



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	2
Desencadenadores	7
Aplicación Desktop	8

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.

Para evitar inconsistencias con el tema de las fechas, se crea una función FechaMigracion al principio del script, con una fecha futura en 6 de Junio de 2014, que será tomada en la migración como la fecha en la que se realiza la misma, para evitar anomalías como consultas futuras concretadas, entre otras cosas.

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.

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

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

En cuanto a su período de atención, se mantiene sin información (NULL) hasta que registre su agenda.

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.

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.

Compras de bonos:

Se infirieron los datos de las compras que ya ocurrieron, previas a la migración. Cada bono cuenta con un campo Compra_Bono_Fecha. Asumimos que, cuando en dos bonos coinciden Compra_Bono_Fecha y Paciente_Dni (que al momento de la migración es único), entonces esos bonos se adquirieron en una misma operación, ya que no es posible que una misma persona realice dos operaciones de compra a la misma fecha y hora exactamente. Se obtuvieron la cantidad de bonos farmacia y bonos consulta utilizando subconsultas, y agrupando por los dos campos mencionados. Utilizando el número de plan del socio se obtienen los importes unitarios de los bonos, lo que permite calcular el monto de la compra. Se generó un ID_Compra autonumérico.

Una vez generadas las distintas compras, se regresa sobre la tabla de bonos y, utilizando los mismos campos que permitieron identificar compras distintas, se busca en la tabla de compra el ID

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

compra que corresponde a cada bono. Entonces, podemos saber en qué compra se adquirió cada uno de los bonos.

Agendas de los médicos:

Para esta parte del modelo consideramos que, al finalizar la migración, ningún médico tiene agenda, sino que deberá registrarla mediante el caso de uso correspondiente, como lo indica la consigna. Lo que sí sabemos es que días atiende habitualmente (Dias_Atencion) pero no sabemos la fecha de inicio y de fin (estos campos están en la tabla médico). Además, la consigna sugiere que la agenda es única, y que no es modificable. Por lo tanto, modelamos agendas únicas por profesional, y no modificables.

La tabla Agendas_Medicos, por lo tanto, guarda para cada médico todos los timeslots de media hora que corresponden a sus días de atención durante su período de atención, con una columna indicando el número de turno que tiene a esa hora, o NULL si está libre. Cada fila puede identificarse con el *id del médico y la fecha y hora*, ya que la agenda es única. No hay ningún ID de agenda porque es única por médico, es decir que con el ID de médico basta para conocer su agenda.

Una vez generadas las agendas mediante el caso de uso, la fecha de inicio y de fin se puede consultar en la tabla médico, los días de la semana con sus horarios en Dias_Atencion, y la totalidad de los timeslots resultantes (la agenda en sí) en la tabla Agendas_Medicos.

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.

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.

Agenda profesional:

El enfoque que tomamos para esta parte del modelo es que, apenas finalizada la migración, ningún médico tiene agenda. No se sabe qué días va a atender ni a qué hora, por lo que no se pueden pedir turnos con ese médico. El médico (o un administrativo en su nombre) debe ejecutar el caso de uso "Registrar agenda profesional" para generar, por única vez, la agenda que utilizará. De acuerdo con los lineamientos del enunciado, no se tomó en cuenta la posibilidad de tener múltiples agendas distintas, ni tampoco la posibilidad de realizar modificaciones una vez registrada la agenda.

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.

Fuera de la migración de datos, se crearon las siguientes tablas necesarias para operar con la aplicación.

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.

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

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.

Roles:

En la tabla Roles se crearon tres registros con los diferentes roles que se especifican en el enunciado (Administrativo, profesional y afiliado) se le asigna un valor numérico autogenerado como primary key y se asume que estos roles se encuentran habilitados.

Funcionalidades:

En la tabla Funcionalidades se creó un registro por cada una de las funcionalidades que se podrían realizar, también se les asignó un valor numérico como primary key.

Funcionalidades por roles:

La tabla Roles_Por_Funcionalidades se tuvo que crear por la relación muchos a muchos que se daba entre los roles y las funcionalidades, esta tabla va a tener foreign keys hacia las tablas Roles y Funcionalidades, que a su vez van a formar su primary key compuesta.

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.
- 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.
- tr_updateNroConyuge actualiza el número de conyuge, utilizando el procedure *updateNroConyuge*
- tr_cancelarTurnosAgenda cuando se cancela un turno, deja disponible el timeslot de la agenda que ocupaba el mismo.
- tr_up_after_consulta para actualizar los datos del afiliado que son factibles de ser modificados.
- tr_updateAgenda cuando se da de alta un turno, ocupa el timeslot en la agenda en la cual se dio de alta.

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

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.

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, para esto se decidió utilizar expresiones regulares y que los contenidos de texto matcheen con el tipo. También valida que los campos obligatorios estén completos, en caso contrario, devuelve una excepción, que será catcheada y se mostrará un mensaje informando la misma en el textbox errorBox.

El archivo de configuración (App.config) es el que lleva el string de conexión con todos los datos necesarios para conectar con la base de datos GD2C2013 y además contiene la fecha de sistema

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

que será tomada en cuenta como fecha actual en el sistema para la aplicación. Para esta fecha se decidió elegir una fecha posterior a la posible fecha de migración, ambas seteadas en el futuro

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.

Por cuestiones de performance, para no saturar la memoria de la aplicación, se decidió que la tabla que devuelve la consulta SQL contenga solo un número de 100 afiliados como máximo.

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

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

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.

ABM Rol

Para este ABM se creó una pantalla principal que tiene las funcionalidades de alta, baja y modificación para seleccionar. Para la funcionalidad de alta se creó una vista con un textBox en el cual se le tendrá que ingresar obligatoriamente un nombre solo de letras para que se habilite el botón aceptar, también tendrá un checkedListBox del cual será obligatorio seleccionar alguna funcionalidad para poder dar de alta el rol. Para la funcionalidad de baja y modificación se comparte una vista con un listado con los roles actuales para poder seleccionarlos, también cuenta con un filtro para buscar a los roles por su nombre y un botón limpiar para borrar los campos del filtro. En el caso de haber ingresado para una baja, no se permitirá seleccionar del listado un rol que ya se encuentre inhabilitado. Una vez seleccionado el rol del listado, para la baja de ese rol nos pedirá una confirmación mediante un botón aceptar. Para la modificación al igual que en alta, va a ser obligatorio que tenga un nombre y al menos una funcionalidad para poder realizarla; también habrá un botón que permitirá volver a habilitar a un rol antes inhabilitado, si el rol ya se encontrase habilitado, este botón no se podrá utilizar. Las vistas de alta y modificación también contarán con un botón limpiar que dejara vacío el textBox para el nombre, y te dejara el checkedListBox sin ninguna funcionalidad seleccionada. Si se modifica el rol que se está usando actualmente, al volver al menú principal se actualizarán las funcionalidades que ese rol tiene permisos para realizar.

Cancelar Atención

En esta pantalla principalmente se necesita visualizar una grilla con datos sobre los turnos cancelables existentes en la base según el usuario, evitando mostrar los ya cancelados.

Aprovechando que el comportamiento de las cancelaciones es muy parecido para todos los usuarios salvo cuando además haya que sumar algún criterio a la búsqueda cuando se trate de un

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

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 para introducir motivos (de periodo o día) 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 cuyo principal objetivo es reducir el código en la pantalla principal.

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.

Compra de bono

Esta funcionalidad permite comprar bonos de consulta, bonos de farmacia o ambos. Como lo detalla el enunciado, pueden utilizarlo tanto los afiliados en sí como los administrativos en nombre de un socio (en el caso, por ejemplo, que el socio realice la compra por teléfono).

Cuando la utiliza un afiliado, esta ventana se carga automáticamente con sus datos: descripción

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

del plan médico, precio de bono consulta y precio de bono farmacia. Luego, en dos TextBox se puede ingresar la cantidad de bonos a comprar de cada tipo. La aplicación valida que se ingresen cantidades numéricas enteras. No se permite comprar un total de 0 bonos. Para las TextBox se agrega una restricción arbitraria de 4 dígitos por cuestiones estéticas y de control: aunque los requerimientos hablan de compras sin límite esto no se puede implementar por limitaciones físicas, así que elegimos imponer una limitación razonable (se pueden comprar hasta 9999 bonos de cada tipo por compra) por nuestra parte.

Cuando la utiliza un administrador, esta pantalla es más compleja. Se le da un lugar para ingresar el número de afiliado para el cual va a comprar, como explica la consigna. Con un botón buscar se valida que se haya ingresado un número entero positivo, y se procede a buscar a ese afiliado en la base de datos. Si no existe, se lo informa al usuario. Si existe, se cargan todos sus datos y se puede proceder con la compra normalmente. Además, se ofrece un botón “quitar” para permitir resetear el afiliado y cargar uno nuevo.

Al presionar el botón de comprar, se realiza la compra, invocando al procedimiento correspondiente de la base de datos para su actualización. Se generan bonos según la cantidad comprada y se registran los datos sobre la compra. Por pantalla, se le informa al usuario cuál fue el monto total de la compra.

Registrar agenda

Esta funcionalidad permite registrar la agenda profesional de un médico. Es de destacar que, al momento de realizar la migración, se consideró que no conocemos la agenda del médico, por lo que esta funcionalidad siempre debe ejecutarse antes de pedir el turno con un médico. Lo que sí se infirió de la base de datos fueron los días en los que suele atender (tabla Dias_Atencion). Por lo tanto, cuando un médico ingresa a esta funcionalidad, se le ofrecen como default sus horarios inferidos. Sin embargo, puede modificarlos como quiera mediante los botones “agregar día”, el botón “quitar” de cada día y el botón “modificar” de cada día. Todos estos controles son botones o ComboBox, por lo que el input del usuario está cuidadosamente controlado para que no exceda los horarios de funcionamiento de la clínica. Cada cambio en un día conlleva una modificación en la base de datos (Días_Atención); la base de datos informa si alguno de los cambios no está permitido (si el médico excedería las 48 hs laborales), y la aplicación le transmite esto al usuario. Una vez conforme con los días, el usuario debe ingresar a “cargar período” para completar el proceso. Allí se encuentra con dos calendarios que le permiten elegir la fecha de inicio y de finalización de su agenda. Se valida que sea un intervalo positivo, que no exceda los 120 días y que no sea anterior a la fecha del archivo de configuración. Una vez seleccionado el intervalo, el usuario puede pulsar el botón de “registrar agenda”. Con este botón se genera finalmente la

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

agenda: se completa la tabla Agendas_Medicos en la base con los datos de este médico. Es de destacar que esta funcionalidad se ejecuta *una única vez* para cada médico, ya que el enunciado aclara que la agenda no es modificable, y sugiere que es única. Por lo tanto, nuestra decisión es que cada médico tiene una agenda *única, que se puede cargar sólo una vez*. Al pulsar “registrar agenda”, la aplicación le advierte al usuario que esta acción no puede revertirse. Si un médico que ya tiene una agenda registrada ingresa a esta funcionalidad, se le muestra un mensaje informando que no puede realizar esta operación. Finalmente, esta funcionalidad también se le ofrece al administrativo, con el propósito de que toda la aplicación pueda probarse desde un mismo usuario. Para esto, se aplica un mecanismo similar al utilizado para cargar a un afiliado en “compra de bono”, pero con los datos relevantes del médico.

Pedir Turno

Para esta funcionalidad se decidió crear una pantalla en la cual se selecciona el afiliado (deriva en otra pantalla de listado de afiliados que se reutiliza) y se selecciona profesional (deriva en otra pantalla de listado de profesionales, que solo podrá seleccionarse el afiliado si se filtra por la especialidad por la cual se quiere atender el paciente, caso contrario muestra un mensaje de error).

Luego, una vez seleccionado el profesional, se habilita el comboBox con sus fechas de atención disponibles. Seleccionando una fecha, luego se habilita el comboBox con los horarios disponibles de esa fecha.

Una vez completado estos campos, se habilita el botón Aceptar, el cual generará el turno, mostrando una MessageBox si la operación fue realizada correctamente, con los datos proporcionados.

También se tiene un botón limpiar que blanquea todo los campos y vuelve la ventana al estado inicial.

Registro de Llegada

Esta funcionalidad permite al administrador registrar el horario de llegada de un afiliado. La vista consiste en una grilla y campos de búsqueda: afiliado y profesional, ambos al seleccionar llevan a otra vista de listados correspondientes que se reutiliza para otras vistas, que permite seleccionar el afiliado o el profesional, según el caso. Al completar el campo profesional muestra todos los afiliados que tenían turno para ese día con el mismo pero no deja realizar la acción de registrar la llegada hasta que se especifique el afiliado para confirmar que sea correcto. Luego, una vez completos los campos aparece una nueva columna con un botón al final de la fila

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

que permite realizar la acción de registro, que lleva a otra ventana en la que se puede seleccionar el bono de consulta que desea utilizar para la misma.

Los bonos que se pueden elegir son los que coincidan con su plan médico y que hayan comprado en su grupo familiar.

Registro de Resultado de Atención

La vista consiste en una grilla en la que se mostraran las consultas existentes en la base de datos Mostrando por afiliado, profesional, número de turno al cual pertenece esa consulta, el número de consulta, las enfermedades, los síntomas, la fecha del turno y la fecha en que llego dicho afiliado a la clínica.

La búsqueda se puede especificar con solamente el profesional y la fecha de la atención en la que se realizó la consulta, pero solo se mostrará el botón para realizar la acción de registro si se especifica el afiliado también, como en la funcionalidad de Registro de Llegada. También se puede seleccionar si se desea mostrar solo las consultas en que no se ha registrado las enfermedades o los síntomas.

La nueva pantalla que se muestra al seleccionar el botón permite hacer modificaciones del diagnostico si ya tenia cargado previamente alguna enfermedad o sintoma, en este caso, carga en los textbox con esa información.

Si no fue registrado nada todavía, los campos estarán en blanco.

Para el momento en que se registra el resultado de la atención se decidió que pueda ser en cualquier momento después de registrarse la llegada.

Receta Médica

Para la funcionalidad de generar receta se creó una vista que tiene 5 campos textbox, dada la restricción del enunciado de tener esta cantidad como máximo, estos que contendran la descripción del medicamento seleccionado.

Para seleccionar el medicamento se creó una vista con grilla que filtra por campos de búsqueda de número de medicamento y descripción que tenga la cadena de texto especificada. Se creó una columna con botones para seleccionar la fila del medicamento a elegir.

Una vez seleccionado el medicamento, se permite determinar su cantidad (entre 1 y 3, según las restricciones).

Cuando se elige un medicamento más de una vez en la misma receta, no se permite generarla y muestra un mensaje en el TextBox errorBox con una leyenda descriptiva sobre esto.

También para generar la receta se debe especificar el número de consulta al cual corresponde dicha prescripción médica, para esto se delega en otra ventana de búsqueda con una grilla de consultas según afiliado y profesional, la misma que se reutiliza en otras vistas. Luego se puede elegir el bono de farmacia a utilizar de los que tenga el afiliado en su grupo familiar y que coincidan con su plan médico.

Una vez completado los requisitos necesarios, se habilita el botón Aceptar para generar la receta y

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

ser persistida con los respectivos datos y la fecha en que se realizó dicha acción en el bono de farmacia utilizado.

Si se desea recetar más de 5 medicamentos distintos o una cantidad mayor de 3 para un medicamento, se debe generar una receta para la misma consulta pero con otro bono.

El botón Aceptar, que realiza la acción de generar la receta, solo estará habilitado cuando este especificado el número de consulta, el bono a utilizar y al menos un medicamento con su respectiva cantidad. En el caso de haber medicamentos sin sus respectivas cantidades, el botón permanecerá inhabilitado hasta que se especifiquen las mismas.

Para poder limpiar un campo de medicamento individualmente, alcanza con hacer click en Seleccionar Medicamento y cancelar la ventana de búsqueda de medicamentos, esto blanqueará el campo y su cantidad.

Solo se podrá recetar medicamentos para aquellas consultas en las que se haya registrado el resultado de la atención.

Listados Estadísticos

El requerimiento es mostrar 4 tipos de listados a elegir según el año y semestre, pudiendose desagregar mensualmente. El usuario se encontrará con la opción de elegir el tipo de listado, el semestre de un par de “combobox” y el año de un “datepicker”.

Para elegir el listado según el input del usuario se tiene un stored procedure que llamará al listado correcto, joinando por identificador hasta obtener la lista mostrada. En la base puede verse que hay un procedure por listado, son estos los que tienen la lógica para cumplir los requerimientos, sin embargo, independientemente de que tipo de listado se trate todos se muestran de la misma manera, por eso se necesita un stored procedure adicional que “arma” el listado que se verá en pantalla. Dicho procedure trata de manera polimórfica todos los listados, de manera de separar cómo se muestran los listados independientemente de cuál sea, apoyándose en otro procedure más, cuya única responsabilidad es discriminar que listado usar. Resumiendo, se tiene un procedure por listado con su lógica, otro para discriminar el tipo de listado y un último para mostrarlos según lo pide el enunciado.

Cada listado particular puede pedirse en cualquier intervalo entre meses de un año. A continuación la estrategia para armar cada listado individual:

- TOP-5 Especialidades con mas cancelaciones: Se toman todos los turnos y de aquellos, todos los que se encuentren cancelados (la cancelación es una entidad). Cada turno posee información que especialidad es la que atenderá el turno.

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

- TOP-5 Bonos farmacia vencidos por afiliado: Se envía por parametro a la base la fecha del sistema y con ésta se buscan todos aquellos bonos de farmacia cuya fecha de vencimiento sea menor a ésta y no hayan sido usados (se guarda el ID del usuario del bono, así que si no se usó éste será null).
- TOP-5 Especialidades con más bonos recetados: Como se dijo anteriormente, un turno dice cuál especialidad lo atenderá. Adicionalmente un turno debe convertirse en consulta para haber generado una receta, que será un bono de farmacia. Luego es fácil darse cuenta qué especialidad recetó un bono en la relación turno-consulta-bono.
- TOP-10 utilización de bonos comprados por terceros: Cada bono además de guardar el ID del afiliado que lo compre guarda el ID de aquel socio que lo usó, que debe ser miembro del grupo familiar. En un bono que haya sido usado por una persona que no sea el comprador directo diferirá en sus campos ID comprador e ID usuario (aunque nomás sea en 2 numeros). Solamente hace falta contar por cada bono si dichos campos difieren agrupándolos por comprador.

IMPORTANTE: En los datos proporcionados en la tabla maestra no existen bonos usados por terceros así que dicho listado estará vacío, de haber datos el listado funcionará correctamente. Se decidió no inventar información en la base para no adulterar la tabla “maestra”.