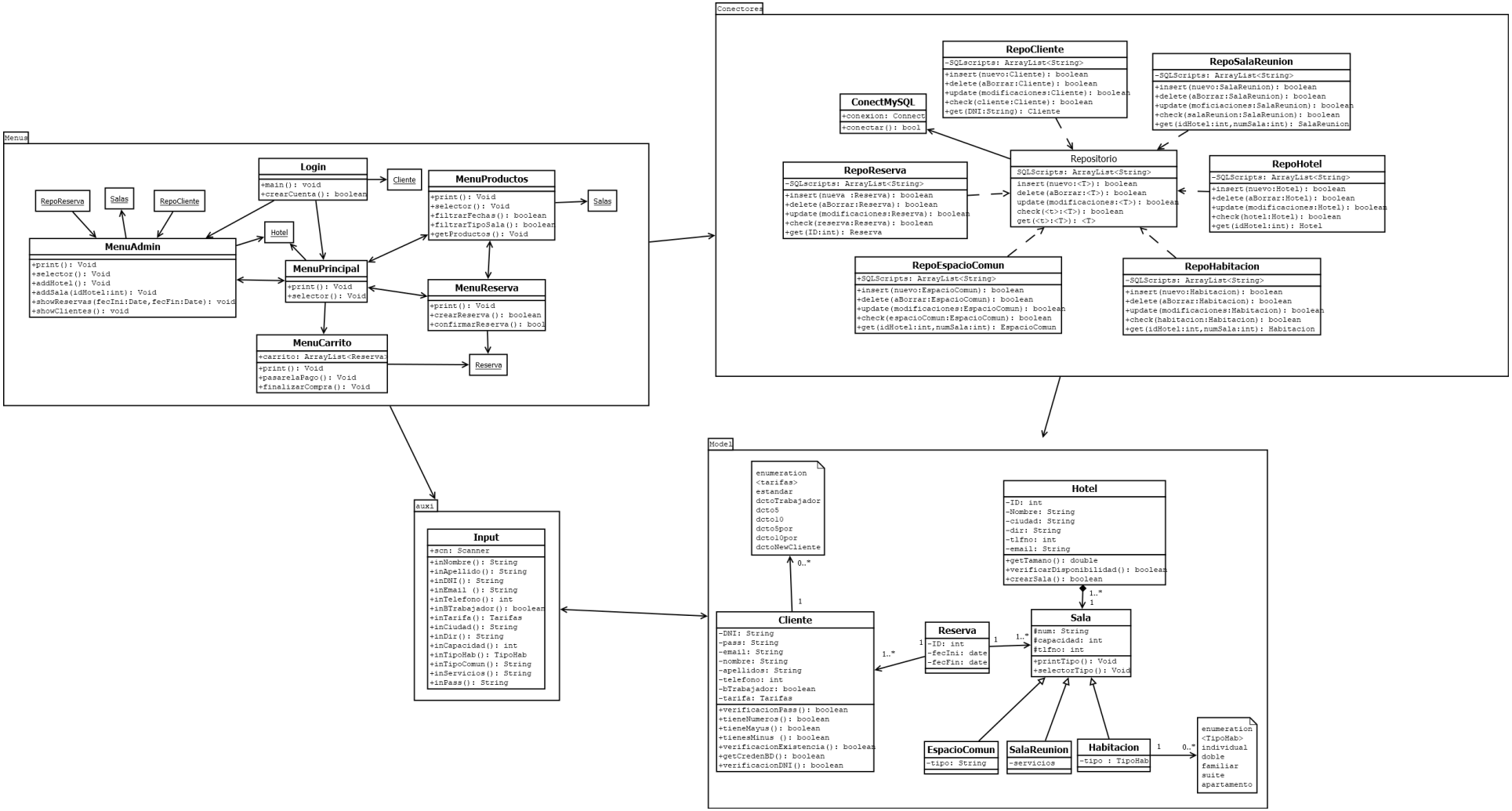
Documentación Hotel APP Reservas

Grupo APE

Contenido

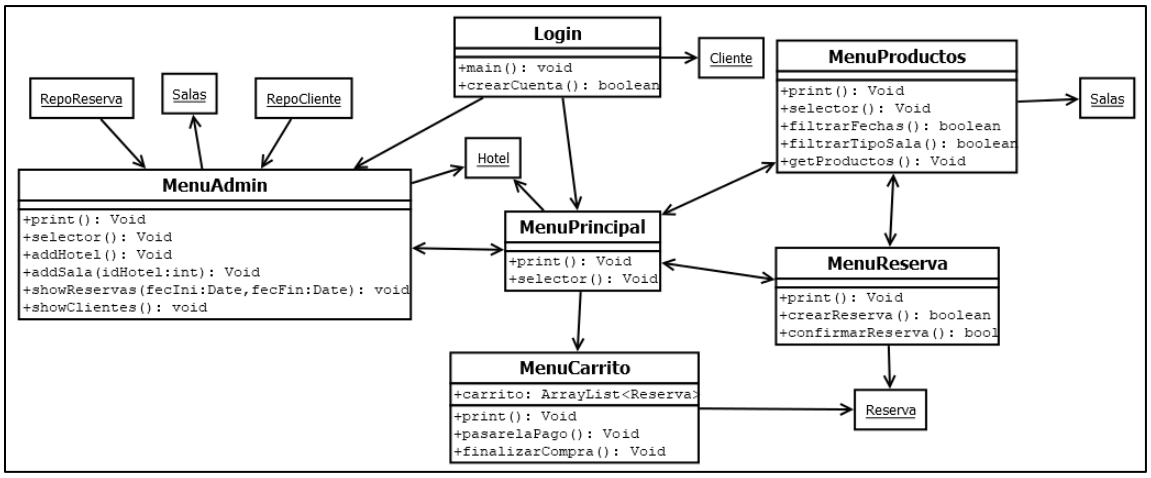
Documentación del Diagrama UML

Este documento describe el diseño del sistema representado en el diagrama de clases UML proporcionado. Se detalla cada componente del sistema, incluyendo las clases, atributos, métodos y relaciones, con el objetivo de proporcionar una visión clara y completa del diseño.



Paquete Menu

El paquete Menu contiene las clases responsables de gestionar las interacciones visuales del usuario con el sistema. Estas clases implementan menús específicos para administrar clientes, reservas, productos y otros elementos del sistema.



Clases del paquete

Login

Sirve para crear cuentas de cliente o trabajador y autentificar a los ya existentes.

Metodos

main() : void

Es la función principal de ejecución del programa en primera instancia. Esta función también contiene el control de acceso a la app y la conexión con la base de datos inicial.

crearCuenta() : boolean

Es una función que sirve para crear cuentas de cliente o trabajador en caso de no tener. Devolverá un boolean dependiendo de si la cuenta se crea de forma correcta o no.

MenuAdmin

Proporciona funcionalidades administrativas, como agregar hoteles y salas, además de mostrar reservas y clientes.

Metodos

print() : void

Esta función contiene todo lo que tiene que ver con imprimir el menú de administración.

selector() : void
En esta función se contendrá el selector del menú donde se ejecutará la operación de switch para elegir la opción.

addHotel() : void
Este método permitirá a los administradores crear hoteles y añadir salas (al menos una).

addSala(idHotel:int) : void
Este método servirá para añadir a un hotel ya existente salas, obligando siempre a añadir un tipo específico de sala.

showReservas(fecIni:Date , fecFin:Date) : void
Este método permitirá mostrar la reserva de habitaciones de todos los hoteles entre unas fechas determinadas.

showClientes() : void
Este método mostrara todos los clientes y trabajadores que se encuentren en la base de datos.

MenuPrincipal
Maneja la navegación principal del sistema entre los diferentes hoteles.

Metodos

print() : void
Esta función imprimirá por pantalla todos los hoteles y permitirá elegir entre ellos.

selector() : void
En esta función se contendrá el selector del menú donde se ejecutará la operación de switch para elegir la opción.

MenuProductos
Muestra las habitaciones del hotel y su disponibilidad más cercana.

Metodos

print() : void
Esta función imprimirá por pantalla todas las habitaciones del hotel y la fecha mas cercana disponible a partir del dia de hoy y permitirá elegir entre ellos.

selector() : void
En esta función se contendrá el selector del menú donde se ejecutará la operación de switch para elegir la opción.

filtrarFechas() : boolean

Este método modifica la consulta para hacer que las habitaciones mostradas solo sean aquellas que cumplan las condiciones de inicio y fin. En caso de que se pueda devolverá true y en caso contrario false.

filtrarTipoSala() : boolean

Este método modifica la consulta para solo mostrar el tipo de sala querido. En caso de que se pueda devolverá true, en caso contrario false.

getProductos() : void

Este método recogerá de cada hotel cuales son las salas disponibles. Y lo devolverá en pantalla junto a su fecha de disponibilidad mas cercana.

MenuReserva

Gestiona la creación y confirmación de reservas.

Metodos

print() : void

Este método imprime las características de la habitación a reservar.

crearReserva() : boolean

Pedira la información de la reserva y comprobara que está disponible en esas fechas. Devolvera false en caso de que no se pueda reservar en esas fechas.

confirmarReserva() : boolean

En caso que la reserva pueda ser creada y el cliente este de acuerdo, se grabara en la base de datos la información de la reserva. En caso de que por cualquier motivo no se pueda grabar devolverá false. Ademas, este método introducirá en la clase MenuCarrito.carrito la reserva.

MenuCarrito

Gestiona el carrito de reservas y el proceso de pago. Y finaliza la ejecución del programa.

Atributos

+carrito:ArrayList<Reserva>

Este atributo guardará todas las reservas hechas por el cliente durante la ejecución de la app y permitirá mostrar toda la información sobre ellas. DEBE DE SER PUBLICO Y ESTATICO.

Metodos

print() : void

Imprimira toda la información del usuario, las reservas que haya seleccionado y el precio de las mismas individualmente y de forma total.

pasarelaPago() : boolean

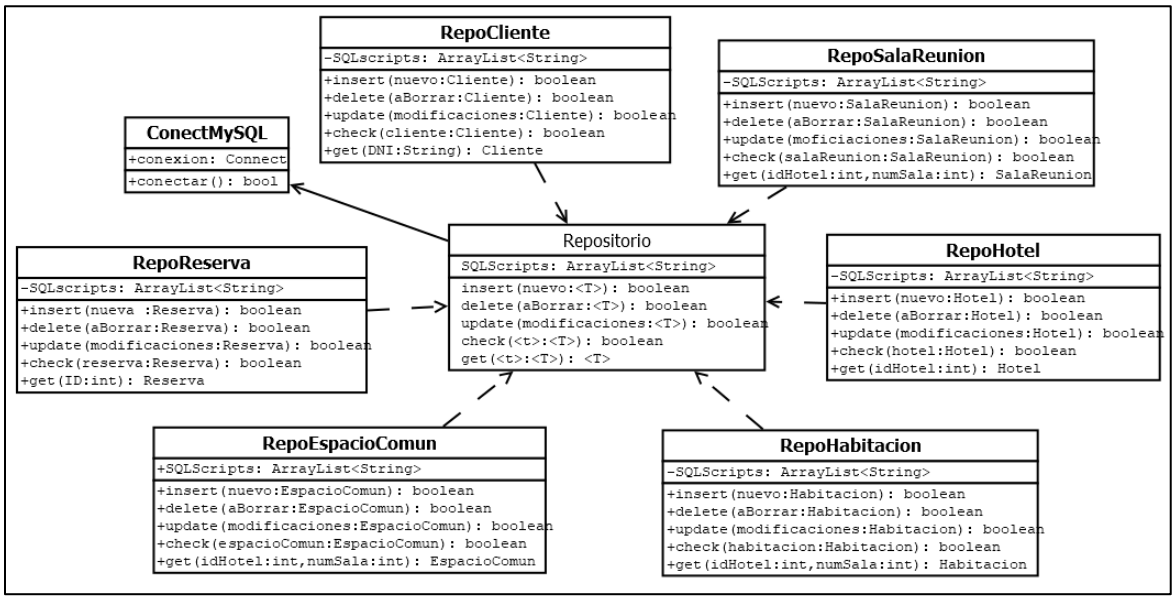
En caso de tener una reserva en el carrito permitirá introducir los datos de pago y finalizar la compra. En caso de que de error devolver false.

finalizarCompra() : void

Una vez la compra ha sido exitosa saldrá del programa, en caso de que el usuario no tenga nada en el carrito o el pago sea infructuoso, eliminara la reserva hecha y/o saldrá del programa.

Paquete Conectores

El paquete Conectores se encarga de la interacción con la base de datos mediante clases repositorio. Cada repositorio gestiona una entidad específica del sistema, proporcionando métodos para insertar, actualizar, eliminar y consultar datos (comprobación de existencia y descarga).



Clases del paquete

ConectMySQL

Clase responsable de gestionar la conexión a la base de datos.

Atributos

+conexión: Connect

Este atributo creara la conexión con la base de datos en función a los parámetros pertinentes según el usuario que se logee.

Metodos

+conectar() : bool

Este método creara la conexión con el servidor dependiendo del tipo de usuario que se conecte. Devolverá false en caso de que la conexión falle.

Repositorio y subclases

Proveen métodos para la manipulación de las entidades correspondientes en la base de datos.

Atributos

-SQLScripts : ArrayList<String>

Este atributo contendrá todos los scripts de sql asociados a la clase pertinente. Ordenados de esta forma:

Índice	Función
0	Insertar
1	Borrar
2	Modificar
3	Comprobar existencia
4	Traer la información
5 o superior	Otras funciones

Metodos

insert(nuevo:<T>) : boolean

Metodo que creara la secuencia de inserción en la base de datos y devolverá true o false dependiendo de si lo ha podido hacer.

delete(aBorrar:<T>) : boolean

Metodo que eliminara de la tabla correspondiente la información recibida y devolverá true o false dependiendo de si lo ha hecho o no.

update(modificaciones:<T>) : boolean

Recibira un parámetro de modificaciones el cual será un objeto con todos los valores por defecto excepto los que vayan a ser modificados. Cogera la clave principal y modificara los datos que no estén por defecto.

check(<t>:<T>) : boolean

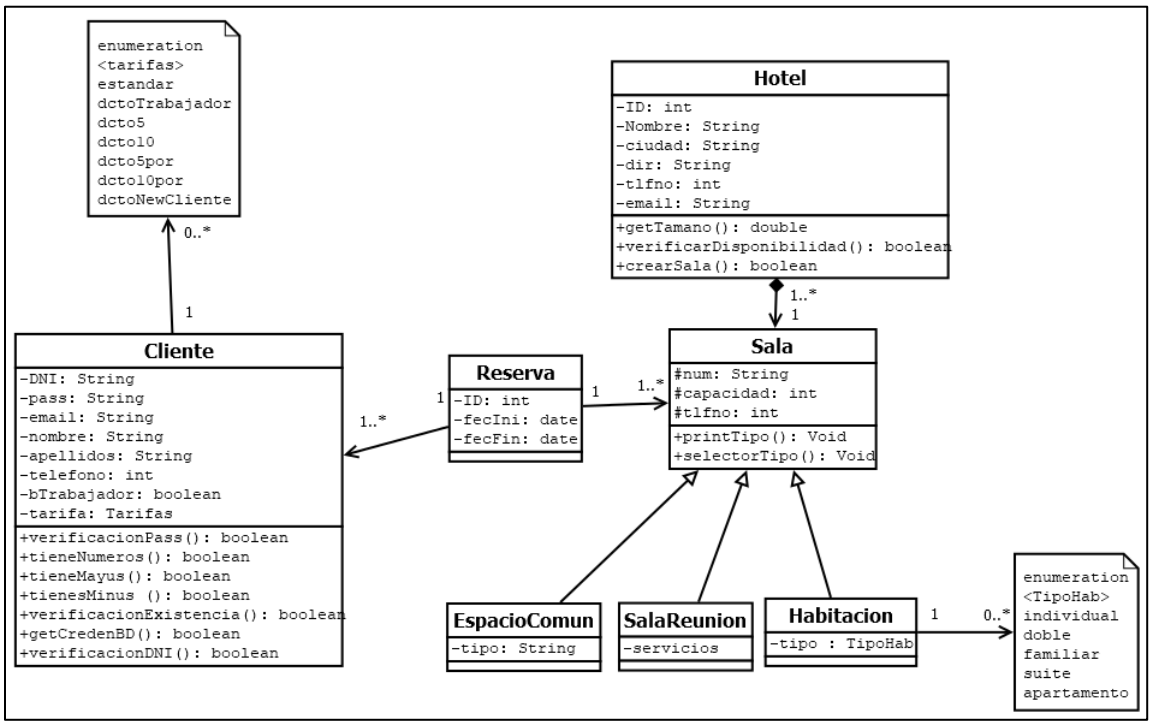
Comprobara la existencia del objeto recibido en la base de datos.

get(<t>:<T>) : <U>

Recuperara mediante la clave primaria el objeto de la base de datos, lo transformara en un objeto de la clase pertinente y lo devolverá.

Paquete Model

El paquete Model contiene las clases que representan las entidades principales del sistema y sus relaciones. Estas clases incluyen atributos y métodos específicos para modelar la lógica del negocio.



Clases del paquete

Cliente

Objeto que representa a los usuarios del sistema. Incluye atributos como DNI, nombre, email y métodos para validar datos y gestionar reservas.

Enumerados

Tarifas

Este enumerado contendrá todos los tipos de tarifas aplicables, tales como bonos de descuento por ser trabajador o descuentos por otros motivos. Por defecto todo el que lo use usara “estándar”.

Atributos

-DNI : String

CLAVE PRIMARIA del objeto cliente. Debera ser comprobado cada vez que sea incluido en el objeto como un valor correcto.

-pass: String
Contraseña de acceso a la app. Debera ser guarda de forma que se desconozca para el superadmin (hasheada).

-email: String
Información de contacto del cliente que debe ser mostrada en el proceso de compra, para un futuro servicio de telemarketing.

-nombre: String
Nombre del usuario identificado. Usado para cortesías y compra-venta.

-apellidos: String
Apellido/s del usuario identificado. Usado para cortesías y compra-venta.

-telefono: int
Información de contacto, se presupone que solo se podrán guardar números del estado español, es decir, de 9 digitos que no empiezan por 0.

-bTrabajador: boolean
Información que permite acceder al panel administrador, solo los superadmin podrán modificar este valor en la base de datos, de forma que el resto de usuarios solo podrán leerlo.

-tarifa: Tarifas
Contendra la información de tarifa aplicable en dicho momento al usuario. Ademas, da cabida a un futuro sistema de cupones.

Metodos

verificacionPass() : boolean
Comprobara que la contraseña es lo suficientemente segura para ser guardada.

tieneNumeros() : boolean
Comprobara que la contraseña tiene números.

tieneMayus() : boolean
Comprueba que la contraseña tiene mayúsculas.

tieneMinus() : boolean
Comprueba que la contraseña tiene minúsculas.

verificacionExistencia () : boolean
Se revisa en la BBDD que no exista otro usuario con el mismo email.

getCrendBD () : boolean
Revisa la información del cliente en la BD y devuelve el valor de 'bTrabajador'

verificacionDNI() : boolean
Comprueba que el DNI del objeto es correcto.

Hotel

Objeto que representa a los hoteles del sistema, con atributos como ID, nombre, ciudad y métodos para la creación de salas.

Atributos

-ID : int
CLAVE PRIMARIA es un numero que identifica el hotel.

-nombre: String
Nombre del hotel.

-ciudad: String
Ciudad del hotel

-dir: String
Direccion postal completa del hotel.

-tlfno: int
Telefono de contacto del hotel, teniendo en cuenta que todos los hoteles estarán en territorio español.

-email: String
Informacion de contacto vía telemática del hotel, podrá ser null.

Metodos

getTamano() : double
Metodo que devuelve la capacidad del hotel en número de personas.

verificarDisponibilidad () : boolean
Verifica en la base de datos si tiene alguna habitación libre en el día de hoy.

crearSala () : boolean
Crea una sala nueva, siendo siempre uno de sus subtipos.

Sala

Clase generalizada para modelar salas, que debe especializarse en subtipos. NO DEBE SER USADA COMO OBJETO.

Atributos

#num: String
CLAVE PRIMARIA de identificación de la sala junto al ID del hotel.

#capacidad: int
Numero de personas que pueden estar en dicha sala.

#tlfno: int
Extension de teléfono para contacto directo dentro del hotel.

Metodos

printTipo() : void
Imprime por pantalla las opciones disponibles de tipos de salas.

selectorTipo () : void
Selecciona entre los diferentes tipos de sala y muestra por pantalla el atributo añadido.

SalaReunion
Incluye atributos específicos como servicios. Que tiene un atributo llamado servicios, donde se guarda la información de los servicios disponibles en la sala de reuniones.

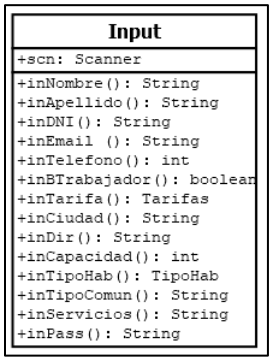
Habitacion
Contiene un enumerado que usa para identificar el modelo de habitación que es.

Incluye el atributo `tipo` para diferenciar entre individual, suite, etc.

EspacioComun
Representa espacios compartidos, con atributos como tipo (de formato abierto) para identificar que es (bar, recepción, piscina, etc).

Paquete Auxi

El paquete `Auxi` se encarga de gestionar aquellas funciones auxiliares como por ejemplo, la entrada de datos del usuario.



Clase Input

La clase `Input` incluye métodos para capturar información como nombres, fechas, tarifas, tipos de habitaciones, entre otros.