



Gestión de Datos 2013
K3022
Lunes – Miércoles Mañana

Clínica FRBA

Estrategia

Grupo: CUATRO_QUESOS N° Grupo: 20

Nombre y Apellido	Legajo
Darío Cappellini	143514-0
Jonathan Corallo	144.522-4
Hernan Maschwitz	144741-5
Julian Selser	143100-6

Universidad Tecnológica Nacional
Facultad Regional Buenos Aires



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Contenidos

	Página
Migración de datos	3
Controles adicionales	6
Desencadenadores	7
Aplicación Desktop	7



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Migración de datos

Se comienza la migración de los datos creando las tablas del DER. También se agregan las constraints de primary key y foreign key, las constraints para las columnas que no admiten nulos, constraints de unicidad y constraints "check" para los campos con dominio acotado.

La migración de los datos se realiza "de afuera hacia adentro", comenzando por las tablas que no poseen foreign keys, y pasando a las que sí las tienen una vez que las tablas referenciadas fueron creadas.

Planes médicos, medicamentos, especialidades médicas, tipos de especialidades médicas:

Se tomaron los datos que existían en la tabla maestra. Todos los datos necesarios para el modelo estaban disponibles.

Afiliados:

Se tomaron los afiliados existentes en la tabla maestra. Se autogeneró el número de afiliado siguiendo los lineamientos del enunciado. Se eligió arbitrariamente el número de DNI como orden para generar los números de afiliado.

Al tener un campo Paciente_DNI, se asumió que los números disponibles eran de DNI, y por lo tanto este número pasó a ser el número de documento, y el tipo de documento para todos los pacientes existentes es DNI.

Al no haber información de la familia, se supuso que todos los socios existentes son titulares y no tienen familiares a cargo. También se asumió que son solteros. Finalmente, se consideró que los afiliados existentes están activos.

Como no hay información del sexo, se dejó indefinido (valor '-'). Se calculó el número de consulta en base a las veces que ya se atendió en la clínica.

Médicos:

Se tomaron los médicos existentes en la tabla maestra. Se introduce un número ID_Médico autogenerado como primary key para los médicos, similar a lo que ocurre en cualquier empresa con los números de empleado.

En cuanto al sexo y al documento, se siguió el mismo criterio que para los afiliados.

El número de matrícula se mantiene indefinido (en este caso 0) ya que esa información no está disponible.



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Se infiere el período de atención del médico: atiende desde la fecha de la migración hasta dentro de un mes. Si hubiera turnos más allá del mes, su período de atención se extiende hasta ese momento. Se verificó que este criterio no genera períodos de atención mayores a 120 días.

También se asumió que todos los médicos que aparecen están activos.

Bonos:

Se asume que los bonos se general al ser comprados, y por lo tanto todos los bonos que aparecen en la tabla fueron comprados. Esto se ve confirmado por el hecho de que todos tienen un número de afiliado asociado. Sin embargo, no hay información sobre cuándo se compraron, por lo que el campo ID_Compra se deja sin información (NULL).

Se eligió una sola tabla que contenga tanto los bonos consulta como los bonos farmacia, con una PK doble: número de bono y tipo de bono.

Se verificó que Compra_Bono_Fecha, cuando contiene datos, es igual a la fecha de impresión de cada bono (consulta o farmacia). Además, se verificó que siempre que un bono está asociado a un turno, esa fila también tiene información sobre los síntomas, enfermedades y medicamentos. Por lo tanto, un bono para el cual existe una fila con número de turno, significa que ese bono se usó en una consulta. A partir de esto se calcula el Nro_Consulta que sólo aplica para los bonos consulta.

Para los bonos farmacia, se supone que la persona que la usó es la misma persona que lo compró, ya que no hay forma de suponer lo contrario. La fecha de vencimiento se mantuvo inalterada, por más que no sea a 60 días de la fecha de impresión, porque existe la posibilidad de que antes existieran otras reglas para determinar esa fecha. Además, recalcular la fecha de vencimiento significaría perder datos.

Días de atención:

Se calcularon los días de atención en base a los turnos existentes en la base de datos. Se supuso que un médico atiende un día en el que tiene turnos, y atiende desde el primer turno hasta finalizado el último, sin importar si tiene huecos en el medio. Sin embargo, no se tomaron en cuenta como días de atención aquellos que no cumplen con los horarios de la clínica. Se verificó que, tomando este criterio, la carga semanal de los médicos existentes en ningún caso supera las 48 horas.

Especialidades por médico:

Para completar esta parte del modelo se observó que en algunos turnos un médico aparece con una cierta especialidad, pero puede aparecer con otra en otros. Por lo tanto, la tabla Especialidades_Médicos contiene las filas que permiten consolidar esta relación muchos a muchos.

Turnos:



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Se tomaron los turnos existentes en la tabla maestra. Aunque algunos turnos debieron ser cancelados, en ningún caso dejan de aparecer en la tabla Turnos. Un turno cancelado es aquel turno que aparece tanto en Turnos como en Cancelaciones.

Cancelaciones:

Se cancelaron los siguientes turnos encontrados en la tabla maestra:

-Turnos en el pasado que nunca se concretaron: turnos que tienen fecha anterior a la migración, pero no hay una fila que lo asocie con un bono consulta - esto significa que nunca se concretó. Se dieron por cancelados ya que debieron haberse concretado en el pasado y no fue así.

-Turnos no permitidos en el futuro: turnos que tienen fecha futura pero no caen dentro de los horarios de funcionamiento de la clínica.

Consultas:

Se toman las consultas de las filas que tienen tanto número de turno como número de bono consulta no nulos. La tabla maestra todos los datos necesarios para completar esta parte del modelo.

Las consultas que ocurrieron en el futuro se pasan a una tabla especial Consultas_Exceptions, que no se relaciona con el resto del modelo, y cuyo propósito es simplemente almacenar estos datos que no parecen tener sentido para su posterior tratamiento. Los bonos utilizados en estas consultas futuras se dan como no aún utilizados.

Agenda profesional:

Se creó la vista Agendas_Medicos, que muestra, para cada médico, todos los posibles horarios en los que puede trabajar, tomando en cuenta sus fechas de inicio y de fin, sus días y sus horarios de atención.

Se recurrieron a dos tablas auxiliares para generar los horarios del día y los distintos días. Una vez generados, su producto cartesiano muestra todos los posibles intervalos de media hora. Una vez obtenido esto, se filtra con los datos específicos del médico (fechas y horarios de atención) y, finalmente, se verifica en cuáles de esos intervalos el médico tiene un turno. Así se determina cuándo está libre para atender y cuándo está ocupado.

Recetas:

Las recetas están representadas en la tabla Medicamentos_Por_Consulta. Se toman los medicamentos obtenidos de las consultas existentes en la tabla maestra, suponiendo que se recetó sólo una instancia de cada uno de éstos.



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Usuarios:

Se crearon usuarios para todos los afiliados y los Médicos.

El formato para los afiliados es #A donde # es el número de afiliado (longitud variable) y A es la letra A.

El formato para los médicos es nna#M donde nn son las primeras dos letras del nombre, a es la primera letra del apellido, # es el ID_Medico (longitud variable) y M es la letra M.

Para todos ellos la contraseña es UTNFRBA.

También se creó el usuario admin, al que se le asignó el rol de administrativo, con la contraseña w23e especificada en el enunciado.

Controles adicionales

Se implementaron las siguientes constraints adicionales:

- limitarAgendaProfesional sobre la tabla Medicos, verifica que la diferencia entre fecha_hasta y fecha_desde, es decir, la longitud de su agenda, no exceda los 120 días.
- verificarCantidadMedicamentoIndividual sobre la tabla Medicamentos_Por_Consulta, verifica que no se receten más de tres instancias del mismo medicamento para una misma receta.

Los siguientes triggers también implementan controles:

- limitarMedicamentosPorBono sobre Medicamentos_Por_Consulta, evita que se receten más de 5 medicamentos en un solo bono.
- limitarJornadaLaboral sobre Dias_Atencion, no permite a un médico trabajar más de 48 hs semanales.
- obedecerHorariosClinica sobre Dias_Atencion, no permite a un médico atender fuera de los horarios de la clínica.



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Desencadenadores

Los siguientes triggers implementan otros requerimientos del enunciado:

- quitarRolInhabilitado le quita un rol inhabilitado a todos los usuarios que lo posean.
- revocarUsuario inhabilita a un usuario que acumule tres intentos de login fallidos.
- cancelarTurnosMedicosDesvinculados cancela todos los turnos de un médico que sea dado de baja.
- cancelarTurnosPacientesDesvinculados cancela todos los turnos de un afiliado que sea dado de baja.

Aplicación Desktop

Para la aplicación en C#, inspirados por el patrón de diseño MVC (modelo-vista-controlador), se decidió dividir el proyecto en carpetas cada una conteniendo clases separadas por su distinta índole:

- Una carpeta por cada una de las ABM y funcionalidades en específico con sus elementos de vista (ABM de afiliado, ABM de profesional, ABM de rol, Cancelar Atención, Compra de Bono, Generar Receta, Listados Estadísticos, Pedir Turno, Registrar Agenda, Registro de Llegada, Registro Resultado Atención)
- Una carpeta Domain, que contiene las clases de los objetos de negocio (profesionales, especialidades, afiliados, etc) que son objetos que carecen de comportamiento pero tienen estado, que tienen su utilidad para almacenar en memoria valores de variables en sus atributos y poder transaccionarlos entre las vistas para operar, dar de alta, baja, modificar, entre otras cosas.
- Una carpeta AppModel, que contiene las clases de los objetos App, que tienen por responsabilidad ser los data access objects, acceden a la base de datos y traen Data Tables, valores de retorno o ejecutan procedures que hacen insert, delete o update, agrupando cada uno por el ABM u objeto de negocio al que se enfocan y son responsables de gestionar en la persistencia. Estos objetos App heredan del conectorSQL a la base de datos, que utiliza el archivo App.config para obtener el string de conexión. Posee métodos para ejecutar procedures o chequear existencias, que luego son especificados en cada App.



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

En esta carpeta también se encuentran objetos de uso común entre las vistas para ser reutilizables, como exceptions definidas por usuario, una clase Globales que tiene un método para obtener la fecha del sistema del archivo de configuración App.config, un listado de búsqueda de profesionales para el ABM de profesional, pedir turno y registro de llegada, entre otros.

Además se tiene una clase Controlador que se encarga de validar los inputs de los campos, que sean del tipo de dato esperado, o validar que los campos obligatorios estén completos, en caso contrario, devuelve una excepción.

ABM Afiliados

Para este ABM se creó una pantalla principal que tiene las funcionalidades de alta, baja y modificación para seleccionar.

-La funcionalidad de alta es una vista con campos a completar requeridos para el alta de un afiliado, un botón de limpiar para blanquear los campos a su valor inicial, en caso de existir un afiliado con los mismos datos aparece un dialog emergente con dicha información, en caso de ser correcto se abre un dialog con un feedback que dio de alta con el número de afiliado que le fue asignado. Luego si el afiliado tenía cónyuge o familiares a cargo, se inicia una nueva ventana intermedia que da la posibilidad a dar de alta a dichos potenciales afiliados, esto se puede omitir haciendo click en finalizar.

-La funcionalidad de baja y modificación, ambas reutilizan una vista de listado de afiliados, con sus respectivos datos por columna que, dependiendo de cual funcionalidad se seleccionó, mapea un botón para cada fila con la funcionalidad especificada para cada afiliado mapeado en la grilla, abriendo una nueva ventana.

El listado tiene campos que pueden quedar en blanco o no, para poder filtrar de forma combinable. Se puede buscar por nombre y apellido "contiene", y también por nro afiliado raíz, lo cual devuelve a todo el grupo familiar de un titular.

También se tiene un botón limpiar para blanquear los campos. Si el afiliado esta dado de baja, no permite utilizar ninguna funcionalidad de baja o modificación, mostrando un dialog con dicha información en caso de que sucediere.

-La funcionalidad de baja pide una confirmación del afiliado seleccionado, luego la baja que se realiza es lógica, asignándole al afiliado una fecha de baja en el respectivo campo, utilizando la fecha de sistema del archivo de configuración.

-La funcionalidad de modificación muestra nuevamente la ventana intermedia de petición, dando la posibilidad de dar de alta un cónyuge o familiar a cargo si no se hizo en el momento luego del alta el afiliado titular. En caso de modificar al afiliado, inicia una nueva ventana similar a la de alta,



Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

con los campos que pueden ser editados inicializados con los datos que ya tenía dicho afiliado almacenados en la base de datos.

ABM Profesional

Para este ABM se creó una pantalla principal que tiene las funcionalidades de alta, baja y modificación para seleccionar.

Al igual que el ABM de afiliados, la funcionalidad de alta cumple con un comportamiento similar, provee una checkedListBox para poder seleccionar las especialidades del médico y luego dar sus respectivas altas en la tabla de especialidades por médico.

Las funcionalidades de baja y modificación reutilizan una vista de listado de profesionales, que también se reutiliza en la vista de Pedir Turno y Registro de Llegada.

Cancelar Atención

En esta pantalla principalmente se necesita visualizar una grilla con datos sobre los turnos existentes en la base según el usuario.

Aprovechando que el comportamiento es muy parecido salvo cuando además haya que sumar algún criterio a la búsqueda cuando se trate de un profesional (elegir un periodo para cancelar) o un administrativo (elegir el afiliado o médico) se usó una clase abstracta base.

La clase abstracta define en líneas generales que debe haber una grilla y botones, delegando en las subclases que la implementen el deber de saber que datos traer y con criterios.

Hay dos posibilidades para cancelar los turnos mostrados, seleccionando un día en particular o un periodo (para profesional y administrador), debiéndose especificar el motivo en un cuadro de texto. Nuevamente las ventanas comparten casi todo sus rasgos, en este caso siendo visualmente las mismas, variando el comportamiento en algunos parámetros, por eso se vuelve a usar una clase abstracta que define en su mayoría los dos casos, introduciendo las subclases las pequeñísimas variaciones de código.

Adicionalmente se cuenta con una clase que usa el patrón factory para saber cuál de las clases instanciar según el usuario para alivianar la cantidad de lógica en la pantalla principal.

Gestión de Datos		
Lunes – Miércoles Mañana	Clínica FRBA	Grupo:CUATRO_QUESOS

Login

La aplicación recibe a quien accede al sistema por primera vez con una pantalla que tiene como único botón el que da la posibilidad de loguearse, que lo llevará a la típica pantalla de ingreso de usuario y contraseña.

Luego se consulta la base de datos mediante una función con valores de retorno que indican si el login fue exitoso o en caso negativo la cantidad de intentos que le quedan vía aplicación. Dicha función incrementa la cantidad de intentos hasta que un trigger deshabilita al usuario si llega al máximo, ante un usuario deshabilitado se devuelve el código de error correspondiente y se le informa al usuario por pantalla. De ser el login exitoso se procede a traer los roles del usuario (no se puede traer la persona ya que puede no existir en el caso del administrador) y se permite elegir con que rol ingresar. Ahora sí, dependiendo del usuario se traen de la base todos los datos de la persona con las funcionalidades para su rol, cargándose en un objeto que sigue el patrón singleton.

Nuevamente en la pantalla principal se puede notar que los botones mostrados son los correspondientes a la funcionalidad del rol adecuado, puede observarse el nombre del usuario y se da la opción de desloguearse. Esto se logra mediante un "FlowLayout" y habilitando o deshabilitando visibilidades correspondientemente, permitiendo a la pantalla expandirse para que entren todos sus botones.

