

Gestión de Datos

Estrategia de Migración y manejo de datos en la aplicación

Integrantes

- **Acosta Naiara**
 - Legajo:146.701-3
 - Curso: K3021
- **Fadel Samanta**
 - Legajo: 113.391-3
 - Curso:K3171
- **Laredo Gonzalo**
 - Legajo: 146.587-9
 - Curso: K3173

2014

Índice

[Objetivo](#)

[Alcance](#)

[Estado de entrega 23-11-2014](#)

[Correcciones recibidas](#)

[Diagrama de Entidad de Relación](#)

[Inconsistencia o faltantes de información en la tabla Maestra](#)

[Estrategia de migración](#)

[Resolución de inconsistencias y faltantes de información](#)

[Consideraciones](#)

[Estrategia de manejo de Negocio en aplicación](#)

[Ingreso a la aplicación](#)

[Configuración de fecha de sistema](#)

[Validaciones Generales](#)

[Funcionalidades](#)

Objetivo

El objetivo del documento es especificar todas las consideraciones tomadas para la migración de los datos expuestos en la tabla maestra otorgada por la catedra de Gestión de Datos como así también indicar la estrategia tomada para el vuelco de información y su posterior, incluyendo reglas de negocio en una aplicación en el Lenguaje C# siguiendo las funcionalidad definidas.

Alcance

El alcance va desde indicar la estrategia tomada para la migración de los datos, indicando las consideraciones que correspondan, hasta detallar la estrategia para la gestión de los datos en la aplicación y las validaciones aplicadas.

Estado de entrega 23-11-2014

A continuación se detalla el contenido la presente entrega:

- Diagrama de Entidad de Relación (Se incluye en este documento y además una imagen del mismo dentro del zip para una mejor visualización)
- Estrategia (Presente documento). Se resaltó con el siguiente tag **[Nuevo]** las explicaciones incorporadas respecto a la versión anterior en la sección de “Estrategia de Migración” y “Estrategia de manejo de Negocio en aplicación”
- Script script_creación_inicial.sql el cual contiene:
 - Creación de Esquema con nombre del grupo
 - Creación de tablas (Se corresponden con el DER)
 - Creación de Procedures/Funciones
 - Inserts de migración de datos
- Aplicación con las siguientes funcionalidades:
 - Login y Seguridad
 - ABM Hotel
 - Generar o Modificar Reserva
 - Cancelar Reserva
 - Registrar Estadía (Check-In/Check-Out)
 - Registrar Consumibles
 - Facturar Estadía
 - Listado Estadístico
 - ABM Usuario
 - ABM Cliente
 - ABM Habitación
 - ABM Régimen (Como menciona el enunciado, no se desarrolla)
 - ABM Rol

Correcciones recibidas

A continuación se detallan las correcciones recibidas (Copia textual), y luego en cada caso se comento los componentes impactados.

#MIGRACION

Faltan migrar facturas y sus respectivos items

Se informó por mail que hubo un error en la prueba, y que las cantidades eran correctas.

Impacto: Ninguno.

#MODELO

Bastante bien pero tiene unos detalles muy importantes a remarcar como ser:

-Cancelación de reserva, debería ser una entidad por separado, porq? No todas las reservas se cancelan, lo que generaría que aquellas reservas que no se cancelan queden con campos en null, no siendo performante.

Se agregó la entidad RESERVA_CANCELADA, y se quitaron los campos USR_USERNAME_CANC , ESERVA_MOTIVO_CANC, RESERVA_FECHA_CANC de la tabla RESERVA.

Impacto:

- Modelo
- Procedure SP_CANCELAR_RESERVA que tiene la responsabilidad de cancelar la reserva.

-LA entidad estadia_hab_con debería relacionarse con los items de la factura. Porq? Porq de esta manera se que se item representa un consumible de esa estadia, se entiende el concepto? Sino quedaria como el item de facturación sale de la nada cuando en la mayoría de ellos son producto de consumible de una habitacion

Se agrego el campo ITEM_ID_EST_CON en la tabla ITEM_FACTURA y se agregaron los campos EST_HAB_CON_ID y CANTIDAD en la tabla ESTADIA_HAB_CON.

Impacto:

- Modelo
- Migración
- Procedure SP_REALIZAR_FACTURACION que realiza la facturación de la estadia
- Procedure SP_REGISTRAR_CONSUMIBLE que registra los consumibles de una estadia

-La factura debería tener una relacion al cliente, de manera que se pueda identificar de manera rápida a que cliente se le facturó, si bien podemos hacer esto haciendo joins hasta caer en el cliente, una buena practica nos dice que siempre una factura debe apuntar a quien se le imputa la misma, es mas, nos daría la facilidad de que en un futuro que a quien se le realice la factura no sea necesariamente a la persona que se alojo, como ser una empresa. Es moneda corriente que una persona se hospede en un hotel y a quien se le factura dicha estadia sea a una empresa en donde el cliente trabaja.

Se agrego el campo FACTURA_ID_CLIENTE en la tabla FACTURA.

Impacto:

- Modelo
- Migración
- Procedure SP_REALIZAR_FACTURACION que realiza la facturación de la estadia

#APLICACION

A modo de sugerencia, sería bueno que luego de terminado algun proceso de alta, modificación, baja o X se informe al usuario un todo ok, o lo que sea porq no se identifica si hizo el insert - update correspondiente
Se tomó la sugerencia para todas las pantallas desarrolladas.

Impacto

- Aplicación

ABM HOTEL

Si buscamos hoteles sin ningun parametro ingresado (para la busqueda) la grilla aparece vacia
Se realizó la misma acción, pero sin embargo se tuvieron resultados satisfactorios. Al buscar sin ningún parámetro la aplicación devolvió todos los hoteles.

Impacto:

CHECK IN - OUT

Como sugerencia le podrían agregar ademas de la busqueda por codigo de reserva, por nro de doc como para tener otro punto posible de entrada.

Se tomó la sugerencia

Impacto:

- Aplicación

FACTURACION

Se inteneto realizar una facturación sin haber realizado un check out y dio error. Si bien esto esta bien, no deberia haberme dejado continuar si dicha reserva o estadia no tenia un check out hecho.

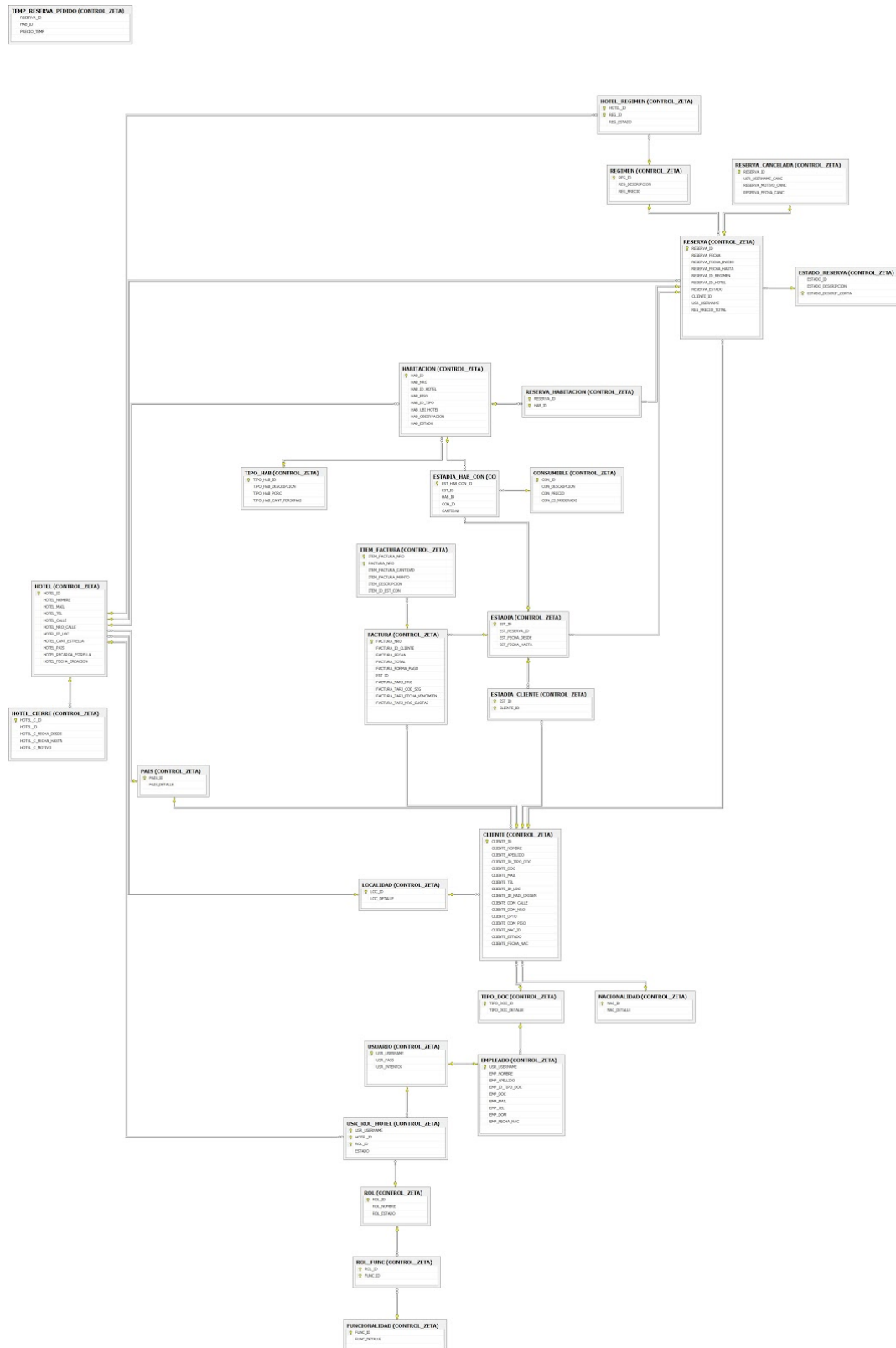
Una forma sencilla de validarlo es que solamente se pueda facturar luego de hacer un check out :P, de ultima, antes de redirigir del form de check out al de facturacion se emite un mensaje preguntado si se quiere cargar algun consumible antes de facturar y en funcio nde ello sigo con mi workflow

Se corrigió la validación, sin embargo no de modo sugerido sino por medio del Procedure que tenía la validación pero con un error sintáctico.

Impacto:

- Procedure SP_REALIZAR_FACTURACION que realiza la facturación de la estadia

Diagrama de Entidad de Relación



Inconsistencia o faltantes de información en la tabla Maestra

A continuación se detallan las inconsistencias encontradas en la tabla maestra

1. Diferentes clientes con pasaporte iguales. Ejemplo:

```
select
Cliente_Nombre,Cliente_Apellido,Cliente_Fecha_Nac,Cliente_Pasaporte_Nro
from gd_esquema.Maestra
where Cliente_Pasaporte_Nro= 1652782
```

Uno de los casos devueltos

MAYDI Paz	1922-06-18 00:00:00.000	1652782
REBECA Fernández	1981-06-19 00:00:00.000	1652782

2. Mails de clientes con formato incorrecto. Ejemplo:

```
select top 1 Cliente_Nombre,Cliente_Apellido, Cliente_Mail
from gd_esquema.Maestra
where Cliente_Nombre='ALINA ALIRA'
and Cliente_Apellido='Flores'
```

ALINA ALIRA Flores	alina alira_Flores@gmail.com
--------------------	------------------------------

3. Los hoteles no poseen nombre.

4. Diferentes clientes con un mismo mail, siendo una regla de negocio que los mails de los clientes deben ser únicos. Ejemplo:

```
SELECT DISTINCT UPPER(CLIENTE_APELLIDO + ', ' + CLIENTE_NOMBRE ) AS
NOMBRE,
CLIENTE_PASAPORTE_NRO,
CLIENTE_MAIL,
COUNT(*)
FROM GD_ESQUEMA.MAESTRA M
GROUP BY UPPER(CLIENTE_APELLIDO + ', ' + CLIENTE_NOMBRE ),
CLIENTE_PASAPORTE_NRO,
CLIENTE_MAIL;
```

Uno de los casos devueltos por la consulta

ACOSTA, AIKE 70183321	aiko_Acosta@gmail.com
ACOSTA, AIKE 92651060	aiko_Acosta@gmail.com

5. Se encontraron diferentes casos/estados en que se encuentran las reservas en la tabla Maestra

- Reservas con fecha de inicio futura que poseen factura, se consideran Reservas inconsistentes .
- Reservas en curso con factura, se consideran Reservas inconsistentes
- Reserva en curso sin factura, reservas consistentes.
- Reserva finalizadas con Factura, reservas consistentes.
- Reservas finalizadas sin Factura, se consideran Reservas inconsistentes.
- Reservas con fecha inicio futura sin factura, reservas consistentes.

Estrategia de migración

Resolución de inconsistencias y faltantes de información

1. Los Clientes con pasaporte duplicado serán migrados como clientes distintos, luego en la aplicación al intentar consultar clientes que contengan pasaportes duplicados se mostrará un aviso indicando la situación que se presenta. Este campo en la tabla clientes no será Primary Key, ni la tabla tendrá una constraint de unique para dicho campo.
2. Los Clientes con mail duplicado serán migrados, como si sus mails fuesen distintos, considerando que luego en la aplicación se mostrará un aviso, para dar a conocer la situación que se presenta. Este campo en la tabla clientes no será unique.
3. Los campos que se solicitan en el enunciado pero que no se encuentran en la tabla Maestra, serán almacenados vacíos (null).
 - a. Respecto a esta situación se tiene un caso a destacar que es el nombre del hotel. Los hoteles en la tabla maestra no poseen nombre, se consideró que al migrarlos no se les incluirá nombre ficticio o genérico, sino que quedarán como null. En la aplicación para identificar los distintos hoteles se colocará la dirección (Ciudad, Calle, Número) ya que esto permite identificar unívocamente a un hotel.

Consideraciones

- Para la migración de Reservas de acuerdo a los diferentes casos/estados encontrados para las reservas se definió agregar nuevos estados que pueden tener la reserva además de los que menciona el enunciado, los mismos son:
 - RESERVA SIN FACTURA. Código: RSF
 - RESERVA INCONSISTENTE. Código: RINC
 - RESERVA EN CURSO. Código: REC
 - 'RESERVA FINALIZADA SIN FACTURA. Código: RFSF

A continuación se detalla con qué estado se migrarán los diferentes casos de Reservas:

- Reservas con fecha de inicio futura que poseen factura. Se migra con estado 'RINC'.
- Reservas en curso con factura. Se migra con estado 'RINC'.
- Reserva en curso sin factura. Se migra con estado 'REC'
- Reservas finalizadas con Factura. Se migra con estado 'RI'
- Reservas finalizadas sin Factura. Se migra con estado 'RSF'
- Reservas con fecha inicio futura sin factura. Se migra con estado 'RC'

Nota: El estado RFSF no aplica al momento de la migración

- Para la migración de Facturas, ya que no se cuenta con los datos de tarjetas, o información de Pago se tomará que la Forma de Pago fue en Efectivo('E').
- Para el régimen All Inclusive Moderado se entiende que este incluye algunos de los consumibles pero no todos, por lo tanto se optó por poner en la tabla de CONSUMIBLE un campo (CON_ES_MODERADO) con una marca que indica si se incluye o no en el régimen All inclusive moderado. Esta decisión en la realidad no se tomaría arbitrariamente sino que sería consultada con el Usuario. En la migración este régimen incluye:
 - Coca Cola
 - Whisky
 - Agua Mineral

- Para cada tipo de habitación se decidió definir cuántas personas pueden ingresar en esa habitación, esto se incluye en un campo en la tabla *HABITACION*, y permitirá calcular el precio total de la estadía. El criterio para definir cuántas personas permite la habitación se realizó aplicando sentido común respecto al nombre del tipo de habitación.
- Los clientes serán migrados en estado Habilitado ('H')
- Tal como da a entender el enunciado, el detalle de la habitación que indica si la misma se encuentra al frente o no (Campo: *habitacion_frente* de la tabla Maestra), puede ser extendido a más posibilidades, por lo que al migrar la habitación si el campo mencionado tenía el valor 'S' se migra con valor 'FRENTE', en caso que tenga el valor 'N' se migra con valor 'INTERIOR'.
- **[Nuevo]** La tabla *ITEM_FACTURA* tiene relación con la tabla *ESTADIA_HAB_CON* la cual almacena los consumibles de cada estadía por habitación. Por medio de varios Join y calculos se puede obtener la cantidad consumida, monto total y descripción del consumible, sin embargo se decidió persistir estos campos en la tabla *ITEM_FACTURA* (Campos: *ITEM_FACTURA_CANTIDAD*, *ITEM_FACTURA_MONTO*, *ITEM_DESCRIPCION*) ya que se considera más performante tener esos campos en la tabla *ITEM_FACTURA_CANTIDAD* ya sea para el momento de calcular el monto total de la factura, como para en caso que se extienda la aplicación y se pida visualizar las facturas anteriores. Además en el caso de los montos redundar este valor permite que la factura quede consistente en caso que haya un ajuste en el precio de los consumibles.

Estrategia de manejo de Negocio en aplicación

Ingreso a la aplicación

Usuario Administrador

- Usuario: ADMIN
- Contraseña:w23e
- Rol: ADMIN
- ID Rol: 1

Ya que no se cuenta con la funcionalidad de alta de usuarios desarrollada en su totalidad, en la migración se creará un usuario Recepcionista de modo que se pueda contar con todos los roles incluidos en el enunciado

Usuario Administrador

- Usuario: RECEPCIONISTA
- Contraseña: recep
- Rol: RECEPCIONISTA
- ID Rol: 2

Configuración de fecha de sistema

La aplicación cuenta con una opción para los usuarios administradores que permite modificar la fecha del sistema, de todos modos la misma también puede ser tomada del archivo de configuración app.xml donde también se encuentran los datos de conexión a la base de datos. Esta opción se encuentra en la pantalla principal como "Opciones de Desarrollador"

Validaciones Generales

A continuación se detallan algunas validaciones que realiza la aplicación además de las que se requieren según indica el enunciado

1. En la tabla Maestra se encontraron pasaportes duplicados para diferentes clientes, la aplicación mostrará un aviso al consultar ese pasaporte, o cliente indicando la situación presente.
2. En la tabla Maestra se encontraron diferentes clientes con un mismo mail, la restricción de negocio es que este dato debe ser único para cada cliente, se mostrará un aviso en la aplicación cuando consulte los datos de un cliente con datos en esa situación, es decir que no cumplan la regla de negocio.
3. El Nombre, Apellido, Tipo de documento y Nro de documento del Cliente no puede ser nulo.
4. No se puede realizar un alta/modificación de un hotel sin al menos un régimen.

Funcionalidades

La lógica de negocio fue puesta en Procedimientos y Funciones, ya que esto nos permite modificar, y extender funcionalidad (En caso que fuese necesario) de una forma más simple y práctica. Teniendo, a su vez, una capacidad de procesamiento mayor a la que se puede tener desde la aplicación realizando las mismas acciones, ya que por medio de la aplicación se utilizan componentes para la comunicación a la base de datos, que aunque es mínimo, pueden tener overhead.

A continuación se detallarán algunas estrategias tomadas para ciertas funcionalidades críticas

- **Generar/Modificar Reserva:** Se cuenta con tres procedures que actúan en esta funcionalidad. Uno de estos verifica la disponibilidad y en caso que exista disponibilidad es cargado a una tabla auxiliar *TEMP_RESERVA_PEDIDO*. Si el usuario está de acuerdo con el precio y su pedido se procede a cargar la reserva afectada a la tabla *RESERVA* y *RESERVA_HABITACION* está última se carga a partir de la tabla auxiliar en donde se cargó el pedido.
- **Cancelar Reserva:** Sólo se pueden cancelar reservas que no fueron canceladas anteriormente. En caso que se quiera cancelar una reserva ya cancelada se informa al usuario.
- **Costo de reserva:**
 - *Costo de habitacion:* $(Cant_Pers * Tipo_hab_porc * Tipo_reg_precio)$
 - *Cant_Pers:* Cantidad de personas que se pueden hospedar en la habitacion.
 - *Tipo_hab_porc:* Porcentaje por el tipo de habitacion.
 - *Tipo_reg_precio:* Precio base del regimen.
 - *Costo por Noche:* $(Costo_de_habitacion + (Rec_estrella * Cant_estrella))$
 - *Rec_estrella:* Recargo por cada estrella que tenga el hotel al que se va a hospedar.
 - *Cant_estrella:* Cantidad de estrellas del hotel.
 - *Costo por estadia:* $(Costo\ por\ Noche) * cantidad_noches$
 - *Cantidad Noches:* se obtiene con la diferencia de *reserva_fecha_inicio* y *reserva_fecha_hasta*
- **ABM Usuario:** Un administrador sólo podrá dar de alta a un usuario para el mismo hotel en el que inició sesión.
- **Estados de Reserva**
 - Una reserva se da de alta con esta 'RC' Reserva Correcta
 - Si una reserva es modificada pasa a tener estado 'RM'
 - Al hacer el check in sólo se toman reservas que estén en estado 'RC' o 'RM' y la reserva pasa a tener estado 'REC'
 - Al hacer el checkout solo se toman reservas que estén en estado 'REC' y pasan a tener estado 'RFSF'
 - Luego para facturar, primero se debió hacer el check-out por lo tanto se toman únicamente facturas que estén en estado 'RFSF' y pasan a estado 'RI'

- **Cambio de Contraseña [Nuevo]** : Un administrador puede modificar la contraseña de cualquier usuario que pertenezca al hotel que esté loggeado.
La aplicación no le solicita al administrador la contraseña anterior ya que se entiende que ese rol tiene la facultad de “reiniciar” la contraseña en caso de que un usuario olvide su contraseña. Si existiese la funcionalidad de que los usuarios puedan modificar su contraseña, si se les solicitaría la contraseña anterior.
- **ABM de Clientes [Nuevo]** : La migración de datos cuenta con clientes inconsistentes, por lo tanto desde la aplicación se maneja esta situación de la siguiente manera:
 - Si un usuario de la aplicación quiere modificar o dar de baja (lógica) a un cliente que no es inconsistente la misma se realiza automáticamente con los datos brindados.
 - En caso que se quiera modificar o dar de baja (logica) a un usuario que es inconsistente respecto a otros, esto significa que posee el mismo tipo y número de documento que otro cliente o que posee el mismo mail que otro cliente, entonces no se realiza la modificación o baja.
Se muestra en una grilla todos los clientes que cumplen inconsistencia respecto al que se buscó y se le da la libertad al usuario para que solucione las inconsistencia. Al seleccionar un cliente de la grilla se realiza la modificación o se da de baja.
- **Facturación [Nuevo]**: Al realizar la facturación se tiene identificado los consumibles por habitación, por lo tanto en el detalle de la factura (items de la factura) se incluye en la descripción del ítem (si es consumible) el N° de habitación en donde fue consumido.