el esquema completo de todas las relaciones. El segundo incluye solo algunos servicios y es el que se maneja en esta entrega dadas las especificaciones del enunciado.





Tablas

Hoteles

nombre	cant_habitaciones
PK, NN	NN
Tequendama	600

Normalización:

1NF: Lo cumple. La tabla Hoteles porque nombre es un varchar y cant_habitaciones es un integer.

2NF: Lo cumple. La tabla Hoteles tiene atributos totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Habitaciones

id	tipoHabitaciones	tv	minibar	cafetera
PK, NN	FK.tipoHabitaciones.tipo, CK	NN	NN	NN
1	suite	true	true	false

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos de la clase son independientes entre ellos.

3NF: Lo cumple. No hay transitividad entre los atributos.

Boyce-Codd: Los atributos de la clase no forman ciclos entre ellos.

tipoHabitaciones

tipo	capacidad
PK, NN	NN
suite	1

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos tipo y capacidad son independientes entre ellos.

3NF: Lo cumple. Tipo no implica capacidad ni capacidad implica tipo.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

productosHabitaciones

idHabi	producto	costo
PK, NN	PK, NN	NN
1	chocoramo	15000

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

ServiciosHotel

idServicio	hotel
PK, FK.Servicios.idServicio	NN
1	Tequendama

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos idServicio y hotel son independientes entre ellos.

3NF: Lo cumple. IdServicio no implica hotel ni viceversa.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Servicios

idServicio	tipo
PK, NN	NN, CK
1	piscina

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Piscinas

idServicio	capacidad	profundidad	costo
PK, FK.Servicios.idServicio	NN	NN	NN
1	20	1.6	0

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

HorariosPiscinas

idPiscina	dias	horaInicial	horaFinal
PK, FK.Piscinas.idServicio	NN	NN	NN
1	LMIJ	2:00 p.m	4:00 p.m

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Gimnasios

idServicio	capacidad	costo

PK, FK.Servicios.idServicio	NN	NN
2	40	0

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Maquinas

idGimnasio	nombre
PK, FK.Gimnasios.idServicio	NN
2	prensa

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

HorariosGimnasios

idGimnasio	dias	horaInicial	horaFinal
PK, FK.Gimnasios.idServicio	NN	NN	NN
2	VSD	8:00 a.m	4:00 p.m

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Spas

idServicio	duracion	costo
PK, FK.Servicios.idServicio	NN	NN
3	45	30000

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

HorariosSPAS

idSpa	dias	horaInicial	horaFinal
PK, FK.Spas.idServicio	NN	NN	NN
3	LMVSD	8:00 a.m	2:00 p.m

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos tipo y capacidad son independientes entre ellos.

3NF: Lo cumple. Tipo no implica capacidad ni capacidad implica tipo.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

HorariosSalones

idSalon	dias	horaInicial	horaFinal

PK, FK.Salones.idServicio	NN	NN	NN
4	LV	8:00 a.m	9:00 a.m

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

tipoSalones

tipo	horarioAdicional
PK	NN
conferencia	12

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Lavados

idServicio	tipoPrenda	costo
PK, FK.Servicios.idServicio	NN	NN
5	Camisa	10000

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Internet

idServicio	capacidad	costo
PK, FK.Servicios.idServicio	NN	NN
11	20	12000

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Tiendas

idServicio	nombre
PK, FK.Servicios.idServicio	NN
6	Bershka

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

ProductosTiendas

idTienda	idproducto	costo	nombre
PK, FK.Tienda.idServicio	PK	NN	NN
6	1	3400	camisa

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos de la clase son independientes entre ellos.

3NF: Lo cumple. No hay transitividad entre los atributos.

Boyce-Codd: Los atributos de la clase no forman ciclos entre ellos.

PrestamoUtensilios

idServicio	tipo	costoDanio
PK, FK.Servicios.idServicio	NN	NN
7	sarten	200000

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Supermercados

idServicio	nombre	estilo
PK, FK.Servicios.idServicio	NN	NN
8	Jumbo	Jumbo

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

ProductosSupermercados

idSupermercado	idproducto	costo	nombre
PK, FK.Supermercados.idServicio	PK	NN	NN
8	1	3400	salsa de tomate

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Restaurantes

idServicio	nombre	estilo	capacidad
PK, FK.Servicios.idServicio	NN	NN	NN
9	Di Lucca	italiano	100

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

ProductosRestaurantes

idRestaurante	nombre	costo
PK, FK.Restaurantes.idServicio	PK	NN
9	pasta	39000

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Bares

idServicio	nombre	estilo	capacidad
PK, FK.Servicios.idServicio	NN	NN	NN
10	BBC	moderno	80

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos de la clase son independientes entre ellos.

3NF: Lo cumple. No hay transitividad entre los atributos.

Boyce-Codd: Los atributos de la clase no forman ciclos entre ellos.

ProductosBares

idRestaurante	nombre	costo
PK, FK.Bares.idServicio	PK	NN
10	cerveza negra	12000

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Usuarios

idUsuario	nombre	email	rol

PK	NN	NN	CK, NN
mario123	Mario Perez	mario@gmail.com	cliente

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Reservan

idUsuario	idReserva
PK, FK.Usuarios.idUsuario	PK, FK.Reservas.idReserva
mario123	1

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Reservas

idReservas	habitacion	numPersonas	fechaInicial	fechaFinal	cuentaConsumos	planConsumos
PK	NN, FK.habitaciones.id	NN	NN	NN	NN,.FK.CuentaConsumo.id	NN,.FK.planesConsumo.tipo
1	1	1	29/09/2023	01/10/2023	1	todo incluido

Normalización:

1NF: Lo cumple. No hay ninguna tupla en ninguna posición de la tabla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

CuentaConsumos

id	reservas	pagado
PK	NN, FK.Reservas.idReserva	NN
1	1	false

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

consumosHabitaciones

idCuentaConsumos	productoConsumido	cantidad	costo
PK, FK.CuentaConsumos.id	PK, FK.productosHabitacion.idHabi	NN	NN
1	1	1	15000

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

planesConsumos

tipo	descuento	dias
PK, CK	NN	NN
Todo Incluido	10	7

Normalización:

1NF: Lo cumple. Ningún elemento de la tabla genera una tupla.

2NF: Lo cumple. Los atributos son totalmente independientes entre ellos.

3NF: Lo cumple. No existen relaciones de transitividad entre los atributos.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

planesServicios

tipoPlan	servicioConsumido	descuento
PK, FK.planesConsumo.tipo	PK, FK.servicios.idServicio	NN
todo incluido	9	100

Normalización:

1NF: Lo cumple. No se generan tuplas en ninguna posición de la tabla.

2NF: Lo cumple. No se generan dependencias entre las clases.

3NF: Lo cumple. Los atributos no tienen relaciones de transitividad entre ellas.

Boyce-Codd: Lo cumple, no se generan ciclos entre los atributos.

Requerimientos Funcionales

Nombre	RF1. Registrar / actualizar / borrar / consultar tipos de usuarios
Resumen	Dentro de este requerimiento, se maneja todo lo asociado a los tipos de usuario. Para esto, se generan acciones de registro, actualización, borrado y consulta. Esto lo hace el administrador.
Entradas	Parámetros para consulta.
Salidas	Retorno de la consulta realizada.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF2. Registrar / actualizar / borrar / consultar usuario
Resumen	Dentro de este requerimiento, se manejan las acciones asociadas a los usuarios. Estas incluyen registro, actualización, borrado y consultas. Esto lo debe hacer el administrador.
Entradas	Parámetros para consulta.
Salidas	Retorno de consulta.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF3. Registrar / actualizar / borrar / consultar tipo de habitación
Resumen	Se implementan las operaciones CRUD para los tipos de habitación que tienen ciertas características. Esta acción la realiza el administrador.
Entradas	Características de la habitación a buscar.
Salidas	Resultados de la búsqueda.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF4. Registrar / actualizar / borrar / consultar habitación
Resumen	Se realizan las consultas y operaciones CRUD sobre las instancias de las habitaciones de las cuales dispone el hotel. Realizado por el administrador.
Entradas	Parámetros para búsqueda de habitación.
Salidas	Resultados de búsqueda de habitación.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF5. Registrar / actualizar / borrar / consultar un servicio del hotel
Resumen	CRUD para encontrar los servicios del hotel. Se realiza por el administrador del hotel.
Entradas	Parámetro de búsqueda de servicios.
Salidas	Resultados de búsqueda de servicios.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF6. Registrar / actualizar / borrar / consultar un plan de consumo
Resumen	Se realizan las operaciones de planes de consumo dentro del hotel. Realizado por el administrador.
Entradas	Parámetros de búsqueda de los planes de consumo.
Salidas	Resultado de la búsqueda.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF7. Registrar / actualizar / borrar / consultar una reserva de alojamiento
Resumen	Operaciones de reservación de una habitación. La realiza un cliente.
Entradas	Parámetros de búsqueda de reservación de habitación.
Salidas	Resultado de la búsqueda
RNF asociados	Persistencia, privacidad.

Nombre	RF8. Registrar / actualizar / borrar / consultar una reserva de un servicio del hotel por parte de un cliente
Resumen	Operaciones de reservación de un servicio. La realiza un cliente.
Entradas	Parámetros de búsqueda de reservación de servicio.
Salidas	Resultado de la búsqueda
RNF asociados	Persistencia, privacidad.
Evaluación	Funciona parcialmente, en algunos casos no realiza la acción que debe. Falta de interfaz

Nombre	RF9. Registrar / actualizar / borrar / consultar la llegada de un cliente al hotel
Resumen	Operaciones de registro de llegada de un cliente al hotel. La realiza un recepcionista.
Entradas	Parámetros de búsqueda de registro de llegada de un cliente.
Salidas	Resultado de la búsqueda.
RNF asociados	Persistencia, privacidad.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Nombre	RF10. Registrar / actualizar / borrar / consultar un consumo de un servicio del hotel por parte de un cliente o sus acompañantes
Resumen	Operaciones CRUD para poder manejar y actualizar los consumos que tiene un cliente.
Entradas	Información del consumo.
Salidas	Actualiza la tabla de consumos y la cuenta a pagar.
RNF asociados	Persistencia.
Evaluación	Funcionamiento con algunas fallas en casos específicos, falta de interfaz.

Nombre	RF11. Registrar / actualizar / borrar / consultar la salida de un cliente
Resumen	Cambia los datos de salida de un cliente. Lo realiza el administrador del hotel.
Entradas	Ninguno
Salidas	Actualización del estado de la reserva.
RNF asociados	Persistencia.
Evaluación	Funcionamiento perfecto, falta de interfaz.

Requerimientos No Funcionales

Nombre	RNF1. Privacidad
Resumen	Los usuarios de HOTEL DE LOS ANDES solo pueden manipular y consultar la información que les es propia o a que tengan derecho en función de la definición de tipos de usuario correspondiente.
Evaluación	No se realizó el requerimiento

Nombre	RNF2. Persistencia
Resumen	La información manipulada por la aplicación debe ser persistente. Es decir, se deben actualizar todas las tablas que tengan información relacionada cada vez que se ejecute un cambio dentro de la aplicación.
Evaluación	Se cumple para todas las entidades del diseño.

Nombre	RNF3. Distribución
Resumen	La base de datos está centralizada.
Evaluación	Se cumple para todas las tablas.

Relaciones entre las clases

Las clases que se manejan se dividen en modelo, controlador, repositorio e interfaz. Dentro del modelo están todas las clases que se encargan de la creación de tablas con sus respectivas llaves. Dentro del controlador, se maneja el paso de solicitudes entre la interfaz y el modelo. Dentro del repositorio se generan las clases que determinan los métodos para modificar y realizar consultas sobre la base de datos. Finalmente, la interfaz maneja la parte visual del proyecto.

Objetivos Alcanzados

Se logró implementar el código para realizar las diferentes consultas conectándose a la base de datos. Asimismo, se especificó el diseño para el problema generando las respectivas relaciones para una abstracción adecuada. Se generaron las tablas dentro de SQL developer y el código para manejarlas adecuadamente.

Objetivos por Mejorar

La interfaz gráfica de la aplicación no se pudo culminar a cabalidad. Es por esto, que se debe mejorar este aspecto a futuro para tener una aplicación de más fácil interacción.

Pruebas

Con respecto a las pruebas, se revisó el comportamiento efectivo de los comandos dentro de SQL developer. Esto permitió ver que las sentencias eran adecuadas. Sin embargo, no se implementaron pruebas dentro de la aplicación.

Conclusión

Como conclusiones generales de este proyecto, se puede decir que se generó un diseño adecuado y completo para el problema. Adicionalmente, se realizaron las sentencias de consulta dentro de la aplicación y se declararon las entidades. Sin embargo, fallaron temas con respecto a la interfaz.