UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL BUENOS AIRES

**GESTIÓN DE DATOS**

**TRABAJO PRÁCTICO**

**FRBA-HOTEL**

**1ER CUATRIMESTRE 2018**

CURSO K2013

GRUPO NRO 50

Neagoe, Martín

143.741-0

Saldivia, Pablo

140.568-8

Santos, Gastón

147.743-2

**Índice**

**Decisiones sobre el Diagrama Entidad-Relación 5**

**Sobre las Tablas 5**

Tabla cliente 5

Tabla clienteInhabilitado 5

Tabla consumible 5

Tabla consumiblePorEstadia 6

Tabla estadia 6

Tabla estadiaCliente 6

Tabla estadoReserva 6

Tabla factura 7

Tabla funcionalidadNombre 7

Tabla habitación 7

Tabla habitacionReserva 8

Tabla hotel 8

Tabla hotelBaja 8

Tabla hotelHabitacion 8

Tabla hotelPorUsuario 9

Tabla hotelRegimen 9

Tabla regimen 9

Tabla reserva 9

Tabla reservaCancelada 10

Tabla rol 10

Tabla rolFuncionalidad 10

Tabla rolPorUsuario 11

Tabla tarjeta 11

Tabla tipoHabitacion 11

Tablas usuarioDatosPersonales y usuarioLogin 11

**Sobre las Relaciones entre las Tablas**  12

Relación funcionalidadNombre–rol 12

Relación rol–usuarioLogin 12

Relación usuarioLogin–hotel 12

Relación usuarioLogin–usuarioDatosPersonales 12

Relación habitacion–tipoHabitacion 13

Relación habitacion–hotel 13

Relación habitacion–reserva 13

Relación hotel–regimen 13

Relación hotel–hotelBaja 13

Relación hotel–cliente 13

Relación cliente–factura 13

Relación factura–tarjeta 13

Relación factura–estadia 14

Relación estadia–consumible 14

Relación estadia–cliente 14

Relación cliente–reserva 14

Relación reserva–reservaCancelada 14

Relación reserva–estadoReservas 14

**Decisiones sobre la Migración 15**

Tabla tipoHabitacion 15

Tabla hotel 15

Tabla hotelBaja 16

Tabla regimen 16

Tabla consumible 16

Tabla habitacion 16

Tabla hotelHabitacion 16

Tabla hotelRegimen 17

Tabla funcionalidadNombre 17

Tabla rol 17

Tabla rolFuncionalidad 17

Tabla usuarioLogin 17

Tabla hotelPorUsuario 17

Tabla rolPorUsuario 17

Tabla usuarioDatosPersonales 17

Tabla reserva 18

Tabla reservaCancelada 18

Tabla estadoReserva 18

Tabla cliente 18

Tabla clienteInhabilitado 19

Tabla estadia 19

Tabla consumiblePorEstadia 19

Tabla estadiaCliente 19

Tabla habitacionReserva 19

Tabla factura 19

Tabla tarjeta 20

**Decisiones sobre la Aplicación de Escritorio 21**

**Decisiones sobre el Diagrama Entidad-Relación**

**Sobre las Tablas**

**Tabla cliente**

Fue creada para almacenar información propia de cada cliente.

La misma cuenta con los siguientes campos:

* **numID. PK**
* **mail. PK**
* apellido.
* nombre.
* tipoID.
* fechaNac.
* telefono.
* domicilio.
* piso.
* depto.
* localidad.
* paisOrigen.
* nacionalidad.

Las PKs son **numID** y **mail**, dado que entre ambas representan unívocamente a un y sólo a un cliente en particular.

**Tabla clienteInhabilitado**

Se creó para registrar a los clientes que están inhabilitados en ciertos hoteles.

Los campos de esta tabla son:

* idHotel. FK
* numID. FK
* mail. FK

Los tres campos son FKs debido a que referencian a PKs en otras tablas (**idHotel** refiere a la tabla **hotel** mientras que **numID** y **mail** refieren a la tabla **cliente**).

**Tabla consumible**

Esta tabla fue creada para almacenar los datos referentes a cada consumible.

Sus campos son:

* **codConsumible. PK**
* descripcion.
* precioUnitario.

Para identificar a cada consumible, se utilizará a **codConsumible**. Es por eso que es PK.

**Tabla consumiblePorEstadia**

Esta tabla fue creada con el propósito de relacionar a cada estadía con sus respectivos consumibles que se fueron pidiendo durante la misma.

Sus campos son:

* idEstadia. FK
* codConsumible. FK

Tanto **idEstadia** y **codConsumible** son FKs porque cada uno de ellos referencia a las tablas **estadia** y **consumible**, respectivamente.

**Tabla estadia**

Esta tabla fue creada con el objetivo de guardar en cada registro información relevante a cada estadía.

Tiene los siguientes campos:

* **idEstadia. PK**
* idReserva. FK
* idHabitacion. FK
* fechaIngreso.
* fechaEgreso.
* usuarioACargoIngreso.
* usuarioACargoEgreso.
* cantNoches.

**idEstadia** es PK porque identifica unívocamente a una estadía.

**idReserva** y **idHabitación** son FKs ya que relacionan a una única reserva y única habitación, referenciadas a las tablas **reserva** y **habitacion**, respectivamente.

**Tabla estadiaCliente**

Esta tabla fue creada teniendo en cuenta que un cliente puede registrar varias estadías. De esa manera, se relacionará a cada estadía con su cliente correspondiente.

Cuenta con estos campos:

* idEstadia. FK
* mail. FK
* numID. FK

Los tres campos son FKs dado que referencian a una PK en otra tabla. **idEstadia** refiere a la tabla **estadia**, en tanto que **mail** y **numID** refieren a la tabla **cliente**.

**Tabla estadoReserva**

Se creó esta tabla para optimizar recursos. Cada registro es un nuevo estado de reserva. Es mejor almacenar repetitivamente un código en la tabla reserva (y que cada código represente cierta leyenda, descripta en esta tabla) que almacenar en forma repetitiva una cadena de caracteres de una extensión considerable.

Cuenta con los siguientes campos:

* **estadoReservas. PK**
* leyendaEstado.

**estadoReservas** es PK porque identifica a una única leyenda del estado de una reserva.

**Tabla factura**

Se creó esta tabla con el objetivo de registrar cada factura, con toda la información relevante.

Sus campos son:

* **numFactura. PK**
* idHotel. FK
* idEstadia. FK
* numID. FK
* mail. FK
* formaDePago.
* nroTarjeta. FK
* Fecha.
* itemDescuentoPorAllInclusive.
* Total.

Cada factura está representada unívocamente por **numFactura**, que es PK.

**idHotel**, **idEstadia**, **numID**-**mail** y **nroTarjeta** son FKs dado que referencian a las tablas **hotel**, **estadia**, **cliente** y **tarjeta**.

**Tabla funcionalidadNombre**

Pensada con el objetivo de optimizar recursos, se hace almacenar (en lugar de una leyenda que haga mención a la funcionalidad) un código de funcionalidad que represente en forma unívoca al nombre de dicha funcionalidad. Es por eso que **codFunc** es PK.

Los campos son los siguientes:

* **codFunc. PK**
* nombreFunc.

**Tabla habitacion**

A cada habitación le corresponderá un registro en esta tabla. A cada una de ellas se las identifica con la PK idHabitación.

Los campos de esta tabla son:

* **idHabitacion. PK**
* tipoHabitacion. FK
* numero.
* piso.
* ubicacion.
* enMantenimiento.
* estado.

**tipoHabitacion**, al referir a la tabla del mismo nombre **tipoHabitacion**, es FK.

**Tabla habitacionReserva**

Esta tabla fue creada dado que cada reserva tiene su propia habitación y que cada habitación tiene su propia reserva.

Por lo tanto, fue necesario agregar estos campos:

* idHabitacion. FK
* idReserva. FK

Ambos campos son FKs, dado que referencian a las tablas **habitación** y **reserva**.

**Tabla hotel**

Cada hotel tiene su fila correspondiente en esta tabla para almacenar información relevante. Y para identificar a cada hotel, se utiliza como PK a **idHotel**.

Sus campos son:

* **idHotel. PK**
* nombre.
* mail.
* telefono.
* direccion.
* cantEstrellas.
* recargaEstrella.
* ciudad.
* pais.
* fechaCreacion.

**Tabla hotelBaja**

Esta tabla fue creada para no cargar con información redundante (como la fecha inicial, la fecha final y la razón de la baja) dentro de la tabla **hotel**. De esta manera, en esta tabla cada fila representa una baja de un hotel.

Sus campos son:

* idHotel. FK
* fechaInicioBaja.
* fechaFinBaja.
* razonBaja.

**idHotel** es FK dado que es necesario referenciar al hotel que fue dado de baja.

**Tabla hotelHabitacion**

Cada habitación concierne a un hotel en particular. Por ende, corresponde una nueva fila en esta tabla que relacione a cada habitación con su hotel correspondiente.

Los campos de esta tabla son los siguientes:

* idHotel. FK
* idHabitacion. FK

Tanto **idHotel** como **idHabitacion** son FKs porque es preciso que tengan su referencia a las tablas **hotel** y **habitación**, respectivamente.

**Tabla hotelPorUsuario**

Esta tabla fue creada con el objetivo de satisfacer el requerimiento de registrar el hotel en el que se desempeña un usuario a la hora de registrar a este último. Además, esta tabla facilita la obtención de cierta información (como es la obtención del nombre del usuario a partir de una reserva).

La tabla **hotelPorUsuario** cuenta con los siguientes campos:

* idUsuario. FK
* idHotel. FK

Ambas son FK dado que referencian a un usuario y a un hotel en particular, respectivamente.

**Tabla hotelRegimen**

Debido a que cada hotel cuenta con uno o varios regímenes de estadía, en esta tabla se irá almacenando el par hotel-régimen. Los campos son, entonces:

* idHotel FK
* codRegimen FK

Ambos son FKs dado que cada una referencia a un hotel y a un régimen de estadía en particular.

**Tabla regimen**

Fue necesario crear esta tabla para registrar cada régimen de estadía con sus correspondientes características.

Presenta los siguientes campos:

* **codRegimen. PK**
* precioBase.
* descripcion.
* estado.

**codRegimen** es PK porque representa a un y solamente a un único régimen de estadía.

**Tabla reserva**

Fue creada con el fin de registrar cada reserva que se realice, relacionando a cada una de ellas con su hotel, con el cliente que la generó, con el régimen elegido, con el tipo de habitación elegido y demás particularidades.

La tabla cuenta con estos campos:

* **idReserva. PK**
* idHotel. FK
* codRegimen. FK
* tipoHabitacion. FK
* idUsuario. FK
* numID. FK
* mail. FK
* fechaInicio.
* fechaDesde.
* fechaHasta.
* estadoReserva. FK
* costoTotalReserva.

**idReserva** representa en forma unívoca a una reserva. Es por eso que es PK.

Tanto **idHotel** como **codRegimen**, **tipoHabitacion**, **idUsuario**, **numID**-**mail** y **estadoReserva** refieren a: un hotel, un régimen de estadía, un tipo de habitación, un usuario, un cliente y un estado de reserva. Es por eso que todas ellas son FKs.

**Tabla reservaCancelada**

Al cancelarse una reserva, se crea una nueva fila en esta tabla guardando información representada en los siguientes campos:

* idReserva. FK
* idUsuario. FK
* fechaCancelacion.
* motivoCancelacion.

**idReserva** y **idUsuario** son FKs dado que cierta reserva es cancelada por un determinado usuario, por eso es necesario relacionar a cada reserva cancelada con las tablas **reserva** y **usuario**.

**Tabla rol**

Esta tabla fue creada con la finalidad de registrar los estados de cada rol.

Es por eso que cuenta con estos campos:

* **rol. PK**
* estado.

El campo **rol** es PK porque identifica a un único rol en particular.

**Tabla rolFuncionalidad**

Se creó esta tabla para relacionar a cada rol con cada funcionalidad, dado que cada rol tiene ciertas funcionalidades diferentes a las de otros roles.

La tabla contiene con estos campos:

* rol. FK
* codFunc. FK

Ambos campos son FKs dado que **rol** refiere a la tabla **rol** y **codFunc** refiere a la tabla **funcionalidadNombre**.

**Tabla rolPorUsuario**

Esta tabla se creó para dejar registro de qué roles puede tomar cada usuario y en qué hoteles puede ejercer dichos roles.

Sus campos son:

* rol. FK
* idUsuario. FK
* idHotel. FK

Cada uno de los campos son FKs dado que hacen referencia a las tablas **rol**, **usuario** y **hotel**, respectivamente.

**Tabla tarjeta**

Cada fila corresponde a una tarjeta diferente, con los siguientes datos:

* **nroTarjeta PK**
* banco.
* marca.
* fechaVencimiento.
* cvv.

**nroTarjeta** es PK dado que representa a una y solamente a una tarjeta.

**Tabla tipoHabitacion**

Esta tabla almacena registros por cada tipo de habitación.

Tiene los siguientes campos:

* **tipoHabitacion. PK**
* descripcion.
* porcentual.

**tipoHabitacion** es una PK dado que representa unívocamente a un solo tipo de habitación.

**Tablas usuarioDatosPersonales y usuarioLogin**

Si bien toda la información referente a un usuario depende en forma exclusiva de cada usuario, consideramos importante segmentar dicha información en dos partes para simplificar futuros requerimientos: por un lado, los datos personales del usuario y, por el otro, los datos necesarios para realizar el *log-in*. Es por eso que se crean dos tablas diferentes: **usuarioDatosPersonales** y **usuarioLogin**.

La tabla **usuarioDatosPersonales** cuenta con los siguientes campos:

* idUsuario. FK
* nombre
* apellido
* tipoDoc
* numDoc
* mail
* telefono
* direccion
* fechaNac

El campo **idUsuario** es FK dado que refiere a la tabla **usuarioLogin**.

La tabla **usuarioLogin** cuenta con los siguientes campos:

* **idUsuario. PK**
* username.
* password.
* cantIngresosFallidos.
* estado.

**idUsuario** es PK dado que dicho campo identifica unívocamente a un usuario.

Tomamos la decisión de que la PK esté en la tabla **usuarioLogin** dado que, a comparación de la información almacenada en **usuarioDatosPersonales**, la primera es más relevante para nuestros propósitos: contiene información como el nombre de usuario y la contraseña, primordiales para realizar correctamente el registro e ingreso al sistema, independientemente de lo datos personales de cada usuario.

**Sobre las Relaciones entre las Tablas**

**Relación funcionalidadNombre–rol**

Como cada rol puede tener más de una funcionalidad y, además, cada funcionalidad puede estar presente en más de un rol, fue necesario desprender esa relación “muchos a muchos” creando una nueva tabla intermedia: la tabla **rolFuncionalidad**.

**Relación rol–usuarioLogin**

Dado que cada usuario puede tener más de un rol y que cada rol puede ser ejercido por más de un usuario, es necesario crear una nueva tabla para desglosar esa relación “muchos a muchos”. A eso se debe la existencia de la tabla **rolPorUsuario**.

**Relación usuarioLogin–hotel**

Dado que cada usuario puede estar presente en más de un hotel y que cada hotel puede ser manejado o trabajado por más de un usuario, fue necesario desprender esa relación “muchos a muchos”. Esto se logró creando la tabla **hotelesPorUsuario**.

**Relación usuarioLogin–usuarioDatosPersonales**

Los datos necesarios para el *log-in* (como el nombre de usuario y la contraseña) se corresponden con un **idUsuario**, campo por el cual se relacionan también los datos personales de dicho usuario (como lo son el nombre, el apellido, el número de documento, el correo electrónico y el teléfono, por ejemplo). Es por eso que dicha relación es “uno a uno”.

**Relación habitacion–tipoHabitacion**

Cada habitación tiene un único tipo de habitación y cada tipo de habitación puede contener a una o a muchas habitaciones. Eso explica la relación “muchos a uno”.

**Relación habitacion–hotel**

Para simplificar futuros requerimientos en la aplicación de escritorio, se separó esta relación creando una nueva tabla **hotelHabitacion**, la cual almacena a cada habitación con el hotel al cual pertenece.

**Relación habitacion–reserva**

Dado que una habitación puede tener varias reservas y que cada reserva puede tener más de una habitación, fue necesario destrabar el inconveniente de esa relación “muchos a muchos”. Por lo tanto, se creó una tabla **habitacionReserva**, que almacena todos los registros de cada habitación con su respectiva reserva.

**Relación hotel–regimen**

Cada hotel tiene sus regímenes y cada régimen puede usarse en varios hoteles. Tenemos entonces una relación “muchos a muchos” que, nuevamente, se desarma creando una nueva tabla. Esta tabla se llama **hotelRegimen**, y tiene una fila por cada hotel con su correspondiente régimen con el que dicho hotel trabaja.

**Relación hotel–hotelBaja**

Al darse de baja un hotel por cierto tiempo, se carga un nuevo registro en la tabla **hotelBaja**.

En la tabla **hotel** tenemos registrados a todos los hoteles (con sus respectivas características), mientras que en la tabla **hotelBaja** se van cargando todos aquellos que se van dando de baja. Cada registro se vincula con otro mediante el campo **idHotel**, presente en ambas tablas.

**Relación hotel–cliente**

Dado que existe la posibilidad de que un cliente sea inhabilitado en cierto hotel, tenemos que tener en cuenta almacenar dicha información. Necesitamos relacionar a la tabla **hotel** con la tabla **cliente** teniendo en cuenta que un cliente puede estar inhabilitado en varios hoteles y que un hotel puede inhabilitar a varios clientes. Dicho eso, debemos separar esa relación “muchos a muchos” creando una nueva tabla: **clienteInhabilitado**.

**Relación cliente–factura**

Dado que en una factura puede aparecer un único cliente, tal relación es uno a muchos.

**Relación factura–tarjeta**

Una factura puede o no pagarse con tarjeta. Pero una tarjeta puede haber pagado varias facturas. Por lo tanto, la relación es “muchos a cero/uno”

**Relación factura–estadia**

A cada estadía le corresponde una única factura, y viceversa. Por ende, la relación es “uno a uno”.

**Relación estadia–consumible**

Como una estadía puede tener registrados varios consumibles y un consumible puede estar en varias estadías, fue necesario crear una tabla intermedia (la tabla **consumiblePorEstadia**) para eliminar esa relación “muchos a muchos”. La tabla creada registrará a cada estadía con sus respectivos consumibles que se fueron pidiendo durante la misma.

**Relación estadia–cliente**

Dado que una estadía puede tener a más de un cliente y un cliente puede estar registrado en varias estadías distintas, se crea una relación “muchos a muchos”. Pero para no generar inconvenientes, es necesario eliminar dicha relación y colocar una tabla en su lugar, relacionando a ambas tablas. Dicha tabla, **estadiaCliente**, relaciona a cada estadía con su/s correspondiente/s cliente/s.

**Relación cliente–reserva**

Un cliente puede tener varias reservas, pero una reserva puede tener solamente a un cliente. Eso explica la relación “uno a muchos”.

**Relación reserva–reservaCancelada**

Puede o no haber reservas canceladas por cada reserva, por lo tanto, la relación es de tipo “uno a cero/uno”.

**Relación reserva–estadoReservas**

A cada reserva le corresponde un y solamente un estado. Y a cada estado de reservas puede corresponderle uno o bien ninguna reserva (puede existir la posibilidad de que no haya reservas canceladas, por ejemplo, entonces ninguna reserva tendría ese valor correspondiente en la tabla **reserva**). Eso explica la relación “cero/uno a uno”.

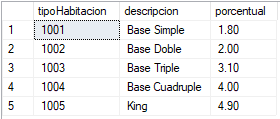
**Decisiones sobre la Migración**

Tiene su justificación el orden de ejecución: primero deben crearse las tablas que tienen PKs para que, luego, al crear las que tienen FKs, éstas tengan a quién referenciar sus campos.

**Tabla tipoHabitacion**

* **tipoHabitacion** (PK) *Idéntico al campo* ***Habitacion\_Tipo\_Codigo*** *de la tabla maestra.*
* descripcion *Idéntico al campo* ***Habitacion\_Tipo\_Descripcion*** *de la tabla maestra.*
* porcentual *Idéntico al campo* ***Habitacion\_Tipo\_Porcentual*** *de la tabla maestra.*

A continuación, el contenido de la tabla luego de la migración:

****

**Tabla hotel**

* **idHotel** (PK) *Creado para identificar a cada hotel y para diferenciarlos de los demás.*
* nombre *Dado que se desconoce el nombre de cada hotel, se almacena* ***NULL****.*
* mail *Dado que se desconoce el mail de cada hotel, se almacena* ***NULL****.*
* telefono *Dado que se desconoce el teléfono de cada hotel, se almacena* ***NULL****.*
* direccion *Almacena la concatenación de los campos* ***Hotel\_Calle*** *y* ***Hotel\_Nro\_Calle*** *de la tabla maestra.*
* cantEstrellas *Idéntico al campo* ***Hotel\_CantEstrella*** *de la tabla maestra.*
* recargaEstrella *Idéntico al campo* ***Hotel\_Recarga\_Estrella*** *de la tabla maestra.*
* ciudad *Idéntico al campo* ***Hotel\_Ciudad*** *de la tabla maestra.*
* pais *Sabiendo que todas las ciudades son argentinas, para cada hotel en este campo se almacenó la cadena “****Argentina****”.*
* fechaCreacion *Dado que se desconoce la fecha de creación de cada hotel, se almacena* ***NULL****.*

A continuación, el contenido de la tabla luego de la migración:



**Tabla hotelBaja**

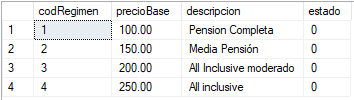
* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* fechaInicioBaja *Preparado para almacenar la fecha en la cual se dio de baja al hotel en cuestión.*
* fechaFinBaja *Preparado para almacenar la fecha en la cual terminó la baja de dicho hotel.*
* razonBaja *Preparado para almacenar el motivo por el cual el hotel fue dado de baja.*

Dado que al momento de la migración inicial no hay ningún hotel dado de baja, esta tabla está vacía.

**Tabla regimen**

* **codRegimen** (PK) *Creado para identificar a cada régimen de estadía y para diferenciarlos de los demás.*
* precioBase *Análogo al campo* ***Regimen\_Precio*** *de la tabla maestra.*
* descripcion *Análogo al campo* ***Regimen\_Descripcion*** *de la tabla maestra.*
* estado *Representa un flag que indica si dicho régimen de estadía está activo o inactivo.*

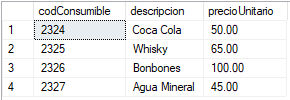
A continuación, el contenido de la tabla luego de la migración:



**Tabla consumible**

* **codConsumible** (PK) *Análogo al campo* ***Consumible\_Codigo*** *de la tabla maestra.*
* descripcion *Análogo al campo* ***Consumible\_Descripcion*** *de la tabla maestra.*
* precioUnitario *Análogo al campo* ***Consumible\_Precio*** *de la tabla maestra.*

A continuación, el contenido de la tabla luego de la migración:



**Tabla habitacion**

* **idHabitacion** (PK) *Creado para identificar a cada habitación y para diferenciarlas de los demás.*
* tipoHabitacion (FK) *Refiere al tipo de habitación al que pertenece.*
* numero *Análogo al campo* ***Habitacion\_Numero*** *de la tabla maestra.*
* piso *Análogo al campo* ***Habitacion\_Piso*** *de la tabla maestra.*
* ubicacion *Análogo al campo* ***Habitacion\_Frente*** *de la tabla maestra.*
* enMantenimiento *Representa un flag que indica si dicha habitación se encuentra o no en mantenimiento.*
* estado *Representa un flag que indica si dicha habitación está habilitada o bien deshabilitada.*

**Tabla hotelHabitacion**

* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* idHabitacion (FK) *Refiere al campo del mismo nombre de la tabla* ***habitacion****.*

**Tabla hotelRegimen**

* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* codRegimen (FK) *Refiere al campo del mismo nombre de la tabla* ***regimen****.*

**Tabla funcionalidadNombre**

* **codFunc** (PK) *Creado para identificar a cada funcionalidad y para diferenciarlas de los demás.*
* nombreFunc *Creado para almacenar la leyenda correspondiente a cada funcionalidad.*

**Tabla rol**

* **rol** (PK) *Creado para identificar a cada rol y para diferenciarlas de los demás.*
* estado *Representa un flag que indica si dicho régimen de estadía está activo o inactivo.*

**Tabla rolFuncionalidad**

* rol (FK) *Refiere al campo del mismo nombre de la tabla* ***rol****.*
* codFunc (FK) *Refiere al campo del mismo nombre de la tabla* ***funcionalidadNombre****.*

**Tabla usuarioLogin**

* **idUsuario** (PK) *Creado para identificar a cada usuario y para diferenciarlos de los demás.*
* username *Creado para almacenar el username.*
* password *Creado para almacenar la contraseña hasheada bajo el algoritmo SHA256.*
* cantIngresosFallidos *Creado para almacenar la cantidad de ingresos fallidos al querer ingresar al sistema.*
* estado *Representa un flag que indica si dicho usuario se encuentra habilitado o deshabilitado.*

**Tabla hotelPorUsuario**

* idUsuario (PK) *Refiere al campo del mismo nombre de la tabla* ***usuario****.*
* idHotel (PK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*

**Tabla rolPorUsuario**

* rol (FK) *Refiere al campo del mismo nombre de la tabla* ***rol****.*
* idUsuario (FK) *Refiere al campo del mismo nombre de la tabla* ***usuario****.*
* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*

**Tabla usuarioDatosPersonales**

* idUsuario (FK) *Refiere al campo del mismo nombre de la tabla* ***usuario****.*
* nombre *Creado para almacenar el nombre del usuario.*
* apellido *Creado para almacenar el apellido del usuario.*
* tipoDoc *Creado para almacenar el tipo de documento del usuario.*
* numDoc *Creado para almacenar el número de documento del usuario.*
* mail *Creado para almacenar el mail del usuario.*
* telefono *Creado para almacenar el teléfono del usuario.*
* direccion *Creado para almacenar la dirección del usuario.*
* fechaNac *Creado para almacenar la fecha de nacimiento del usuario.*

**Tabla reserva**

* **idReserva** (PK) *Creado para identificar a una reserva en particular y para diferenciarlas de las demás.*
* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* codRegimen (FK) *Refiere al campo del mismo nombre de la tabla* ***regimen****.*
* tipoHabitacion (FK) *Refiere al campo del mismo nombre de la tabla* ***tipoHabitacion****.*
* idUsuario (FK) *Refiere al campo del mismo nombre de la tabla* ***usuario****.*
* numID (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* mail (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* fechaInicio *Creado para almacenar la fecha en la cual se registró la reserva en cuestión.*
* fechaDesde *Creado para almacenar la fecha en la cual comienza dicha reserva.*
* fechaHasta *Creado para almacenar la fecha en la cual dicha reserva concluye.*
* estadoReserva (FK) *Refiere al campo del mismo nombre de la tabla* ***usuario****.*
* costoTotalReserva *Creado para almacenar al costo total diario de la reserva.*

*Usando los campos de la tabla maestra, su valor se obtiene así:*  ***Regimen\_Precio*** *\** ***Habitacion\_Tipo\_Porcentual*** *+* ***Hotel\_CantEstrella*** *\** ***Hotel\_Recarga\_Estrella****.*

**Tabla reservaCancelada**

* idReserva (FK) *Refiere al campo del mismo nombre de la tabla* ***reserva****.*
* idUsuario (FK) *Refiere al campo del mismo nombre de la tabla* ***usuarioLogin****.*
* fechaCancelacion *Creado para almacenar la fecha en la que se canceló la reserva en cuestión.*
* motivoCancelacion *Creado para almacenar el motivo por el que fue cancelada dicha reserva.*

Dado que actualmente no hay ninguna reserva cancelada, esta tabla se encuentra vacía.

**Tabla estadoReserva**

* **estadoReserva** (PK) *Creado para identificar unívocamente a un estado de reserva en particular.*
* leyendaEstado *Creado para almacenar la cadena de caracteres del estado de reserva.*

**Tabla cliente**

* **numID (PK).** *Análogo al campo* ***Cliente\_Pasaporte\_Nro*** *de la tabla maestra.*
* **mail (PK).** *Análogo al campo* ***Cliente\_Mail*** *de la tabla maestra.*
* apellido. *Análogo al campo* ***Cliente\_Apellido*** *de la tabla maestra.*
* nombre. *Análogo al campo* ***Cliente\_Nombre*** *de la tabla maestra.*
* tipoID. *Almacena el tipo de identificación de cada cliente. Pasaporte, por ejemplo.*
* fechaNac. *Análogo al campo* ***Cliente\_FechaNac*** *de la tabla maestra.*
* telefono. *Dado que se desconoce el teléfono de cada cliente, se almacena* ***NULL****.*
* domicilio. *Almacena la concatenación de los campos* ***Cliente\_Dom\_Calle*** *y* ***Cliente\_Nro\_Calle*** *de la tabla maestra.*
* piso. *Análogo al campo* ***Cliente\_Piso*** *de la tabla maestra.*
* depto. *Análogo al campo* ***Cliente\_Depto*** *de la tabla maestra.*
* localidad. *Dado que se desconoce la localidad de cada cliente, se almacena* ***NULL****.*
* paisOrigen. *Almacena el país correspondiente al adjetivo gentilicio provisto por la nacionalidad.*
* nacionalidad. *Análogo al campo* ***Cliente\_Nacionalidad*** *de la tabla maestra.*

**Tabla clienteInhabilitado**

* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* numID (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* mail (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*

**Tabla estadia**

* **idEstadia** (PK) *Creado para identificar a cada estadía y para diferenciarlas de los demás.*
* idReserva (FK) *Refiere al campo del mismo nombre de la tabla* ***reserva****.*
* idHabitacion (FK) *Refiere al campo del mismo nombre de la tabla* ***habitacion****.*
* fechaIngreso *Análogo al campo* ***Estadia\_Fecha\_Inicio*** *de la tabla maestra.*
* fechaEgreso *Dado que se desconoce la fecha de ingreso, se almacena* ***NULL****.*
* usuarioACargoIngreso *Dado que se desconoce el usuario que registró el ingreso, se almacena* ***NULL****.*
* usuarioACargoEgreso *Dado que se desconoce el usuario que registró el egreso, se almacena* ***NULL****.*
* cantNoches *Análogo al campo* ***Estadia\_Cant\_Noches*** *de la tabla maestra.*

**Tabla consumiblePorEstadia**

* idEstadia (FK) *Refiere al campo del mismo nombre de la tabla* ***estadia****.*
* codConsumible (FK) *Refiere al campo del mismo nombre de la tabla* ***consumible****.*

**Tabla estadiaCliente**

* idEstadia (FK) *Refiere al campo del mismo nombre de la tabla* ***estadia****.*
* numID (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* mail (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*

**Tabla habitacionReserva**

* idHabitacion (FK) *Refiere al campo del mismo nombre de la tabla* ***habitacion****.*
* idReserva (FK) *Refiere al campo del mismo nombre de la tabla* ***reserva****.*

**Tabla factura**

* **numFactura** (PK) *Análogo al campo* ***Factura\_Nro*** *de la tabla maestra.*
* idHotel (FK) *Refiere al campo del mismo nombre de la tabla* ***hotel****.*
* idEstadia (FK) *Refiere al campo del mismo nombre de la tabla* ***estadia****.*
* numID (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* mail (FK) *Refiere al campo del mismo nombre de la tabla* ***cliente****.*
* formaDePago *Dado que se desconoce la fecha de ingreso, se almacena* ***NULL****.*
* nroTarjeta (FK) *Refiere al campo del mismo nombre de la tabla* ***tarjeta****. Dado que se desconoce la fecha de ingreso, se almacena* ***NULL****.*
* fecha *Dado que se desconoce la fecha de ingreso, se almacena* ***NULL****.*
* itemDescuentoPorAllInclusive *Dado que se desconoce la fecha de ingreso, se almacena* ***0****.*
* total *Análogo al campo* ***Factura\_Total*** *de la tabla maestra.*

**Tabla tarjeta**

* **nroTarjeta** (PK) *Creado para identificar a cada tarjeta y para diferenciarlas de los demás.*
* banco *Preparado para almacenar el nombre del banco que brinda la tarjeta.*
* marca *Preparado para almacenar la marca de la tarjeta.*
* fechaVencimiento *Preparado para almacenar la fecha en la cual vence dicha tarjeta.*
* cvv *Preparado para almacenar el código de verificación de la tarjeta en cuestión.*

Dado que actualmente no hay ninguna tarjeta registrada, esta tabla se encuentra vacía.

**Decisiones sobre la Aplicación de Escritorio**

Para la pantalla inicial, nos decidimos por tres botones con nombres expresivos, los cuales llevarán a tres formularios distintos: el del **login**, el de **generar reserva** y el de **modificar reserva**.

Como desconocemos las resoluciones de los monitores en los que será ejecutada la aplicación, tomamos las decisiones de hacer que sus tamaños sean los menores posibles y centrarlos en la pantalla.

En lo que respecta al **formulario de** **login**, el usuario no tendrá posibilidad de acceder a menos que complete los campos de usuario (primero) y contraseña (luego). De no completar alguno de ellos, no podrá ingresar al sistema. Una vez ingresado al sistema, el usuario procederá a la selección de rol, hotel y funcionalidad.

Dentro del **formulario de** **selección de rol, hotel y funcionalidad**, nos decidimos por 3 comboBox los cuales irán mostrando los roles, hoteles y funcionalidades disponibles. Una vez ingresado al sistema, el usuario podrá elegir el **rol** que desea ejercer en esta sesión. Luego, una vez elegido el rol, se habilita el comboBox para que el usuario pueda el **hotel** con el que el usuario desee trabajar habiendo accedido con cierto rol. Después, el usuario podrá elegir la **funcionalidad** correspondiente (si dicha funcionalidad es un ABM, se mostrará un cuarto comboBox para elegir en forma particular qué se desea hacer: si dar de alta, si modificar o bien si eliminar). Finalmente, y de acuerdo a lo seleccionado previamente, se habilitará un botón para permitirle al usuario acceder a la funcionalidad requerida.

Para los formularios de Baja y Modificación (que forman parte de un ABM), primero se mostrará un formulario de búsqueda con varios filtros para poder seleccionar lo requerido. Hemos resuelto incluir en los filtros aquellos campos más comunes para buscar (por ejemplo, para buscar un hotel no pusimos filtros para la fecha de creación sino filtros para los nombres, la cantidad de estrellas o la ciudad en la cual se encuentran). Una vez seleccionado, se procede a presionar el botón correspondiente para darlo de baja o modificarlo. Se tomó la decisión de trabajar con un único formulario de búsqueda para cada entidad (uno para los hoteles, uno para los usuarios, uno para los clientes, etcétera) con el objeto de simplificar tareas.