



**UNIVERSIDAD TECNOLÓGICA NACIONAL**  
**Facultad Regional Buenos Aires**

**GESTIÓN DE DATOS**

**1º CUATRIMESTRE - 2018**

**Trabajo Práctico Grupal: El monstruo del Lago Maser**

**Curso: K3013**

**Profesor: Juan Zaffaroni - Enrique Reinoso**

<b>Alumno</b>	<b>Legajo</b>
Fernando Martín Besteiro	155.897-3
Guillermo Perrone	157.275-1
Agustín Javier Lago	138.878-2
Álvaro De La Cruz	159.373-0

# Índice

Decisiones generales aplicadas a las tablas del nuevo modelo .....	4
Descripción de las tablas del nuevo modelo .....	4
Clientes .....	4
Cuentas .....	4
Usuarios .....	4
Tipos_de_documento .....	4
Items_factura .....	4
Facturas .....	5
Formas_de_pago .....	5
Consumos .....	5
ClientesXEstadias .....	5
Cierres_Temporales_Habitacion .....	5
Consumibles .....	5
Estadias .....	5
Habitaciones .....	5
ReservasXHabitaciones .....	6
Tipos_habitacion .....	6
UsuariosXRoles .....	6
Cancelaciones_Reserva .....	6
Reservas .....	6
Estados_reserva .....	6
UsuariosXHoteles .....	6
Países .....	6
Roles .....	7
RolesXFuncionalidades .....	7
Regímenes .....	7
HotelesXRegímenes .....	7
Hoteles .....	7
Funcionalidades .....	7
Cierres_Temporales_Hotel .....	7
Generacion_modificacion_reservas .....	7

Descripciones de los Stored Procedures, Funciones y Triggers .....	8
Procedures de migración .....	8
Procedures .....	8
Triggers.....	15
Detalle de índices del nuevo sistema .....	15
Detalle de vistas del nuevo sistema .....	16
Lógica general aplicada al nuevo sistema .....	16
Detalle de cada funcionalidad implementada .....	18
Migración .....	18
Log In .....	19
Detalles generales de los ABM .....	19
ABM Roles .....	20
ABM Usuarios .....	20
ABM Clientes .....	21
ABM Hotel .....	21
ABM Habitaciones .....	22
ABM Reservas.....	22
Ingreso y egreso de Estadías .....	23
Registrar consumibles .....	24
Facturación.....	24
Generar estadísticas.....	24

## Decisiones generales aplicadas a las tablas del nuevo modelo

Para la realización de este trabajo optamos por que todas las tablas de nuestro sistema cuenten con una única clave primaria (a excepción de aquellas que fueron generadas a causa de un "muchos a muchos"), todas ellas utilizando un identity, y su nombre sigue el patrón de "id\_" seguido del nombre de la tabla en singular. Esto nos facilita el seguimiento de las claves foráneas y el manejo general de las tablas.

## Descripción de las tablas del nuevo modelo

**Cientes:** Tabla que representa a los clientes (huéspedes) existentes en el sistema. El campo "correo\_cliente" es único por cada cliente y el campo "estado\_cliente" representa si este está habilitado o inhabilitado. En caso de no estarlo este no podrá realizar reservas. Dicho estado puede ser modificado por cualquier recepcionista.

**PK:** id\_cliente

**FK:** id\_tipo\_documento, id\_pais.

**Cuentas:** Tabla que posee los datos necesarios para el proceso de log in. Teniendo en ella tanto su username como su respectiva contraseña y guardando los intentos de logueo fallidos junto a estos. El username es único en todo el sistema.

**PK/FK:** id\_usuario

**Usuarios:** Tabla que tiene la información que detalla a cada uno de los usuarios. El campo "estado\_usuario" nos permite inhabilitar al susodicho en caso de una baja, convirtiendo a esta en solo una baja lógica.

**PK:** id\_usuario

**FK:** id\_tipo\_documento

**Tipos\_de\_documento:** Tabla que alberga los tipos de documento válidos teniendo uno extra que representa un tipo de documento desconocido.

**PK:** id\_tipo\_documento

**Items\_factura:** Tabla que representa cada renglón de una factura determinada teniendo en este la información de un consumo determinado.

**PK:** id\_item\_factura

**FK:** id\_consumo, id\_factura

**Facturas:** Tabla que posee un registro de todas las facturas del sistema guardando consigo su correspondiente total. El campo "detalle\_pago" es un campo agregado para que el cliente ponga información extra inherente a la factura tales como cuantas cuotas, el numero de su tarjeta, etc.

**PK:** id\_factura

**FK:** id\_estadia, id\_forma\_de\_pago

**Formas\_de\_pago:** Tabla que contiene todas las formas de pago aceptadas por nuestro sistema. Teniendo una forma extra que representa un medio de pago desconocido.

**PK:** id\_forma\_de\_pago

**Consumos:** Esta tabla representa un consumo de un renglón de una determinada factura, guardando también la fecha y la hora en que el consumible fue ordenado.

**PK:** id\_consumo

**FK:** id\_consumible, id\_estadia, id\_habitacion

**CientesXEstadias:** Tabla que guarda todas las estadías de un cliente y todos los clientes que estuvieron en dicha estadía. Sirve para representar el "muchos a muchos" entre Cientes y Estadías.

**PK/FK:** id\_cliente, id\_estadia

**Cierres Temporales Habitación:** Tabla que contiene los datos de la baja temporal de una habitación en concreto, guardando consigo tanto la fecha de inicio como la de fin y el motivo en particular de la baja momentánea.

**PK:** id\_cierre\_temporales\_habitacion

**FK:** id\_habitacion

**Consumibles:** Tabla que contiene la información de todos los consumibles que se pueden pedir guardando su precio y descripción.

**PK:** id\_consumible

**Estadias:** Tabla que contiene la información de todo tipo de estadías que intervengan en el sistema.

**PK:** id\_estadia

**FK:** id\_reserva, id\_usuario

**Habitaciones:** Tabla que contiene la información de todas las habitaciones (y las relaciona con un hotel). No puede haber dentro de un mismo hotel 2 habitaciones con el mismo número.

**PK:** id\_habitacion

**FK:** id\_hotel, id\_tipo\_habitacion

**ReservasXHabitaciones:** Tabla que contiene todas las reservas de una determinada habitación y las habitaciones de una determinada reserva. Sirve para representar el "muchos a muchos" entre Reservas y Habitaciones.

**PK/FK:** id\_reserva, id\_habitacion

**Tipos\_habitacion:** Tabla que contiene los tipos posibles que puede adoptar una habitación, guardando consigo cuantos huéspedes pueden entrar y su porcentual.

**PK:** id\_tipo\_habitacion

**UsuariosXRoles:** Tabla que contiene que roles tiene que usuario y que usuarios tienen un determinado rol. Sirve para representar el "muchos a muchos" entre Usuarios y Roles.

**PK/FK:** id\_rol, id\_usuario

**Cancelaciones\_Reserva:** Tabla que guarda los datos de las cancelaciones de reservas, guardando quien, cuando y porque realizo la cancelación.

**PK:** id\_cancelacion\_reserva

**FK:** id\_usuario

**Reservas:** Tabla que guarda la información pertinente a las reservas. Esta posee un campo de estado para poder determinar la situación en que se encuentran cada una de ellas.

**PK:** id\_reserva

**FK:** id\_usuario, id\_regimen, id\_estado\_reserva

**Estados\_reserva:** Tabla que guarda todos los posibles estados que pueda llegar a tener una reserva.

**PK:** id\_estado\_reserva

**UsuariosXHoteles:** Tabla que tiene que usuarios pertenecen a un determinado hotel y a que hoteles pertenece un usuario. Sirve para representar el "muchos a muchos" entre Usuarios y Hoteles.

**PK/FK:** id\_usuario, id\_hotel

**Países:** Tabla que contiene el nombre de todos los países del mundo.

**PK:** id\_pais

**Roles:** Tabla que contiene todos los roles posibles que pueden tener los usuarios en el sistema, teniendo también un campo de estado para saber si este está activo o no.

**PK:** id\_rol

**RolesXFuncionalidades:** Tabla que contiene las funcionalidades que tiene un determinado rol y que roles tiene una determinada funcionalidad. Sirve para representar el "muchos a muchos" entre roles y funcionalidades.

**PK/FK:** id\_rol, id\_funcionalidad

**Regimenes:** Tabla que contiene la información de los distintos regímenes aceptados por el sistema.

**PK:** id\_regimen

**HotelesXRegimenes:** Tabla que contiene que regímenes tiene un hotel y que hoteles tienen un determinado régimen. Sirve para representar el "muchos a muchos" entre Hoteles y Regímenes.

**PK/FK:** id\_hotel, id\_regimen

**Hoteles:** Tabla que sirve para registrar la información importante de todos los hoteles que alberga nuestro sistema.

**PK:** id\_hotel

**FK:** id\_pais

**Funcionalidades:** Tabla que contiene todas las funcionalidades que pueden llegar a tener los distintos roles de este.

**PK:** id\_funcionalidad

**Cierres\_Temporales\_Hotel:** Tabla que contiene los datos de la baja temporal de un hotel en concreto, guardando consigo tanto la fecha de inicio como la de fin y el motivo en particular de la baja momentánea.

**PK:** id\_cierre\_temporal\_hotel

**FK:** id\_hotel

**Generacion\_modificacion\_reservas:** Tabla que guarda los datos de que usuario modifico o genero alguna reserva en especifico. El campo tipo\_generacion solo puede valer "G" o "M" (no está definido ese chequeo en la tabla, pero esos son los únicos valores que le dejamos insertar al usuario).

**PK:** id\_generacion\_modificacion\_reserva

**FK:** id\_usuario, id\_reserva

# Descripciones de los Stored Procedures, Funciones y Triggers

## Procedures de migración

**(CREAR\_TIPOS\_HABITACION, CREAR\_RESERVAS, CREAR\_CONSUMIBLES, CREAR\_FACTURAS):**

Estos procedures se utilizan al momento de migrar los datos de la tabla maestra a las distintas tablas de nuestro sistema y estos insertan distintos datos (detallados en sus nombres) en la tabla correspondiente, haciendo que la id que tenía previamente encaje con la nueva que tendrá en el nuevo sistema.

## Procedures

**Aclaración:** Todos los procedures de alta, baja o modificación de algún dato del sistema reciben el id del usuario que quiere realizar la acción e invocan al procedure "VALIDAR\_ROL\_USUARIO " (Descrito más abajo) para validar que el usuario que está haciendo esa acción tenga la funcionalidad necesaria en su rol.

**OBTENER\_USUARIO\_DUMMY:** Procedure que devuelve el id del usuario Dummy. Necesario a la hora de que entre un cliente sin usuario.

**OBTENER\_ROL\_GUEST:** Procedure que devuelve el id del rol Guest. Este procedure es necesario cuando entra algún cliente sin usuario.

**OBTENER\_FUNCIONALIDADES:** Procedure que devuelve la información que hay en las tablas de funcionalidades del sistema. Este se usa en el ABM de roles, para poder cargar todas las funcionalidades disponibles.

**INSERTAR\_NUEVO\_ROL:** Procedure que, teniendo el nombre del nuevo role y una tabla con las funcionalidades que este tendrá, crea dicho Rol en la tabla de roles y agrega sus funcionalidades en RolesXFuncionalidades.

**OBTENER\_ROLES\_FILTRADOS:** Procedure que recibe un nombre y una funcionalidad, el cual filtra y devuelve los roles que cumplan con dicho filtro (o sea, que tengan esa funcionalidad, o que su nombre sea igual y/o contenga el nombre deseado). Usado principalmente tanto para la modificación como para la baja de roles. También posee un tercer campo el cual nos sirve para traer todos los roles o solo los que están activos que cumplan esa condición.

**OBTENER\_FUNCIONALIDADES\_DE\_ROL:** Procedure que recibe un id de un determinado rol y nos devuelve todas las funcionalidades que este posea (solo su id). Este se usa tanto para la modificación como para la baja de roles.



**MODIFICAR\_ROL:** Procedure que modifica un rol en base a un nombre, funcionalidades y estado. Este setea el nuevo nombre, sus nuevas funcionalidades y su nuevo estado y se usa para la modificación de un rol.

**DESHABILITAR\_ROL:** Procedure que recibe un rol y cambia el estado del mismo a deshabilitado. Este procedure se usa para la baja de roles.

**VALIDAR\_ROL\_USUARIO:** Procedure que valida que el rol activo del usuario tenga la funcionalidad que desea realizar. Esto nos permitirá tener un chequeo de seguridad extra, en caso de que por alguna razón algún usuario este usando una función que no le corresponda. En el caso de no cumplir con la funcionalidad pedida, utiliza RAISERROR para forzar la finalización de la ejecución.

**OBTENER\_HOTELES:** Procedure que devuelve todos los hoteles que hay en el sistema, junto a su correspondiente información.

**OBTENER\_ROLES:** Procedure que devuelve todos los roles que hay en el sistema, trayendo toda la información correspondiente.

**OBTENER\_ROLES\_DE\_UN\_USUARIO:** Procedure que recibe una id de un usuario y devuelve todos los roles que tenga el susodicho. Este procedure se usa para la ABM de usuarios.

**OBTENER\_HOTELES\_DE\_UN\_USUARIO:** Procedure que recibe una id de un usuario y devuelve todos los hoteles que tenga el susodicho. Este procedure se usa para la ABM de usuarios.

**OBTENER\_PAIS:** Procedure que recibe una id de un país y devuelve el nombre de dicho país.

**OBTENER\_TIPOS\_DOCUMENTO:** Procedure que devuelve toda la información que se encuentra en la tabla tipos\_documento.

**INSERTAR\_USUARIO:** Procedure que recibe los datos de un usuario e inserta este en la base de datos. Este proceso implica crear al nuevo usuario en la tabla de usuarios, insertar la información pertinente en usuariosXHoteles, usuariosXRoles y Cuentas (Gracias a que tenemos el id del nuevo usuario y algunos de los parámetros que recibe son el nombre de usuario, su contraseña y los hoteles y roles de este). El mismo es usado para la creación de un nuevo usuario.

**OBTENER\_USUARIOS\_FILTRADOS:** Procedure que recibe un nombre de usuario, nombre, apellido, rol, tipo y número de documento, hotel en el cual se desempeña y correo electrónico para así poder devolver un usuario que cumpla con alguno de esos filtros. También recibe un último campo el cual es un bit, permitiéndonos devolver la lista entera de usuarios que estén activos. Este procedure se usa tanto para la modificación como para la baja de usuarios.

**DESHABILITAR\_USUARIO:** Procedure que recibe el rol del usuario que está haciendo la baja y la id del usuario a deshabilitar. Este cambia el estado de el usuario en cuestión de habilitado a deshabilitado. Este procedure se usa para la baja de usuarios.

**MODIFICAR\_USUARIO:** Procedure que recibe el id del rol del usuario que hace la modificación, el id del usuario a modificar, su nombre de cuenta, contraseña, la lista de sus roles, la lista de hoteles, su nombre, apellido, tipo de documento, numero de documento, correo, teléfono, dirección, fecha de nacimiento y estado. Con estos campos actualiza la información del usuario en la tabla usuario, al igual que las otras tablas intervinientes (UsuariosXRoles, UsuariosXHoteles y Cuentas).

**LOGIN\_USUARIO:** Procedure que recibe un nombre de usuario y una contraseña y se encarga del logueo del mismo. En caso de que el usuario ingrese un nombre de usuario inexistente, una contraseña errónea o que la cuenta esta deshabilitada se mostrara el error correspondiente.

**OBTENER\_PAISES:** Procedure que devuelve toda la información de la tabla países. Este procedure se usa cada vez que sea necesario ingresar o modificar el país de algún cliente.

**INSERTAR\_NUEVO\_CLIENTE:** Procedure que recibe los datos de un cliente y luego inserta al nuevo cliente en la tabla clientes. Este procedure es usado a la hora de dar de alta un nuevo cliente.

**MODIFICAR\_CLIENTE:** Procedure que recibe los datos de un cliente y hace un update de ese cliente en específico en la tabla cliente. Usado en el proceso de modificación de un cliente.

**OBTENER\_CLIENTES\_FILTRADOS:** Procedure que recibe un nombre, un apellido un tipo y numero de documento, un correo y un estado de un cliente y según estos datos devuelve la información que haya en la tabla clientes según cualquiera de esos filtros. Este procedure se usa para mostrar y filtrar la lista de clientes en la modificación y borrado de clientes.

**DESHABILITAR\_CLIENTE:** Procedure que recibe el id de un cliente y cambia su estado a deshabilitado. Este procedure se usa para la baja de clientes.

**OBTENER\_REGIMENES:** Procedure que devuelve todos los regimenes que hay en el sistema. Este es usado para el ABM de hoteles.

**OBTENER\_HOTELES\_FILTRADOS:** Procedure que devuelve todos los hoteles que cumplan cierto filtro. Este puede ser por nombre, estrellas, ciudad o país. Este procedure se usa para la modificación o baja de hoteles.

**OBTENER\_REGIMENES\_DE\_UN\_HOTEL:** Procedure que recibe el id de un hotel y nos devuelve el id de los regímenes de este. Este procedure se usa para la modificación de hoteles.

**INSERTAR\_CIERRE\_TEMPORAL\_HOTEL:** Procedure que con la fecha de inicio y fin del cierre, el hotel y el motivo de cierre de este, agrega un nuevo cierre temporal a la tabla cierres\_temporales\_hotel. Este procedure se usa para el alta de cierres temporales de hoteles. Cabe aclarar que el motivo de cierre de un hotel nunca puede ser nulo. También este procedure puede tirar error en el caso de que se intente insertar otro cierre temporal para el mismo hotel que empiece en una fecha en la que

se supone que el susodicho está cerrado. (Por ejemplo, si mi hotel está cerrado del 03/05 al 06/05 no se puede agregar un cierre que interseque con ese mismo rango).

**MODIFICAR\_HOTEL:** Procedure que recibe los datos de un hotel y actualiza los mismos en la base de datos. Esto incluye también los regímenes del mismo. Este procedure se usa para la modificación de hoteles.

**INSERTAR\_CIERRE\_TEMPORAL\_HABITACION:** Procedure que recibe una fecha de inicio, una fecha de fin, una habitación y un motivo y con estos datos crea un nuevo cierre temporal de dicha habitación. Este procedure se usa para la baja de habitaciones. Cabe destacar que el motivo del cierre no puede ser null.

**OBTENER\_REGIMENES\_ACTIVOS\_HOTEL:** Procedure que recibe un hotel y devuelve todos sus regímenes habilitados. Este es usado para el ABM de reservas.

**CANCELAR\_RESERVAS\_VENCIDAS:** Procedure que recibe la fecha de hoy y con ella da de baja todas las reservas que estén vencidas. Una reserva esta vencida cuando el cliente no se presento en el primer día en que inicio esta. Este procedure se usa para la cancelación de reservas.

**OBTENER\_HABITACIONES\_DISPONIBLES\_RESERVA:** Procedure que recibe el intervalo de tiempo de la reserva a crear, la fecha de hoy y un id de reserva. Con estos datos el procedure ejecuta "CANCELAR\_RESERVAS\_VENCIDAS" Y luego se encarga de devolver las habitaciones que estén disponibles en ese periodo. En caso de estar cerrado el hotel en esas fechas el procedure nos devolverá una tabla con la habitación "-1" (Habitación dummy creada para representar al hotel cerrado). Cabe destacar que este procedure al ser usado tanto para la modificación como la creación de reservas el id de reserva que recibe cambia dependiendo en cuál de estas 2 funciones este siendo usado. Para el alta de reservas recibirá un -1 y dado que no existe ninguna reserva con ese id devolverá todas las habitaciones disponibles de esa reserva. Pero en caso de la modificación recibirá el id de la reserva a modificar, cosa que nos permitirá ignorar las habitaciones con esa id de reserva.

**OBTENER\_REGIMEN\_ALL\_INCLUSIVE:** Procedure que devuelve la id del regimen all inclusive. Este procedure se usa para la creación de una reserva.

**VALIDAR\_DOCUMENTO\_UNICO\_CLIENTE:** Procedure que recibe un tipo y numero de documento y valida que estos no sean un dato de alguno de los clientes del sistema. Este procedure es usado para la creación de un nuevo cliente.

**VALIDAR\_CORREO\_UNICO\_CLIENTE:** Procedure que recibe un correo y valida que este no sea un dato de alguno de los clientes del sistema. Este procedure es usado para la creación de un nuevo cliente.

**INSERTAR\_CLIENTE\_PREEXISTENTE:** Procedure que recibe los datos de un usuario que no se pudo insertar durante el proceso de migración y lo inserta en la tabla clientes. También por cada reserva que este cliente tenga se les modificara el id del cliente para que ahora tengan la del nuevo cliente insertado y por último se completa la tabla ClientesXEstadias con la información del nuevo cliente y sus estadias correspondientes.

**OBTENER\_CLIENTES\_COMPLETOS\_FILTRADOS:** Procedure que devuelve todos los clientes que están tanto en la tabla clientes como en la de errores\_migracion. Estos se pueden mostrar en su totalidad o filtrados por tipo y numero de documento y/o mail. Este se usa para poder agregar un cliente de la tabla errores\_migracion a la tabla clientes.

**INSERTAR\_RESERVA:** Procedure que pide una fecha de realización de reserva, una de inicio, una de fin, un cliente, un régimen y unas habitaciones y con estos datos crea una reserva nueva en la tabla reservas. Luego, este puebla la tabla ReservasXHabitaciones con el id de la reserva y sus correspondientes habitaciones y al final inserta un registro en la tabla generacion\_modificacion\_reservas. Este procedure se usa para la creación de reservas.

**INSERTAR\_NUEVO\_CLIENTE\_SIN\_VALIDACION:** Procedure que recibe los datos de un cliente y con estos genera un nuevo registro en la tabla de clientes. La particularidad de este procedure es que no se hace una verificación de si el usuario posee la funcionalidad "Dar de alta cliente" en su rol. Esto nos permite que un usuario "guest" pueda agregar un nuevo cliente al sistema cuando quiera hacer una reserva. Cabe recalcar que INSERTAR\_NUEVO\_CLIENTE utiliza internamente este procedure.

**OBTENER\_RESERVA:** Procedure que recibe el id de una reserva y devuelve los datos de dicha reserva si es que esta se puede modificar. Para eso este procedure se fija que la reserva en cuestión no tenga una estadía asociada y también, recibiendo la fecha del día de hoy, verifica si dicha reserva ya expiro o no (o sea que la fecha de inicio de la reserva es menor a la fecha de hoy). Este procedure se usa para el ABM de reservas.

**OBTENER\_TIPOS\_HABITACION\_RESERVA:** Procedure que recibe el id de una reserva y devuelve todos los tipos de habitación que intervienen en esta. Este procedure se usa para el ABM de reservas.

**OBTENER\_HOTEL\_DE\_RESERVA:** Procedure que recibe el id de una reserva y devuelve el hotel donde se ha hecho esta. Este procedure se usa para el ABM de reservas.

**MODIFICAR\_RESERVA:** Procedure que recibe el id del usuario que realiza la modificacion, el id del rol de dicho usuario, una fecha de inicio y una de fin de la reserva, el régimen de la reserva, la fecha de hoy y las habitaciones de la reserva. Con todos esos datos el procedure actualiza la información en la reserva dada en la tabla reservas y la info que hay en ReservasXHabitaciones. Además, inserta un nuevo registro en la tabla Generacion\_modificacion\_reservas para tener una constancia de quien y cuando modificó que reserva.

**OBTENER\_HABITACIONES\_DE\_RESERVA:** Procedure que recibe un id de una reserva y con ella devuelve todas las habitaciones de esta. Este procedure se usa para listar las habitaciones de las reservas.

**OBTENER\_RESERVA\_APTA\_ESTADIA:** Procedure que recibe la fecha de hoy, la id de una reserva y la id de un usuario y nos devuelve los datos de esa reserva en caso de ser apta para una estadía (ver Lógica general aplicada al nuevo sistema). En caso

de no serlo nos devolverá una reserva con id igual a -1. Este procedure es usado para agregar una estadía a una reserva.

**OBTENER\_ESTADIA\_DE\_RESERVA:** Procedure que recibe un id de reserva y devuelve la fecha de ingreso, de egreso y el id de la estadía relacionada a dicha reserva.

**OBTENER\_CLIENTE\_DE\_RESERVA:** Procedure que recibe el id de una reserva y devuelve el nombre, apellido, correo, id, tipo y numero de documento del cliente que hizo dicha reserva.

**INGRESAR\_ESTADIA:** Procedure que recibe un id de una reserva, un id de el usuario que ingresa la estadía, la fecha de ingreso de la estadía y los clientes que intervienen en esta. Con esta información crea una nueva estadía en la tabla estadías y con el id de la nueva estadía ingresada y los id de los clientes de dicha estadía puebla la tabla ClientesXEstadías. Finalmente, el procedure devuelve el id de la estadía ingresada. Este procedure se usa para ingresar una nueva estadía.

**EGRESAR\_ESTADIA:** Procedure que recibe una fecha de egreso de estadía, una id de estadía y el id del usuario que hizo el egreso. Con estos datos actualiza la estadía deseada en la tabla "estadías". Este procedure se usa para el egreso de estadías.

**OBTENER\_CLIENTES\_ESTADIA:** Procedure que recibe una id de una estadía y con ella devolverá el id, el mail, el nombre, el apellido, el tipo y numero de documento de todos los clientes involucrados con dicha estadía.

**OBTENER\_CANTIDAD\_RESERVAS\_EN\_PERIODO:** Procedure que recibe el inicio y fin de una reserva y el id de un hotel. Con estos datos nos dice cuantas reservas tiene dicho hotel en ese periodo determinado. Esto nos permitirá saber si se puede realizar un cierre temporal del hotel en ese periodo ya que un cierre temporal no puede realizarse si hay alguna reserva hecha en el periodo en que se quiere cerrar el hotel.

**HABITACION\_RESERVADA\_EN\_INTERVALO:** Procedure que recibe la fecha de inicio y fin de una reserva y el id de una habitación. Con estos datos el procedure devolverá 1 si es que la habitación tiene una reserva en ese intervalo o 0 en caso contrario. Este procedure nos ayudara a saber si podemos cerrar temporalmente dicha habitación en un intervalo. Recordemos que una habitación no se puede cerrar temporalmente si es que tiene una reserva en el intervalo en el cual se desea cerrar.

**REGIMENES\_USADOS\_POR\_RESERVAS\_DE\_HOTEL:** Procedure que recibe un id de hotel y la fecha de hoy y con esos datos devuelve la información de los regímenes que están siendo usados por las reservas de ese hotel. El procedure se asegura de que la reserva siga vigente y no se haya cancelado a la hora de traer esta información. Este procedure se usa para la modificación de regímenes. Recordemos que el programa no nos dejara modificar ningún régimen mientras que alguno de estos se encuentre en alguna reserva vigente.

**OBTENER\_ESTADIA:** Procedure que recibe el id de una estadía y devuelve la fecha de inicio y cierre de esta y si sus consumos están cerrados o no.

**OBTENER\_HABITACIONES\_DE\_ESTADIA:** Procedure que recibe el id de una estadía y devuelve las habitaciones que están relacionadas a esta.

**OBTENER\_CONSUMIBLES\_FILTRADOS:** Procedure que recibe una descripción y devuelve todos los consumibles que tengan una descripción remotamente parecida a la pasada por parámetro.

**AGREGAR\_CONSUMO:** Procedure que recibe un id de estadía, un id de consumible, un id de habitación, una fecha y una cantidad. Con estos datos inserta un nuevo consumo a la tabla de consumos. Este procedure se usa para ingresar un consumible a una estadía.

**MODIFICAR\_CONSUMO:** Procedure que recibe un id de consumible, un id de consumo, un id de habitación, una fecha y una cantidad y con estos datos modifica el consumo deseado. Este procedure se usa para la modificación de un consumo.

**OBTENER\_CONSUMOS\_DE\_ESTADIA:** Procedure que recibe un id de estadía y devuelve cuando, cuanto salió y que consumibles se consumieron en su correspondiente habitación que pertenecen a dicha estadía. Este procedure nos permitirá mostrar los consumibles consumidos en determinadas estadías.

**BORRAR\_CONSUMO:** Procedure que recibe un id de un consumo y lo borra de nuestra base de datos. Este procedure se usa para borrar un consumo.

**CERRAR\_CONSUMO\_ESTADIA:** Procedure que recibe una estadía y cierra el consumo de la misma. Esto significa que ya no se podrán agregar más consumibles a esta.

**OBTENER\_DATOS\_ESTADIA\_FACTURACION:** Procedure que recibe un id de estadía y devuelve la fecha de inicio y fin de esta junto a la fecha de inicio, la de fin y el id del régimen de la reserva asociada. Estos datos solo son traídos si la estadía en cuestión tiene sus consumos cerrados (dicho campo en 1), su fecha de egreso no es nula y esta reserva no tiene factura asociada. Este procedure es usado para la facturación.

**CALCULAR\_COSTO\_DIARIO\_ESTADIA:** Procedure que recibe un id de una estadía y calcula el costo diario de la misma (ver sección de Lógica general aplicada al nuevo sistema). Este procedure se usa para la facturación.

**OBTENER\_REGIMEN\_ALL\_INCLUSIVE\_MODERADO:** Procedure que devuelve el id del régimen all inclusive moderado.

**OBTENER\_REGIMEN\_DE\_ESTADIA:** Procedure que devuelve la información de la estadía relacionada al id de estadía recibido.

**OBTENER\_FORMAS\_DE\_PAGO:** Procedure que devuelve toda la información que haya en la tabla formas de pago.

**CREAR\_FACTURA:** Procedure que recibe una fecha de facturación, un total, un id de estadía, un id de forma de pago, un detalle de pago y una lista con los ítems de la factura. Con estos datos crear una nueva factura en la tabla factura. Luego ingresa un

nuevo ítem de factura por cada ítem de la lista pasada y finalmente devuelve el id de factura recién ingresada. Este procedure se usa para la facturación.

**TOP5\_HOTELES\_RESERVAS\_CANCELADAS:** Procedure que recibe una fecha de inicio y una de fin y devuelve los 5 hoteles con mayor cantidad de reservas canceladas dentro de ese periodo.

**TOP5\_HOTELES\_MAYOR\_CANTIDAD\_CONSUMOS:** Procedure que recibe una fecha de inicio y una de fin y devuelve los 5 hoteles con mayor cantidad de consumibles facturados en ese periodo.

**TOP5\_HABITACIONES\_MAYOR\_CANTIDAD\_DIAS\_ESTADIAS:** Procedure que recibe una fecha de inicio, una de fin y el día de hoy y con esto devuelve las 5 habitación con mayor cantidad de días y veces ocupadas. El día de hoy es necesario para considerar las estadías todavía en curso (es decir, las que no tienen fecha de egreso).

**TOP5\_CLIENTES\_PUNTOS:** Procedure que recibe una fecha de inicio, una de fin y el día de hoy y con esto devuelve los 5 clientes con mayor cantidad de puntos. (el cálculo de puntos se encuentra en el enunciado).

## Triggers

**Add\_users\_for\_new\_hotel:** Trigger que se activa al insertar un nuevo hotel al sistema. Este le agrega al usuario dummy y al administrador el nuevo hotel que fue insertado, permitiéndole a estos usuarios tener al nuevo hotel como uno de los hoteles donde él se desempeña.

## Detalle de índices del nuevo sistema

Elegimos las tablas "clientes" y "usuarios" para tener índices debido a que son las tablas con mayor volumen de datos en todo el sistema. También tuvimos en cuenta para esta decisión que son tablas en las cuales estaremos casi constantemente buscando, modificando e insertando información.

**Índices en la tabla clientes:** Cliente\_nombre y Cliente\_apellido.

Estos índices fueron creados para una búsqueda mucho mas eficiente de los clientes, dado al tamaño de dichas tablas. Estos filtros son vitales a la hora de filtrar rápidamente a los clientes. Cabe destacar que no se ha hecho un índice por correo debido a que dicho valor ya es único.

**Índices en la tabla de reservas:** Fecha\_inicio\_reserva y Fecha\_inicio\_fin\_reserva.

El primer índice es solo por la fecha de inicio del sistema. Este índice nos facilita el recorrer todas las reservas de nuestro sistema a la hora de saber si esa reserva ya expiro (para lo cual solo necesitaremos su fecha de inicio).

El segundo es por las fechas de inicio y fin de la reserva. A la hora de hacer una reserva se necesitan las habitaciones disponibles. Y para saber cuáles son, hay que recorrer toda la tabla de reserva en busca de cuales están disponibles en un determinado intervalo. En síntesis, este índice mejora el rendimiento del proceso de reservar una habitación.

## Detalle de vistas del nuevo sistema

**clientes\_completos:** Vista que nos muestra tanto a los clientes que están "oficialmente" en el sistema como a los que no se los agrego durante la migración debido a algún problema. Esta vista nos ayudara a pasar a esos clientes que tuvieron problemas a la tabla clientes si es que alguno de ellos intenta hacer una reserva nuevamente en nuestro nuevo sistema.

## Lógica general aplicada al nuevo sistema

- La aplicación se puede configurar con el archivo application.properties (admite parámetros de BBDD y de fecha del día de "hoy").
- Siempre que se pida un correo se verificara que este tenga la forma de "Algo@algo.com".
- El usuario "dummy" será el usuario por defecto de cualquier persona que ingrese al sistema sin loguearse, permitiéndole a este poder hacer una reserva en cualquier hotel.
- Existe un archivo .log el cual guarda la información de cualquier excepción que genere la aplicación.
- La cuenta que seguimos para calcular lo que vale una reserva en una habitación es la siguiente:

$$\text{Precio habitación por día} = \text{Precio base} * \text{huéspedes} * (1 + \text{porcentual}/100) + \text{estrellas} * \text{recarga Por Estrellas}.$$

- A la hora de hacer reservas, estas deberán ser de al menos un día.
- Cuando un cliente que no se había podido insertar durante el proceso de migración corrige sus datos este se inserta en el sistema, luego se inserta en ClientesXEstadias las estadías que faltaron ingresar durante la migración debido a la inexistencia de este cliente en el sistema.
- A la hora de modelar las reservas, observamos que el enunciado nos pide almacenar los tipos de habitación requeridos.



Por otro lado, también notamos, que los números de habitación se ingresarían al finalizar la estadía.

Con esto, llegamos a la conclusión que, si almacenamos simplemente el tipo de habitación, al querer intentar guardar un cierre temporal de habitación, no habría forma de saber si esta está en uso, debido a que solo sabríamos su número tras finalizar la estadía.

Por esto, decidimos almacenar en ReservasXHabitaciones las habitaciones directamente (que cumplen con los tipos de habitaciones solicitados por los clientes, y correspondientes a la reserva), de forma tal que ahora sí se podrá validar cuales habitaciones estarán reservadas o no (para evitar que se puedan cerrar). De ser necesario saber los tipos de habitación relacionados, simplemente se debería ver el tipo de habitación de la habitación (id\_tipo\_habitacion).

De todas formas, al finalizar la estadía, el usuario podrá visualizar las habitaciones "ocupadas" de la reserva y, debido a esto, cuando el usuario quiera registrar un fin de estadía no tendrá que ingresar las habitaciones donde estuvo debido a que el sistema las tiene almacenadas.

- Consideramos que las reservas que podrán tener estadías son las que tengan el estado en nuevo, modificado o desconocido, la fecha de inicio de la reserva sea igual al día de hoy y que la id de estadía este en null (o sea, que la reserva no tenga ya una estadía).
- Consideramos que el régimen "All inclusive moderado" te hará un descuento del 50% sobre todos los consumibles que pidas durante tu estadía ya que el enunciado especifica que existe este y el all inclusive normal pero nunca detalla que tienen de diferente.
- Consideramos que una estadía sigue abierta mientras que su fecha de egreso este en null.
- Una vez cerrada la estadía no se podrán añadir más consumibles pero si modificar o borrar los que ya estaban.
- Puede darse el caso de que suceda una excepción no controlada durante la ejecución de la aplicación. En ese caso es posible que la aplicación se minimice y que el IDE requiera de un click en continuar con la ejecución. Una vez presionado dicho botón la aplicación mostrará un cartel pero se podrá seguir la ejecución normalmente.
- Debido a las posibles inconsistencias de la tabla maestra (cuando el total de la factura es distinto a la sumatoria del precio de todos sus consumibles), decidimos desnormalizar el total de la factura para poder almacenar los valores.
- Una vez hecha una facturación en el directorio src se creará la carpeta "facturas" la cual contendrá las facturas que se van generando por el nuevo sistema con el número de factura como nombre. (en archivos .txt).
- A la hora de calcular el top 5 del punto 13, cualquier fecha que esté en null no será contada para ninguno de los cálculos (por ejemplo, si un cliente tiene la

fecha de consumo en null, pero la cantidad de este no lo es igualmente no se lo tendrá en cuenta debido a que no sabemos si ese consumo fue hecho durante el trimestre deseado).

- Para los temas relacionados a estadísticas, en el caso de haber una estadística que corresponda cierto valor a un trimestre, y cierto valor a otro (debido a que es una estadía que empieza en un trimestre y termina en otra, por ej.) los puntos se contarán correctamente, pues cada trimestre tendrá el valor que le corresponde.
- Para el cálculo de puntos del cliente, solo se le otorgarán puntos por los días alojados.

## Detalle de cada funcionalidad implementada

### Migración

Para realizar esta tomamos las siguientes consideraciones:

- Crear el Esquema a utilizar y luego las nuevas tablas que habíamos diseñado previamente. En esta primera instancia no asignamos las claves foráneas ya que eso nos permitirá una migración más sencilla.
- Poblar las tablas que no tengan ningún dato que venga del esquema maestro que nos suministraron. Nos referimos a poblar las tablas "Países", "EstadosReserva", "FormasDePago", "Funcionalidades", "Roles", "RolesX Funcionalidades" y "TiposDeDocumentación" con la información que queremos que soporte el sistema; Por ejemplo, en la tabla países la poblamos con todos los países del mundo, la de formas de pago con los medios de pago que queremos que soporte el nuevo sistema, etc.
- Crear un usuario Dummy. Este usuario le permitirá a cualquier visitante sin cuenta poder hacer, modificar o borrar reservas.
- Crear un cliente Dummy con id = 0. Debido a que ingresamos las reservas primero y debemos pasarles un id (y teniendo en cuenta que este no puede ser nulo) decidimos crear a este cliente para asignárselo a las reservas temporalmente.

Existen casos en la migración donde cliente no se insertará, en ese caso la reserva quedará con id\_usuario = 0 y la estadía relacionada tampoco sería insertada. Esto indica que el cliente no se recuperó del sistema antiguo. Además, este cliente Dummy es necesario para que la clave foránea relacionada con clientes y reservas no tenga errores.

- Crear al usuario Administrador general junto a su respectiva cuenta de admin. Este usuario tendrá todas las funcionalidades del sistema

- Migrar aquellas tablas que ya tenían un id mediante procedures los cuales las insertan en bloques usando "INSERT INTO" en la tabla correspondiente. Esto nos permitió asegurarnos de que cada nueva id empezara desde el mismo número que el dato original (Por ejemplo, id\_tipo\_habitacion arranca desde el 1001 al igual que lo hace Habitacion\_tipo\_codigo) para que no haya discrepancias. Para poder ingresarlos estos tienen que venir ordenados y sin ningún valor faltante, al ser este el caso que cumple el esquema maestro decidimos ir por esta estrategia para poblar algunas tablas.
- Creamos una tabla temporal que tiene el contenido de la tabla maestra, y la vamos utilizando para hacer múltiples INSERT INTO SELECT, mientras que la vamos modificando para facilitar las siguientes inserciones. Luego, a la hora de insertar clientes, para evitar fallos por las claves duplicadas creamos una nueva tabla temporal con índices únicos con IGNORE\_DUP\_KEY, para que posteriormente se inserten en la tabla de clientes.
- Finalmente modificamos las tablas que ya fueron creadas y pobladas, agregándoles las claves foráneas correspondientes.
- Cabe aclarar que se crearan índices solo para la tabla de clientes debido a que esta es la que tiene una mayor cantidad de datos que las demás y la ventaja de estos se aprovecha al máximo en este caso al contrario de que si hiciéramos uno en la tabla de roles por ejemplo. Estos serán por nombre, apellido y correo.
- Los clientes que hayan dado error al a hora de migrar estarán guardados en una tabla especial llamada "migración\_errores" junto a todos sus datos y dos extra. Uno que es el mensaje de error que se produjo al intentar migrarlo y el segundo que es el id de la estadía que le pertenece a dicho cliente. Cuando alguno de estos clientes desee realizar alguna reserva en nuestro nuevo sistema el mismo le pedirá que actualice su información para poder ingresarlo al sistema (ya que el sistema no conoce la información del cliente y por eso es necesario que sea este el que la modifique).

## Log In

Al querer iniciar sesión se solicitará el nombre de usuario y la contraseña. Si se validan correctamente los datos se procede a la selección de uno de los roles y hoteles disponibles según sus permisos. Los motivos de fallo en la validación serian inexistencia del usuario, contraseña incorrecta, o intento de logueo con un usuario inhabilitado. En el caso de fallar tres veces la validación, la cuenta de usuario quedará bloqueada hasta que un administrador la habilite nuevamente.

## Detalles generales de los ABM

En general todos los ABM van a estar distribuidos de una forma parecida: Para la alta aparecerá una ventana que nos pedirá los datos necesarios para realizar esta acción. Para la modificación primero habrá que pasar por una ventana que nos mostrara los datos existentes en la base de datos, posteriormente a apretar el botón "modificar" se pasará a la misma ventana que para agregar un dato solo que este nos permitirá modificar esta vez. Cabe destacar que siempre habrá un checkbox el cual nos permitirá cambiar el estado de deshabilitado a habilitado en caso de ser posible (véase en roles o usuarios). Para la baja se pasará por el mismo listado que para la modificación simplemente que esta vez con apretar el botón "borrar" este será borrado y no volverá a aparecer en esta lista. Se puede observar que al ser una baja lógica este dato "borrado" si aparecerá en el listado de datos de la ventana de modificación como deshabilitado.

## ABM Roles

La ventana de creación y modificación de roles nos muestra 2 listados: A la izquierda el listado con las funcionalidades existentes en el rol y a la derecha las funcionalidades disponibles. El usuario escribirá el nombre del nuevo rol en el text box de arriba y seleccionara las funcionalidades deseadas, agregándolas al presionar el botón "<". Finalmente, al presionar "agregar nuevo rol" se confirmará la acción y se ejecutará creando en nuestra base de datos el nuevo rol en la tabla de Roles y agregando sus funcionalidades en la tabla RolesXFuncionalidades. Si el usuario desea modificar algún rol le aparecerán los roles existentes y al apretar "modificar" lo llevará a la misma ventana de creación de rol solo que en este caso solo podrá agregar o sacar funcionalidades a dicho rol. Por último la ventana de borrado nos permitirá dar una baja lógica de un rol en específico con tan solo 2 clicks, en su proceso nos mostrara también las funcionalidades que este posee.

## ABM Usuarios

La ventana de creación de usuario nos permite setear su contraseña, nombre de usuario, nombre, apellido, año de nacimiento (desplegando un calendario a la hora de elegir dicha fecha y no permitiendo que se ingresen fechas futuras a la actual), su tipo y número de documento (desplegando una lista mostrando los tipos de documento soportados por el sistema), su correo, sus roles y los hoteles donde se desempeñará (mostrando en el caso de estos últimos una lista con los roles y hoteles disponibles y a su izquierda los que tendrá el nuevo usuario). Finalmente, clickeando el botón "agregar nuevo usuario", se creará al usuario con las características pedidas.

La ventana de modificación nos mostrará una lista donde se encuentran todos los usuarios del sistema. Mediante esta ventana podremos filtrar la lista de usuarios por nombre de usuario, nombre, apellido, rol, tipo y número de documento, hotel en el cual se desempeña y correo electrónico. Una vez elegido el usuario, a este se le podrá modificar cualquiera de sus campos.

En la ventana de borrado mostrará la misma lista con los mismos filtros que en la de modificación, solo que esta vez se podrá borrar al usuario del sistema (esta será una baja lógica, cambiando el estado del usuario).

## ABM Clientes

La ventana de creación de clientes nos pedirá un nombre, un apellido, un tipo y número de documento (El tipo lo podremos elegir de una pequeña lista donde se mostraran los que acepta el sistema), un mail, un teléfono, una fecha de nacimiento (la cual elegiremos de un calendario que nos dejará elegir cualquier fecha que no pase la actual), una nacionalidad, una calle, número, piso, departamento, una ciudad y un país (los cuales también se seleccionara de una lista con todos los países del mundo, dándonos la oportunidad de seleccionar un país manualmente). Con todos estos datos al apretar el botón de registrar se creará el nuevo cliente.

La ventana de modificación nos mostrara una lista con los primeros 100 clientes que cumplan el filtro dado, los cuales se podrán filtrar por nombre, apellido, tipo y número de documento y correo. Una vez encontrado al cliente, tras apretar el botón "modificar" se pasará a la misma ventana de creación, la cual ahora sirve para la modificación del cliente seleccionado. En este momento se puede habilitar la casilla de "habilitado" lo que nos permitirá habilitar a un cliente que esté deshabilitado.

Para la ventana de borrado se mostrará la misma lista que para modificación, solo que esta vez solo aparecerán los clientes que estén habilitados en el sistema. Una vez encontrado el cliente de interés apretando el botón "borrar" se deshabilitará al mismo.

## ABM Hotel

La ventana de creación de un hotel nos pedirá un nombre, un teléfono, ciudad, calle y número de domicilio, una cantidad de estrellas (las cuales van de 1 al 7), país (el cual podremos elegir de una lista con todos los países del mundo), fecha de creación, una recarga por estrellas y unos regímenes (estos aparecerán en 2 listas, a la derecha la de los regímenes disponibles y a la izquierda los que tendrá el hotel. Estos podrán moverse de una lista a la otra al seleccionarlos y apretar "<" o ">"). Usando estos datos el programa creará un nuevo hotel.

A la hora de la modificación aparecerá una lista con todos los hoteles del sistema, la cual se podrá filtrar por nombre, ciudad, país y cantidad de estrellas, y una vez elegido el hotel se pasará a la ventana de modificación, la cual es igual a la de creación donde podremos poner los nuevos valores a modificar.

Cabe aclarar que si en algún momento queremos sacarle algún régimen al hotel, el programa verificará que este no se esté usando en ninguna reserva vigente. Si este no es el caso, se dejará que lo modifique, en caso contrario, no.

En cuanto a la ventana de borrado primero se nos mostrará la misma lista que en modificación. Una vez elegido el hotel pasaremos a una nueva ventana. Esta nos

pedirá una fecha de inicio, una de cierre y un motivo. Con esto nos permitirá agregar un nuevo cierre temporal al hotel en cuestión. A la hora de elegir fechas, el sistema no dejara que el usuario escriba alguna fecha menor que la de inicio ni mayor a la de fin. El programa tiene en cuenta que a la hora de hacer un cierre temporal de un hotel este no tenga ninguna habitación con alguna reserva pendiente en esas fechas.

## ABM Habitaciones

La ventana de creación de una habitación nos pedirá un numero, un piso, un tipo (el cual se podrá seleccionar de una lista que nos mostrará los tipos de habitaciones disponibles) y una descripción (la cual no puede ser nula). Además, existe un check box que nos permitirá elegir su ubicación. Con estos datos, al apretar "Agregar habitación", el programa cargará esta al programa.

La ventana de modificación de habitaciones nos mostrará un listado de todas las que existan en el sistema. Estas se podrán filtrar tanto por número, piso y tipo. Una vez elegida la habitación al apretar "modificar" se pasará a una ventana igual a la de creación, solo que esta vez servirá para modificar a la misma.

La ventana de baja temporal nos mostrará la misma lista que en modificación, solo que esta vez al elegir una habitación nos mostrará una nueva ventana. Esta nos pedirá una fecha de inicio y una fecha de cierre (cerciorándose de que la fecha de inicio nunca empiece después que la de cierre y la de cierre nunca antes que la de inicio) y una descripción (la cual no puede ser nula). Con todos estos datos el programa creará una nueva baja temporal para esa habitación. A la hora de hacer esta baja temporal el programa se asegura de que la habitación no tenga ninguna reserva pendiente en estas fechas.

## ABM Reservas

Para crear una reserva se le pedirá al usuario que ingrese una fecha de inicio y una de fin (el programa se encargará de que la fecha de inicio supere a la de fin y que la de fin no sea antes que la de inicio), que seleccione un hotel y un régimen de este. Al tener estos datos y apretar "agregar habitación" se pasará a una nueva ventana donde el programa pedirá el tipo de habitación deseada.

- Si todo está bien el programa nos dará la información de cómo va nuestra reserva en ese momento mostrando la cantidad de días reservados, el subtotal (precio final en caso de ser All Inclusive), etc.
- En caso de querer hacer una reserva que no esté disponible (ya sea porque la habitación deseada ya está reservada en ese periodo o porque esta deshabilitada) el sistema informará el error y le dirá al usuario las habitaciones disponibles para el

periodo que él desea. El sistema también puede tirar error en caso de que la reserva entre en un periodo en que el hotel este deshabilitado.

Finalizado el primer paso se pasará a la pantalla de selección de usuario para así poder asignarle un cliente a la reserva. En caso de estar registrado, el cliente solo se buscará en la lista de clientes, la cual se puede filtrar por tipo y número de documento y correo electrónico. Si por algún motivo el cliente ya se encuentra registrado, pero este se encuentra en gris, significa que este fue uno de los que no se pudo migrar al nuevo sistema. Si este es el caso se le pedirá al cliente que modifique rápidamente sus datos para poder agregarlo satisfactoriamente al sistema y seguir con la reserva.

En caso de que el cliente no esté registrado, con apretar el botón "Regístrate" pasara a una ventana parecida a la de alta de un cliente para que se inserte al sistema.

Y en caso de que el cliente este registrado, pero este se encuentre inhabilitado en el sistema se le informará mediante un error que este no puede realizar reservas y que tendrá que contactar a la administración para su rehabilitación.

Para la modificación de una reserva se le pedirá al usuario el código de la reserva a modificar. Una vez elegida la reserva se mostrará una pantalla parecida a la de reserva, mostrando los datos que tiene la reserva actualmente. En esta pantalla el usuario podrá cambiar la fecha de inicio o de fin de la reserva, el hotel, el régimen e incluso agregar o sacar habitaciones. Una vez modificada la reserva al apretar "modificar reserva" se le mostrara al cliente los datos que tiene ahora su reserva (como la cantidad de días reservados, el subtotal de esta, etc). Cabe destacar que la modificación solo se podrá hacer si es que la reserva está habilitada y esta no tiene una estadía. Recordemos que una reserva se deshabilita si el cliente no ingresa su estadía en el hotel pasado el primer día de la reserva.

Para la cancelación de la reserva el cliente tendrá que ingresar el código de reserva que se le dio. Después de esto se le pedirá al cliente el motivo por el cual se cancelo la reserva y con esto la reserva estará cancelada.

## Ingreso y egreso de Estadías

A la hora de iniciar una estadía en una reserva en específico se le pedirá al usuario el código de su reserva. Una vez ingresada le mostrara una pantalla separada en 2 grandes secciones. La de más arriba es la de estadía, mostrando la fecha de inicio de esta y los clientes que estarán en esta. Al apretar el botón "agregar huésped" se podrá agregar un nuevo huésped a la estadía, dando la opción de seleccionarlo de la lista de los clientes del sistema o registrarlo en el mismo. El programa solo le dejará agregar al cliente hasta el máximo que soporten las habitaciones que el mismo reservó. La sección de abajo mostrara la información de su reserva, mostrando tanto la fecha de inicio y fin de esta como el hotel, el régimen y las habitaciones que intervienen en esta. Finalmente, al apretar el botón "iniciar estadía" la estadía estará iniciada y registrada en el sistema.

Si el usuario ingresa el código de una reserva cuya estadía ya comenzó en vez de dejarle hacer el ingreso de la estadía le dejara hacer el egreso de esta. Para realizar el

egreso basta con solo apretar el botón "finalizar estadía" lo cual cierra la estadía de forma satisfactoria.

Cabe destacar que la fecha de inicio y fin de la estadía no es pedida en ningún momento y esta se guardara automáticamente en cuanto el cliente haga un ingreso o egreso de la estadía (En base al día en cual está siendo ejecutada la aplicación la cual es configurable en el archivo de propiedades).

Si el usuario ingresa el código de una reserva cuya estadía ya está terminada le avisará al cliente de esto y como es lógico, no le dejara al usuario egresar o ingresar una estadía a dicha reserva.

## Registrar consumibles

Para registrar un consumible se le pedirá al usuario que ingrese un código de estadía. El programa se asegurará que sea de una estadía a la cual se le puedan agregar consumibles (lo cual sucede cuando no se ha hecho la facturación, y cuando los consumos no están cerrados). Luego el programa nos mostrará una ventana con los datos de la estadía elegida. Abajo aparecerán los consumibles registrados a dicha estadía y con solo apretar el botón "agregar consumible" podremos pasar a la ventana de agregado. Esta ventana nos pedirá ingresar el producto o servicio, la fecha, la habitación y la cantidad de unidades. Cabe destacar que esta fecha solo podrá ser la actual o anterior, nuestro programa no permite agregar un consumible con una fecha a futuro. Al querer elegir el producto o servicio nos llevará a una nueva ventana la cual nos mostrara todos los consumibles a elegir, los cuales se podrán filtrar por descripción.

Una vez agregado el consumible a la estadía se podrán agregar tantos como se deseen. También a los agregados se los podrá modificar o incluso borrar de la estadía.

## Facturación

Para hacer una factura de una estadía se le pedirá al usuario que ingrese un código de estadía. Una vez validado se le mostrara al usuario una pantalla que indicara desde que día a que día era su reserva y desde que día a que día termino siendo su estadía. Debajo de esos datos se encuentra un botón con el siguiente título: "Agregar Ítem de Factura". Este nos dejara agregar algún detalle a pagar que no sea consumible (daños íntegros a la habitación por ejemplo). En la mitad de la pantalla se mostrará una tabla la cual contiene todos los detalles que se cobraran (ya sean consumibles, cargos adicionales, etc). Por último, abajo de todo el cliente podrá aclarar el medio de pago y dejar algún dato adicional para complementar la factura (número de tarjeta, cantidad de cuotas con las que pagará, etc). Finalmente, al pulsar el botón "Realizar Facturación Final de la Estadía" se creará la factura correspondiente.

## Generar estadísticas



A la hora de generar estadísticas el usuario podrá elegir el año, el trimestre y el tipo de estadística que desee. Una vez elegidos con apretar "generar listado" se listarán los datos pedidos en una tabla la cual se encuentra en la parte inferior de la ventana.