

Estrategia

GRUPO 51 - HOTEL_TRIVAGOS

Nombre y Apellido	Legajo
Daniel Ávila	1419791
Matías Duarte	1446009
Matías Magarzo	1498575
Emilio José Nogueiras	1180046

INDICE	Página
1. TABLAS.....	3
2. MIGRACION.....	4
3. FUNCIONALIDADES.....	4
4. LISTADO ESTADISTICO.....	4
5. APLICACIÓN C#.....	5
5.1. ABM.....	5
5.2. GENERAR RESERVA.....	5
5.3. MODIFICAR RESERVA.....	5
5.4. CANCELAR RESERVA.....	6
5.5. FACTURAR ESTADÍAS.....	6
5.6. REGISTRAR ESTADÍA.....	6
5.7. LISTADO ESTADÍSTICO.....	6
5.8. REGISTRAR CONSUMIBLES.....	6
5.9. ABM HOTEL.....	7

Tablas:

Modelamos los usuarios y clientes en tablas separadas debido a que los primeros se loguean y pueden tener diferentes roles.

Para contemplar las bajas lógicas y deshabilitaciones por login agregamos los bits de Usuario_Baja y Usuario_Deshabilitado, éste último acompañado del campo Usuario_Intentos.

Para los datos personales, ya que los campos se repetían, creamos una sola tabla (DATOS_PERSONALES) y la utilizamos para los dos casos (USUARIOS y CLIENTES).

Los usuarios tienen diferentes roles dependiendo del hotel donde trabajan, por este requerimiento agregamos una tabla intermedia USUARIOS_X_ROL_X_HOTEL.

A su vez cada rol puede tener varias funcionalidades, de ahí la tabla FUNCIONALIDADES_X_ROL.

La tabla HOTELES, aparte de contener los datos propios de cada uno, tiene varios regímenes (modelados en la tabla REGIMENES_ESTADIA y REGIMEN_X_HOTEL) y periodos de cierre (tabla HOTEL_MANTENIMIENTO).

Indicamos que un hotel se encuentra cerrado con el bit Hotel_Cerrado.

Cada habitación tiene una FK al hotel que pertenece y a su tipo de habitación (tabla TIPO_HABITACION cuyos campos son la descripción, el porcentaje y la cantidad máxima de personas)

Para las reservas consideramos las siguientes cuestiones:

- Una reserva tiene una o muchas habitaciones (tabla RESERVAS_X_HABITACION) por lo que el régimen va a ser uno solo por reserva, abarcando a todas las habitaciones de la misma.
- Se almacena en la reserva el cliente que reservó.
- El agregado de la tabla LOG_RESERVA, donde se inserta una fila ante una generación o modificación de Reserva, considerando siempre que si el que modifica la reserva no es un usuario, es entonces el cliente que reservó.

Consideramos que cada estadía es de una habitación sola debido a que al momento de facturar cada ítem de la factura deberá ser una estadía o un consumible. Por este motivo cada estadía tiene una FK a RESERVAS_X_HABITACION donde los datos necesarios de la estadía (es decir la habitación y los datos de fechas de ingreso, régimen, etc) los obtenemos de las filas de las tablas RESERVAS_X_HABITACION y RESERVAS correspondientes.

Para registrar los consumibles de cada habitación creamos la tabla CONSUMIBLES_X_HABITACION que relaciona un consumible con una RESERVAS_X_HABITACION. Con esto sabemos que es lo que se consumió en cada habitación y en cada estadía.

Para modelar lo antes nombrado sobre los ítems de la factura, que sean una Estadía o un CONSUMIBLES_X_HABITACION, definimos que las PKs de éstas sean a su vez FK a la tabla FACTURA_ITEM. Así nos aseguramos que ambas "subclases" de FACTURA_ITEM, aparte de ser polimórficas, sean excluyentes.

Luego en la tabla FACTURAS almacenamos entre otros campos (fecha, total, etc) al cliente que reservó la estadía y a la forma de pago. Para la última, en caso de pagar con tarjeta, almacenamos la información de la misma en la tabla DATOS_TARJETA (teniendo FACTURAS una FK a esta tabla). En caso de que la forma de pago sea en efectivo la FK será null.

Cuando se realiza un check-in, es decir un insert en la tabla ESTADIAS, o se registra un consumible por habitación, se genera un ítem factura. Al momento de cobrarle al cliente se genera una factura y se le asignan los ítems generados anteriormente.

A la hora de crear los Item_factura por Estadía, la lógica fue la siguiente:

GESTION DE DATOS – CURSO K3052

- Al hacer el check_in se crea un Item_factura por el total de las noches.
- Al hacer el check_out si el huésped se retira con anterioridad, se crea un nuevo Item_factura correspondiente a las noches no efectivizadas y se actualiza el creado en el check_in para que solo comprenda el monto por las noches que el cliente se hospedó realmente.

Por último para llevar un control de que cliente está hospedado en que habitación, creamos la tabla HUESPEDES_X_HABITACION que relaciona un cliente con una RESERVAS_X_HABITACION.

Migración

Se utilizaron inserts masivos para la migración de todas las tablas excepto en ESTADIAS y CONSUMIBLES_X_HABITACION. Para esos casos utilizamos un cursor ya que, por cada fila también tenemos que realizar un insert a la tabla FACTURA_ITEM.

Luego por la forma en que modelamos CLIENTES y DATOS_PERSONALES (donde del primero para la migración sólo interesa la nacionalidad y los datos (FK)) los id del cliente coinciden con su FK a datos personales. Por ende para que sea mas performante se decidió migrar primero la tabla DATOS_PERSONALES, luego CLIENTES y por último relacionarlas realizando un update masivo de Clientes asignándole su respectiva FK. Como antes aclaramos, la igualdad se cumple sólo para la migración y se utilizó a fines de mejorar la performance en el tiempo de ejecución.

Se efectivizaron las reservas que tenían una estadía relacionada con un update y considerando que las que no tienen estado "Efectivizada" son "Correctas" ya que no podemos inferir otra información.

Se incluyen en el script inicial 3 usuarios con su respectivo perfil para pruebas iniciales.

Funcionalidades

Las fechas se reciben por parámetro como un nvarchar(15) con el formato YYYYMMDD y para almacenarlas, creamos nuevos datetime y les asignamos directamente ese nvarchar.

Listado Estadístico

Para este punto utilizamos funciones que reciben por parámetro el año, el mes de inicio y fin del periodo y retornan una tabla con la consulta correspondiente a cada punto y a los parámetros.

Estrategia:

A continuación se detalla la estrategia de la aplicación en C# del trabajo práctico.

Para empezar se cuenta con la primer form llamada principal que internamente hace la conexión con la base de datos, la cual en caso de tirar un error, se muestra por pantalla para su solución. De esta form se parte al proceso de login o si es un invitado el botón invitado lo lleva directamente al menú Guest para poder cancelar, generar o modificar una reserva.

En el proceso de login primero se pide el usuario y la contraseña para validarlos en la base. Si el login es correcto se procede a una form para elegir el tipo de rol y después otra para elegir el hotel en el que desea iniciar la sesión. Para esto contamos con una clase UsuarioLogeado (Carpeta Sistema) la cual va a tener el nombre, password y hotel en el que ha iniciado sesión, entre otras cosas.

Todos los accesos que se hacen a la base de datos (como por ejemplo una consulta o ejecutar un procedure); esos "métodos" están contemplados en la clase SQLConnector también en la carpeta Sistema.

Una vez que el login este validado y se hayan elegido todos los campos correspondientes, dependiendo del tipo de rol elegido se carga el menú para el cual primero hay una Clase llamada carga menú que se encarga de ejecutar la form correspondiente.

Ya en el menú se pueden visualizar los botones, acorde a las funcionalidades permitidas.

Aplicación C#

A continuación se procede a explicar sobre la codificación de las funcionalidades:

- **ABM's:**

Para el alta, hay una combinación de textbox y algunos combobox (entre otras cosas) para hacer la carga de datos. Una vez completados todos se procede a ejecutar la misma; si algún campo está sin completar, aparece un cartel con la leyenda "...complete todos los campos...", para que cuando se hace la llamada al SQL de que ejecute el respectivo procedimiento de alta, no tire error por falta de algún dato.

Tanto en la modificación como la baja, primero aparecerá un listado que va a tener un criterio, acorde a cual botón hayamos elegido (sea modificar o baja). En el listado se podrá filtrar por alguno o todos los campos que se deseen, y una vez que es seleccionado el valor deseado se carga la respectiva form sobre la acción que anteriormente se pidió.

ACLARACIÓN:

A la hora de querer hacer una búsqueda nueva, es NECESARIO el botón limpieza que va a limpiar todos los campos incluyendo el datagrid donde se muestra la grilla con la respuesta de la consulta al SQL.

Para la modificación en sí, se cargan todos los datos y se le permite al usuario que modifique cualquiera. Una vez hecho eso se hace una nueva consulta al SQL para que realice la modificación. En caso de que la consulta sea satisfactoria aparece una leyenda de "...modificación exitosa...".

Para la baja no permite ninguna modificación de ningún campo y lo que hace es ejecutar un update a la tabla correspondiente para que modifique el bit de estado sobre el valor seleccionado.

- **Generar Reserva:**

Hay 2 posibles formas de entrar a esta sección, siendo Recepcionista o invitado.

En el caso del recepcionista no se le permite elegir el hotel dado que el mismo solo puede hacer reservas en su hotel. Una vez elegida las fechas y el tipo de régimen, se mostrarán todas las habitaciones que estén disponibles para hacer la reserva.

Una vez elegida una habitación, el botón confirmar pasara a mostrar el precio. Si se confirma se pregunta si el cliente está registrado en la cadena hotelera; en caso afirmativo se procede a buscar el cliente; en caso negativo se levanta la form alta cliente para que sea registrado.

Independientemente de cual haya sido el caso se muestra el id del cliente seleccionado para que después sea cargado en la form en cuestión; esto es necesario dado que generar una reserva necesita un cliente Id. En el caso de haber entrado como invitado lo que difiere es que el recepcionista va a ser NULL. Una vez cargado el id del cliente se procede a finalizar la reserva la cual llama a los procedures y en caso de éxito aparece una leyenda de "reserva generada con éxito", se muestra el código de la reserva en cuestión y se cierra la form.

- **Modificar Reserva:**

Se pide que se ingrese el número de la reserva. El mismo es buscado en la base de datos teniendo en cuenta que no esté finalizada. Se muestra en una grilla las habitaciones que fueron reservadas, cuales habitaciones están libres y el régimen elegido.

Se permite cambiar el tipo de régimen, quitar o agregar habitaciones libres y cambiar las fechas de duración de la reserva. Una vez hecho esto el botón modificar procede a mostrar el nuevo precio de la reserva. Si se acepta se modifica la reserva.

Si la modifica un invitado el campo recepcionista va a estar en NULL y sino se carga la id del recepcionista correspondiente, al igual que al momento de generarla.

- **Cancelar Reserva:**

Se pide que se ingrese el número de la reserva, el motivo por el cual quiere cancelarla y en el caso de que sea un recepcionista se pide que se ingrese la Id del mismo (sino ese campo estará en modo solo lectura para que no se modifique). La fecha de cancelación es la fecha del sistema actual conseguida de la app config (14/06/2018).

Una vez completados todos los campos el botón "aceptar", genera la cancelación de la misma.

- **Facturar Estadías:**

En la misma se mostrarán todas las estadías en las cuales es posible su facturación. Una vez seleccionado alguna de las mismas, se muestra los consumibles que fueron, valga la redundancia, consumidos (botón buscar consumibles). Una vez mostrados todos estos campos y elegida la forma de pago se procede a hacer la facturación con los campos seleccionados.

Una vez hecha la factura se mostrara una leyenda con el cliente ID y el monto final a pagar.

Al finalizar se limpian todos los campos dejando la form lista para hacer otra facturación.

Si el modo de pago seleccionado es tarjeta de crédito a continuación se requerirá completar los datos de la misma.

En caso de que esté registrada no se cargará en la base, evitando redundancias, pero sí se asociará a la factura.

En caso de que se repita el número pero alguno de los otros datos difiera se lanzará una excepción.

- **Registrar Estadía:**

Al ingresar en esta sección se procederá con el ingreso (check-in) o egreso (check-out) del hotel.

En ambos casos se deberá ingresar el número de reserva para ver todas las habitaciones pertenecientes a la misma. Se deberá seleccionar una para proceder con el check-in o check-out de la misma presionando el botón "Registrar".

En ambos casos también para buscar una nueva reserva se debe presionar anteriormente el botón "limpiar".

- **Listado Estadístico:**

Al ingresar en esta funcionalidad se debe ingresar el año sobre el cual se desea hacer la consulta y luego de esto el mes de inicio y el de fin. Abajo hay un selector para seleccionar el tipo de listado que se desea ver. Para proceder con la obtención del listado se debe presionar consultar.

Los resultados son mostrados en una grilla.

Para ver otro listado se debe presionar el botón limpiar, el cual limpia los campos ingresados y la grilla de resultados.

- **Registrar Consumibles:**

Funcionalidad que permite registrar los consumibles de los huéspedes.

Es necesario ingresar el número de reserva y el número de habitación. A continuación se debe ingresar la descripción del consumible y la cantidad consumida.

Al aceptar figurará el nuevo consumible en una grilla con la opción de modificar o borrar el consumible ingresado.

- **ABM Hotel:**

Dentro de la sección de Baja del Hotel aparece una form para completar los datos del período de Cierre del mismo. Los campos son: Periodo_Id, Periodo_Hotel, Periodo_FechaInicio, Periodo_FechaFin y Periodo_Motivo.