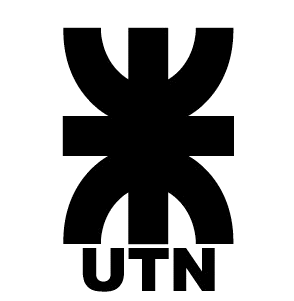
# UNIVERSIDAD TECNOLOGICA NACIONAL

## Facultad Regional Buenos Aires

### Ingeniería en Sistemas de Información

****

## Gestión de datos

#### Clínica FRBA

Curso: K3071

Integrantes:

DIAZ, ADRIAN

DAWID, DEMIAN

GOMEZ, FLORENCIA

ROJAS, CRISTIAN

Año 2013 – Segundo cuatrimestre

ClinicaFRBA – Estrategia

Contenido

[Introducción 4](#_Toc373003578)

[Acerca del modo de trabajo 4](#_Toc373003579)

[Acerca de la aplicación 4](#_Toc373003580)

[Decisiones tomadas al realizar la migración 4](#_Toc373003581)

[Modelo de datos 6](#_Toc373003582)

[Decisiones de desarrollo 7](#_Toc373003583)

[Arquitectura de la aplicación .NET 7](#_Toc373003584)

[Diagrama de componentes 7](#_Toc373003585)

[Diagrama de clases de acceso a datos 9](#_Toc373003586)

[Diagrama de clases de Capa de presentación 10](#_Toc373003587)

[Configuración de la aplicación 11](#_Toc373003588)

[Uso de la aplicación 11](#_Toc373003589)

[Gestión de clínica-> Roles 11](#_Toc373003590)

[Gestión de clínica-> Roles -> Botón Alta 12](#_Toc373003591)

[Gestión de clínica-> Roles -> Botón Modificación 12](#_Toc373003592)

[Gestión de clínica-> Roles -> Botón Baja 13](#_Toc373003593)

[Archivo -> Sesión -> Iniciar Sesión 13](#_Toc373003594)

[Archivo -> Sesión -> Cerrar Sesión 14](#_Toc373003595)

[Gestión de afiliados -> Afiliados 15](#_Toc373003596)

[Gestión de afiliados -> Afiliados -> Botón Alta 16](#_Toc373003597)

[Gestión de afiliados -> Afiliados -> Botón Modificar 18](#_Toc373003598)

[Gestión de afiliados -> Afiliados -> Botón Baja 19](#_Toc373003599)

[Gestión de profesionales -> Profesionales 19](#_Toc373003600)

[Gestión de profesionales -> Profesionales -> Botón Alta 20](#_Toc373003601)

[Gestión de profesionales -> Profesionales -> Botón Modificar 21](#_Toc373003602)

[Gestión de profesionales -> Profesionales -> Botón Baja 22](#_Toc373003603)

[Gestión de profesionales -> Especialidades médicas 22](#_Toc373003604)

[Gestión de profesionales -> Especialidades médicas -> Botón alta 22](#_Toc373003605)

[Gestión de profesionales -> Especialidades médicas -> Botón modificar 23](#_Toc373003606)

[Gestión de profesionales -> Especialidades médicas -> Botón baja 23](#_Toc373003607)

[Gestión de clínica -> Planes 23](#_Toc373003608)

[Agenda -> Registrar 23](#_Toc373003609)

[Gestión de afiliados -> Compra de bonos 24](#_Toc373003610)

[Gestión de afiliados -> Pedir turno 26](#_Toc373003611)

[Gestión de afiliados -> Registro de llegada 28](#_Toc373003612)

[Gestión de afiliados -> Registro de resultados 29](#_Toc373003613)

[Gestión de clínica-> Cancelaciones -> Afiliado 30](#_Toc373003614)

[Gestión de clínica-> Cancelaciones -> Profesional 31](#_Toc373003615)

[Archivo -> Estadísticas 34](#_Toc373003616)

[Conclusiones 35](#_Toc373003617)

# Introducción

## Acerca del modo de trabajo

Para organizarnos dividimos las tareas en tres fases: la definición del modelo de datos, la migración y la aplicación.

En la definición del modelo de datos participamos todos los integrantes hasta definir el DER completo. A continuación nos dividimos en dos equipos, uno se encargó de la migración y el otro de la aplicación.

Para poder trabajar en equipo creamos un repositorio de código en github donde guardamos tanto el código fuente de la aplicación como los scripts.

## Acerca de la aplicación

Creamos una arquitectura que separa el acceso a datos de la presentación de ellos en los formularios. Los detalles están en la sección de arquitectura de la aplicación.

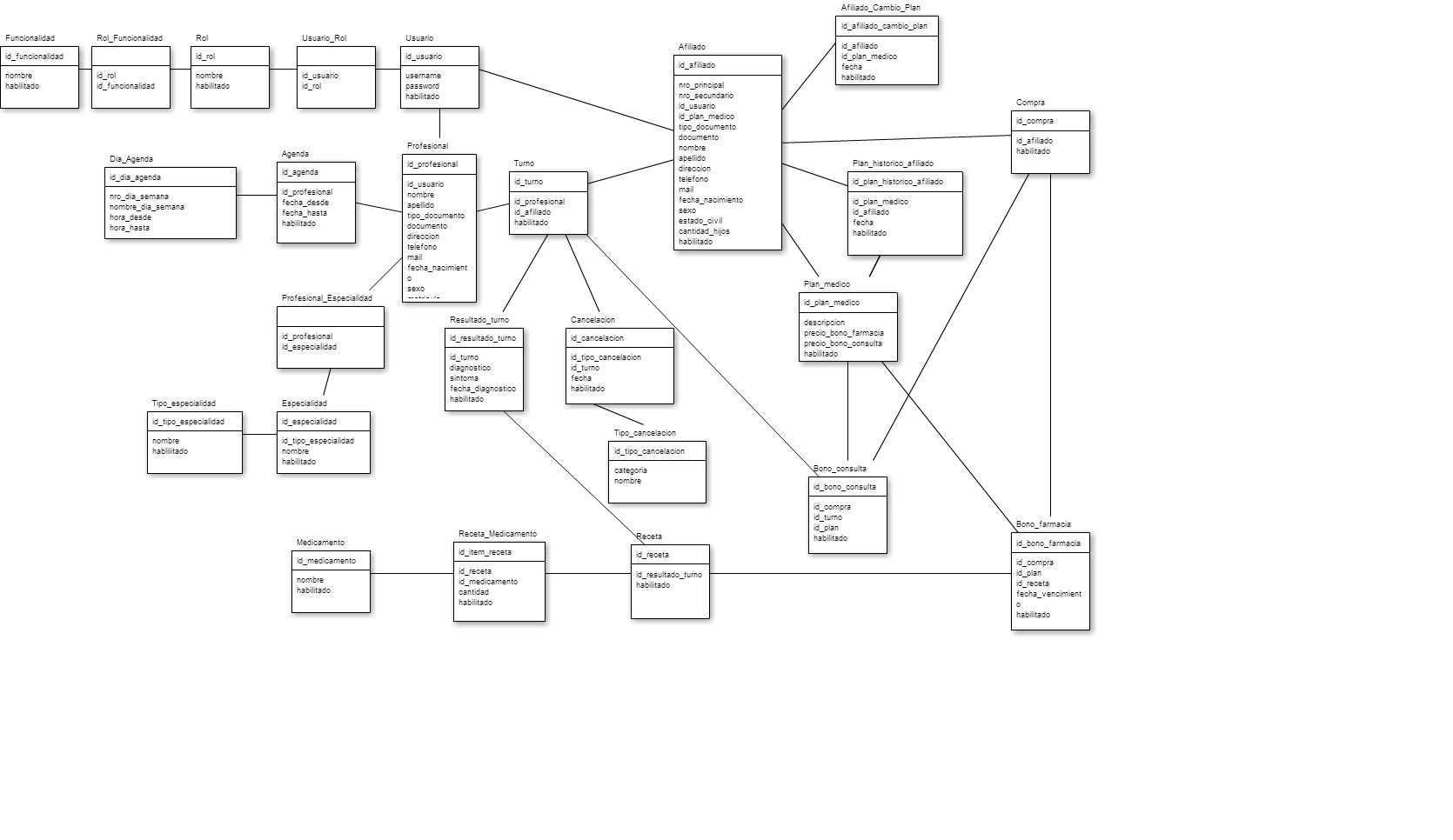
Además para ayudar con el desarrollo y las pruebas hicimos que por cada ejecución se cree un archivo de log registrando todas las invocaciones que nuestro componente de acceso a datos realiza contra la instancia de SQL Server.

Gráficamente definimos trabajar con menúes e iconos para que sea más entendible. Además tenemos un único contenedor de formularios para evitar las múltiples pantallas solapadas una arriba de otra. Las pocas pantallas adicionales que se muestran son modales.

# Decisiones tomadas al realizar la migración

* Por defecto todos los valores que existan en la tabla maestra tienen sexo indefinido, pudiéndose modificar su valor desde la pantalla de modificación de afiliado o profesional. En la aplicación se usa el siguiente esquema:
  + 0 – Indefinido
  + 1 – Mujer
  + 2 – Hombre
* Grupo familiar: Se consideran que son del mismo grupo si tienen la misma dirección.
* Encriptación de password: El hash del password se crea desde la aplicación al momento de hacer el inicio de sesión y se compara contra el varbinary que tiene en el campo password de la tabla Usuario.
* Usuarios: El nombre de usuario es el número de documento y su contraseña es ‘afiliado’ y para los profesionales ‘profesional’
* Tipo documento: Todos quedan con DNI. En la aplicación pueden elegirse otros para las altas o modificaciones.
* No hay tabla de grupos de afiliados, en la tabla afiliado hay un número principal, que van de 100 en 100 y los secundarios de 1 en 1
* La fecha de prescripción es la fecha en que el médico crea la receta y esta fecha se guarda en el bono
* Al eliminar un turno de un profesional, se le cancela el turno al afiliado, sin buscarle reemplazo.
* Al cancelar un rango de días, el profesional puede seleccionar hasta 7 días corridos.

# Modelo de datos



Se encuentra adjunto para su mejor visualización.

# Decisiones de desarrollo

### Barras de estado

Para ayudar a las pruebas, hay dos barras de estado en toda la aplicación con la información del connectionstring, la fecha actual y el usuario identificado en la aplicación

### Carga de los passwords

El campo password de la ventana de login se transforma a SHA256 realizando el siguiente código:

public static byte[] GetSHA256Value(string texto)

{

HashAlgorithm algoritmo = new SHA256Managed();

byte[] textoBytes = new ASCIIEncoding().GetBytes(texto);

byte[] hashValue = algoritmo.ComputeHash(textoBytes);

return hashValue;

}

Para cargar inicialmente en la base de datos los hash passwords de los afiliados y profesionales se creó un stored procedure que recibe ese hash y lo inserta en una tabla auxiliar. De esa tabla copiamos los valores para usarlos en el script inicial.

### Sesión

Al iniciar la aplicación se instancia un objeto IContexto que está vivo durante toda la ejecución de la aplicación. Al realizar un login exitoso, se cargan los valores del usuario y en caso de tener algún profesional o afiliado asociado se lo carga aquí también.

public interface IContexto

{

string LogPath { get; }

ILog Logger { get; }

DateTime FechaActual {get;}

Usuario UsuarioActual { get; }

Rol RolActual { get; }

Afiliado AfiliadoDelUsuario { get; }

Profesional ProfesionalDelUsuario { get; }

void RegistrarUsuario(Usuario usuario);

void RegistrarRol(Rol rol);

void RegistrarAfiliado(Afiliado afiliado);

void RegistrarProfesional(Profesional profesional);

void DesregistrarUsuario();

void DesregistrarRol();

bool SesionIniciada { get;}

}

# Arquitectura de la aplicación .NET

## Diagrama de componentes



Para trabajar de forma desacoplada entre la presentación y los datos almacenados, la solución se compone de varios proyectos:

**ClinicaFRBA**: Es la capa de presentación, contiene los winforms que usarán los usuarios finales.

**GestionDomain**: Conoce las reglas del negocio de la clínica. Resuelve las peticiones de ClinicaFRBA.

**GestionDAL**: En esta capa se transforman las entidades de negocio en objetos compatibles con el Conector. Al recibir los DataSet del conector, los convierte mediante Builders en entidades que GestionDomain usa.

**GestionConector**: Es la capa más cercana a la base de datos. Se basa en ADO.NET y sabe como ejecutar stored procedures y devolver DataSets.

**GestiónCommon**: Contiene objetos comunes a todas las clases. Podemos dividir sus componentes en:

**Entidades**: Representan los objetos del negocio.

**Helpers**: Realizan tareas comunes. Por ejemplo FechaHelper se encarga del manejo de las fechas en el sistema.

## Diagrama de clases de acceso a datos



Para interactuar contra la base de datos creamos una capa llamada GestionConector que tiene la lógica para ejecutar stored procedures o consultas SQL, devolviendo objetos del tipo DataSet.

**Este conector usa el connectionstring del archivo de configuración de la aplicación.**

La capa DAL devuelve objetos que representan a las entidades, convirtiendo las filas del tipo DataRow mediante un objeto builder, que conoce como mapear cada campo contra cada propiedad.

Al conector lo usa la capa DAL, que es la encargada de transformar los objetos de las entidades en parámetros de los stored procedures y usa el conector para comunicarse con la base de datos.

A la capa DAL la conoce la capa de Domain que se encarga de realizar las validaciones de negocio necesarias y gestionar los errores que hayan surgido desde el modelo de datos para que sean legibles para el usuario final.

Para ello cada respuesta de la capa Domain es del tipo IResultado, que enmascara los errores y las devoluciones.

La capa de presentación conoce los datos a partir de invocaciones a la capa Domain.

## Diagrama de clases de Capa de presentación



Todos los formularios de ABM heredan de FormularioBase que conoce una lista de validaciones y sabe validar los datos antes de realizar alguna acción.

A su vez, los formularios de alta, baja y modificación heredan de formularios base que ya tienen los botones predefinidos para cada tipo de operación.

Creamos clases Validadores que reciben controles de winforms y se encargan de validar entradas de datos.

# Configuración de la aplicación

La aplicación cuenta con un archivo de configuración donde se pueden cambiar sin tener que compilar la aplicación los siguientes parámetros:

* DateTimeNow: Donde se le indica a la aplicación que fecha usar como actual.
* DateFormatString: Indica como formatear las fechas en el sistema.
* DateTimeFormatString: Indica como formatear las fechas y horas en el sistema
* TimeFormatString: Indicia como formatear las horas en el sistema.
* Log\_path: Indica en que ubicación guardar el archivo de log asociado a la ejecución del programa.
* BaseDeDatos\_Esquema: Se guarda el valor usado para generar los nombres de los stored procedures automáticamente desde las clases DAL.
* SP\_Obtener, SP\_Obtener\_Todos, SP\_Borrar, SP\_Actualizar, SP\_Insertar, SP\_Filtrar: Indican cómo se llamaran genéricamente los stored procedures para usarlos cuando se generan los nombres automáticamente desde las clases DAL
* Connection\_string: Se guarda el string que contiene como acceder a la instancia de SQL Server asociada al programa.

Este archivo de configuración se encuentra en la carpeta: TP2C2013 k3051 TOP\_4 14\src\Clinica Frba\App.config

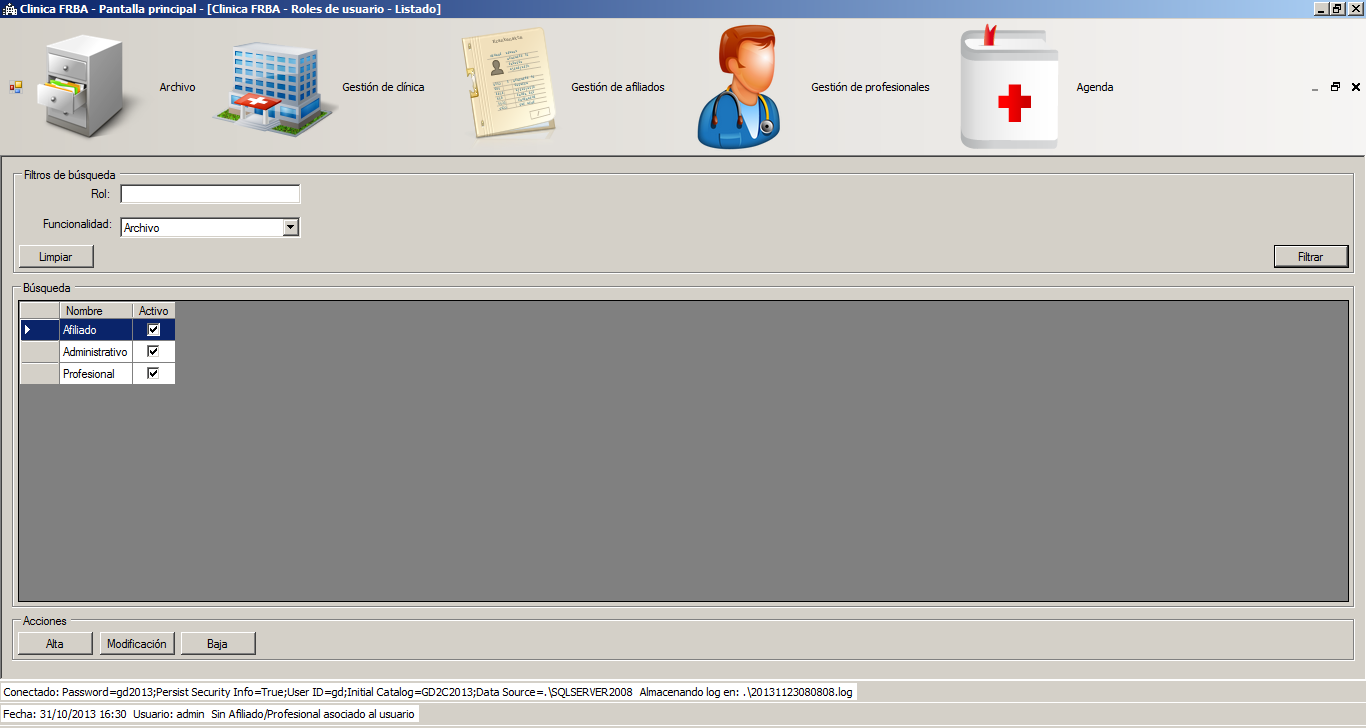
# Uso de la aplicación

En esta sección se describe cómo usar la aplicación

Abm de rol

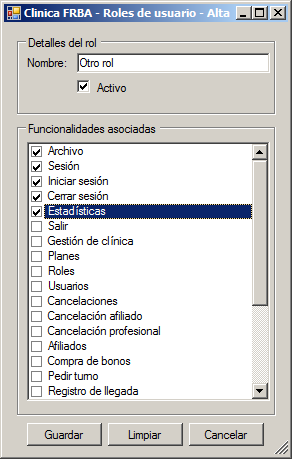
### Gestión de clínica-> Roles

Desde esta pantalla se pueden listar los roles existentes:



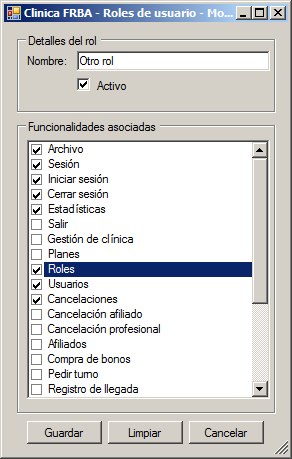
### Gestión de clínica-> Roles -> Botón Alta

Desde esta pantalla se crean nuevos roles.



### Gestión de clínica-> Roles -> Botón Modificación

Desde esta pantalla se puede modificar un rol existente:



### Gestión de clínica-> Roles -> Botón Baja

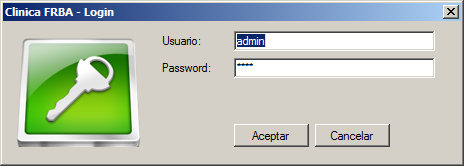
La aplicación consulta si se quiere dar de baja el rol. En caso afirmativo se realiza la baja.

Login y seguridad:

Al iniciar la aplicación aparece la ventana de inicio de sesión y por defecto para las pruebas queda cargado el usuario admin y su password ‘w23c’ cuyo perfil es administrativo.

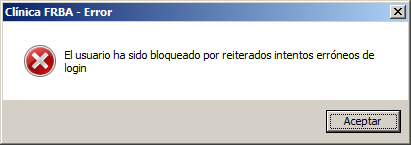
### Archivo -> Sesión -> Iniciar Sesión

La pantalla permite el ingreso del usuario y contraseña. Al hacer clic en el botón aceptar se convierte el string del password a un varbinary usando el algoritmo de encriptación SHA256. Se consulta en la base de datos si los datos ingresados corresponden a un usuario del sistema.

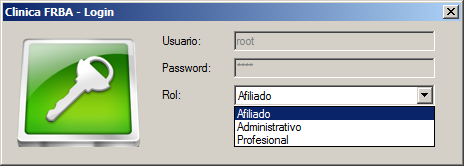


En caso afirmativo, si el usuario tiene un solo rol asociado, se procede a cargar la pantalla principal.

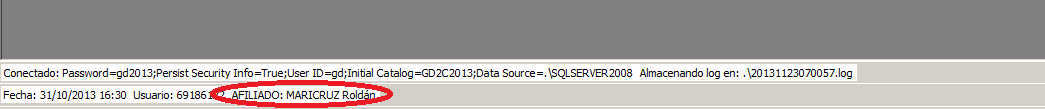
En caso de que el password sea incorrecto se incrementa en la tabla usuario un contador de ingresos fallidos. Al tercer reintento el usuario queda bloqueado:



Si tiene más de un rol asociado, aparece en la ventana un combo para que seleccione el rol con el que quiere ingresar:

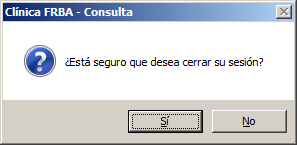


Al cargar la pantalla principal, si el rol tiene un afiliado asociado o un profesional asociado, se muestran en la barra de estado sus datos:

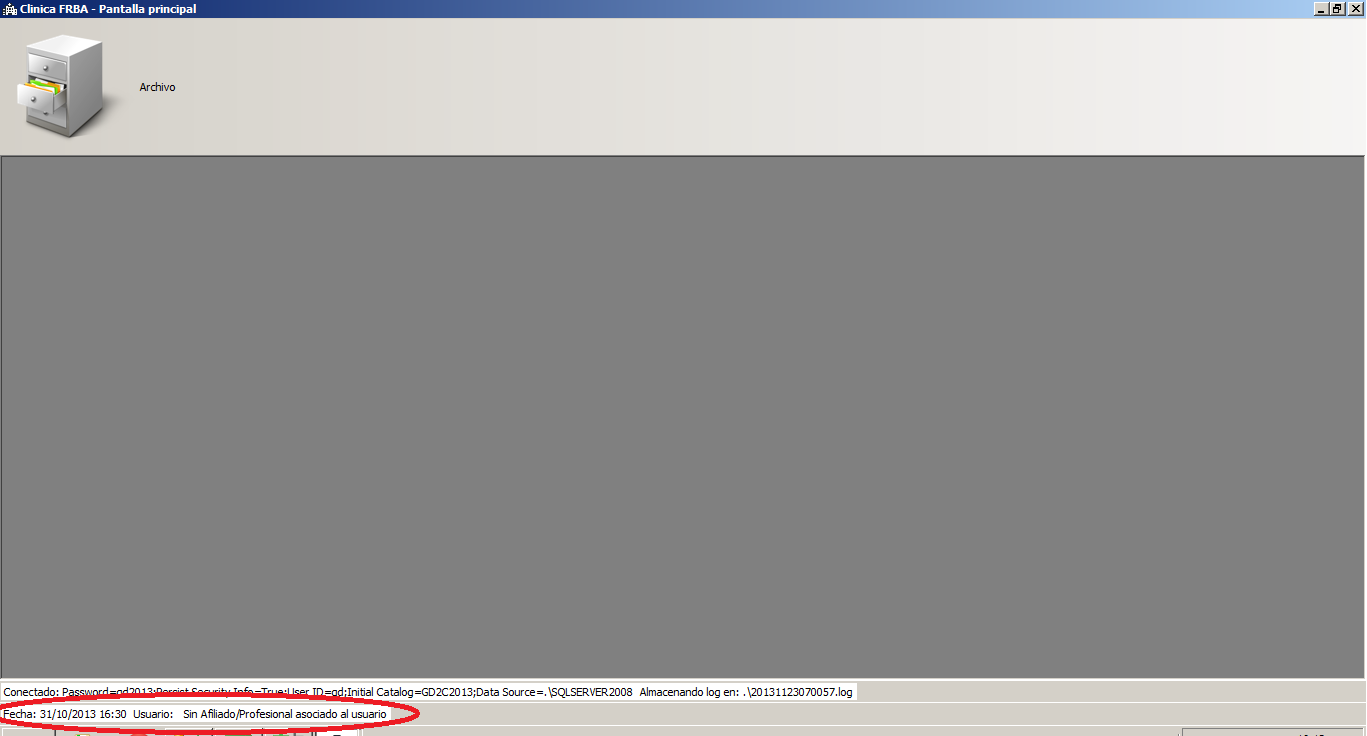


### Archivo -> Sesión -> Cerrar Sesión

Al querer cerrar la sesión se pregunta si el usuario está seguro.



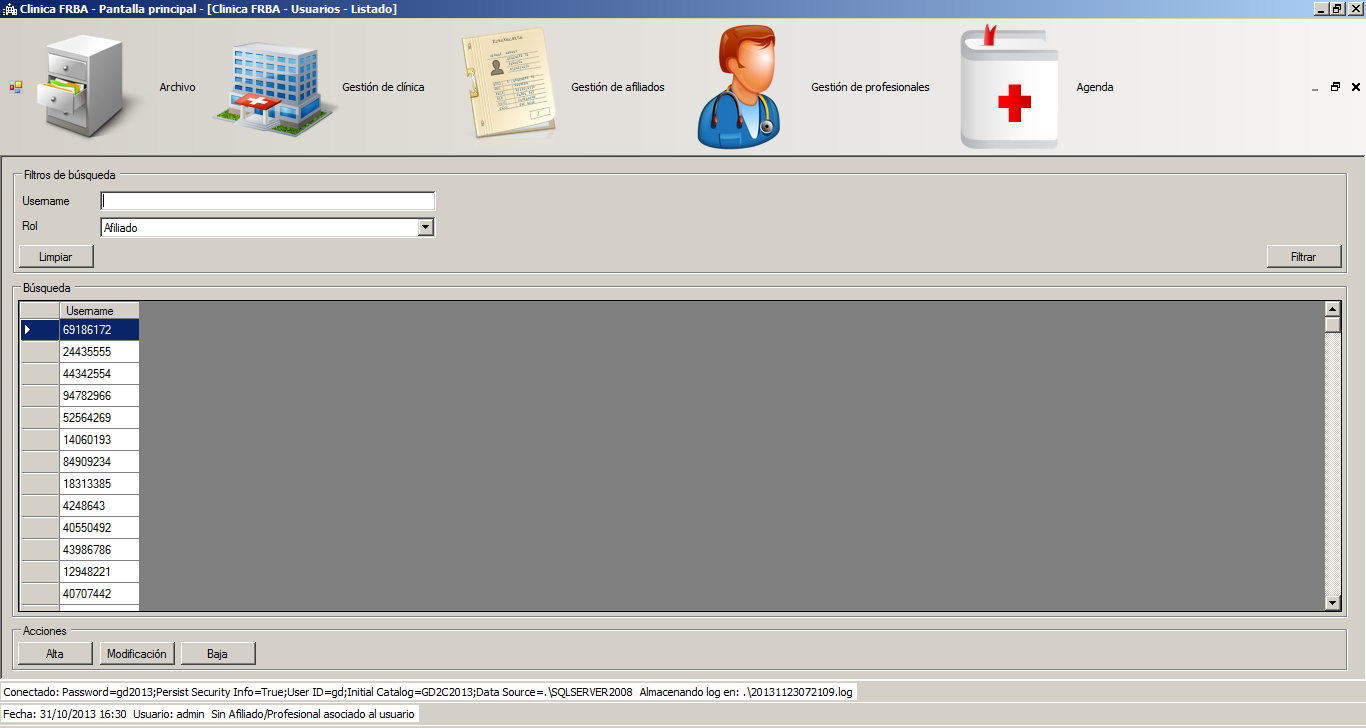
En caso afirmativo, la aplicación queda sin usuario asociado y la barra de menú no muestra las opciones de la aplicación.



Registro de usuario:

Al crear un afiliado o profesional se le asigna un usuario automáticamente tomando el número de documento como nombre de usuario.

Se consultan desde Gestión de Clínica -> Usuarios

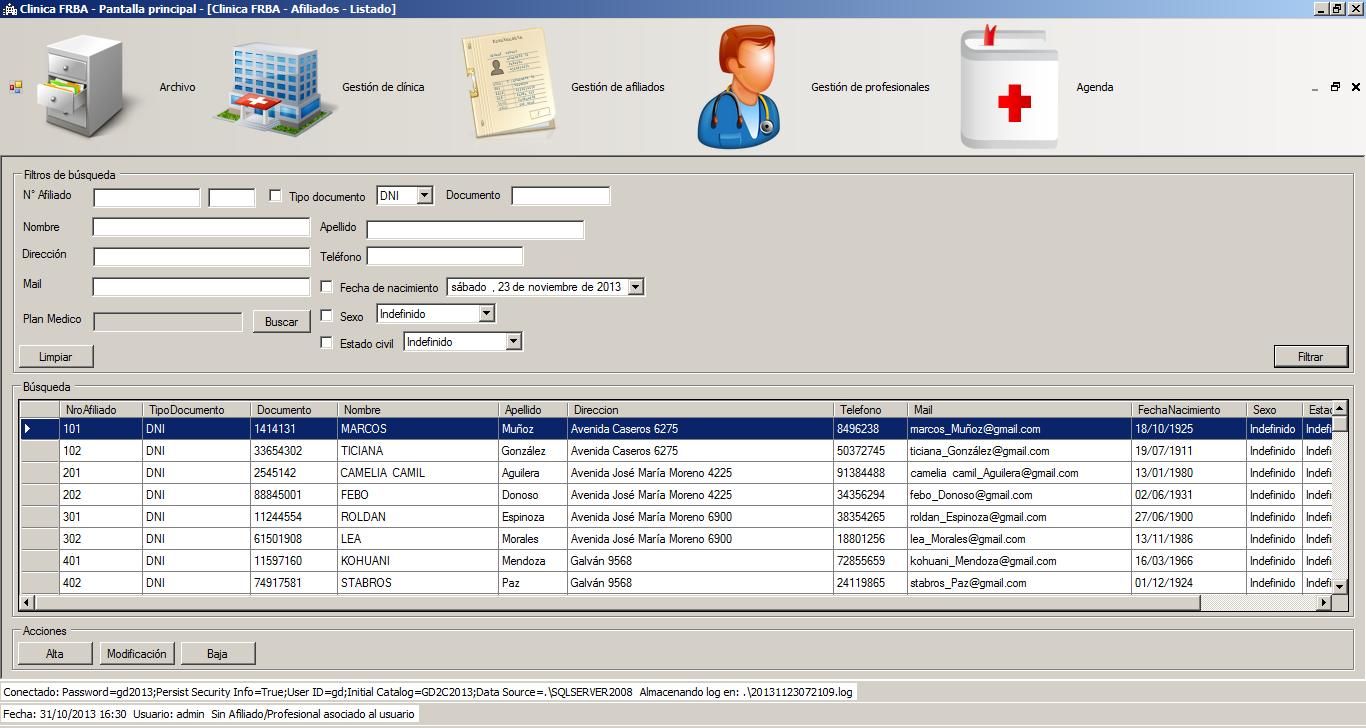


La modificación y baja no se implementa porque está fuera del alcance del trabajo práctico.

ABM de afiliados

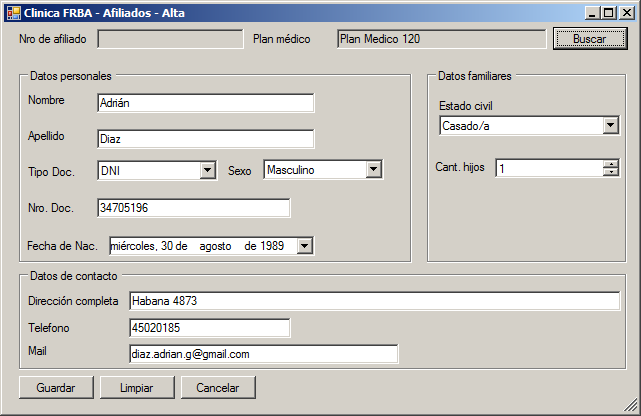
### Gestión de afiliados -> Afiliados

Desde esta pantalla se pueden consultar todos los afiliados registrados en el sistema. Se pueden aplicar filtros para ajustar la búsqueda.

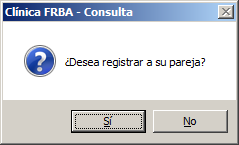


### Gestión de afiliados -> Afiliados -> Botón Alta

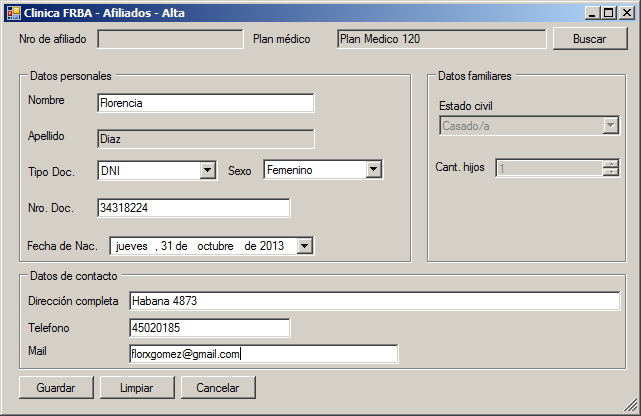
Permite dar de alta un afiliado teniendo en cuenta todas las validaciones de los campos.



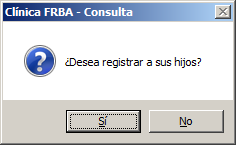
Si el afiliado tiene como estado civil uno con pareja el sistema consulta si quiere registrar al conyugue:



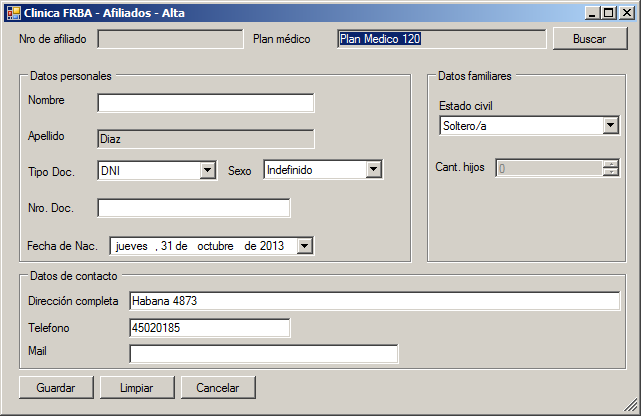
Si acepta, accede a una nueva ventana de alta de afiliado, donde por defecto ya tiene el mismo apellido, dirección, teléfono, plan médico y estado civil junto a la cantidad de hijos. Sólo el apellido, el estado civil y la cantidad de hijos no se pueden cambiar.



Luego del alta del conyugué ( o no, si es que no quiso registrar a su pareja) el sistema consulta si es que tiene hijos si los desea registrar:

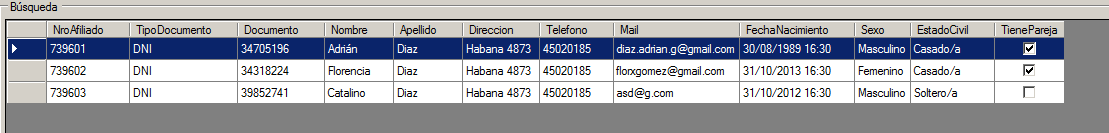


En caso afirmativo, se vuelve a mostrar una pantalla de alta para que cargue sus datos.



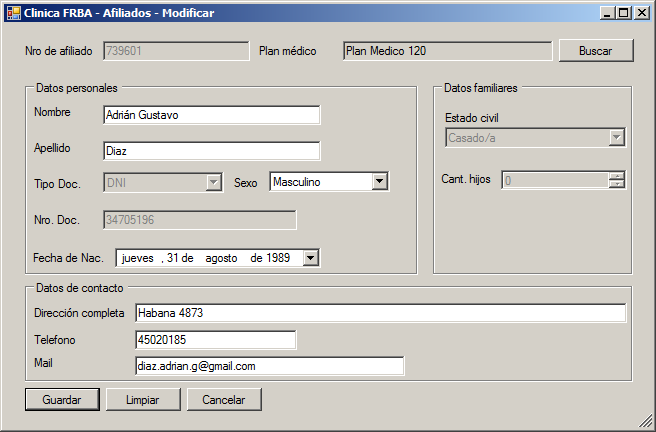
En este caso el hijo puede cambiar su estado civil.

Al finalizar el alta el sistema vuelve al listado y carga los usuarios nuevamente. Si filtramos por el afiliado creado obtenemos:



### Gestión de afiliados -> Afiliados -> Botón Modificar

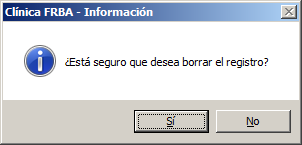
El sistema consulta si está seguro realizar la modificación y luego se accede a la pantalla de modificación de afiliados:



Al salir, se vuelve a la pantalla de afiliados.

### Gestión de afiliados -> Afiliados -> Botón Baja

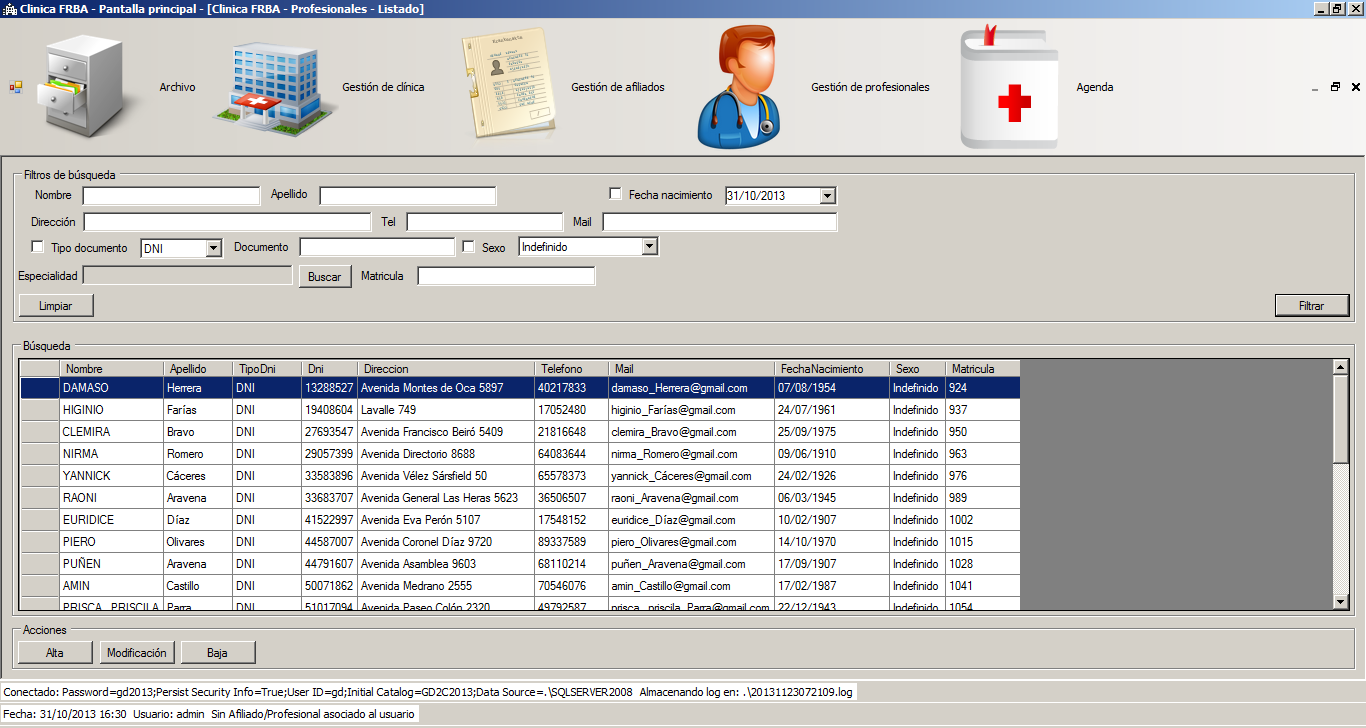
El sistema consulta si se quiere dar de baja al afiliado, en caso de confirmar, se realiza la baja lógica.



ABM de profesional

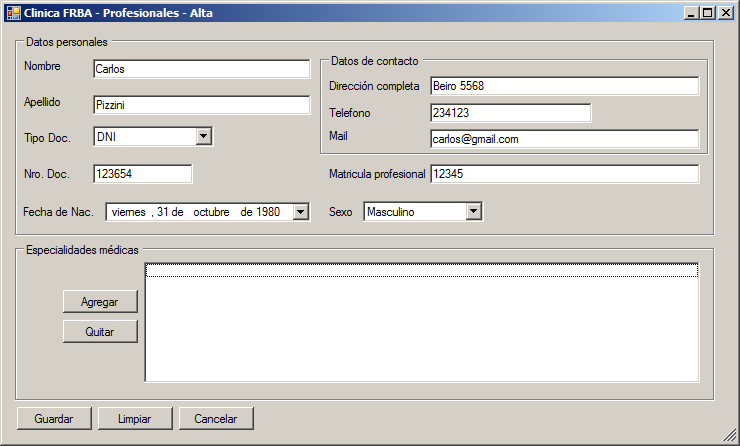
### Gestión de profesionales -> Profesionales

Desde esta pantalla se listan todos los profesionales de la clínica.

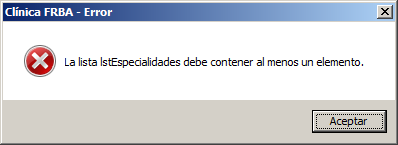


### Gestión de profesionales -> Profesionales -> Botón Alta

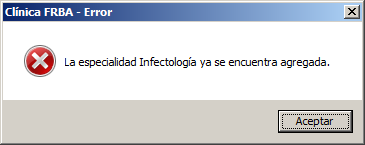
Permite dar de alta un nuevo profesional teniendo en cuenta las validaciones necesarias.



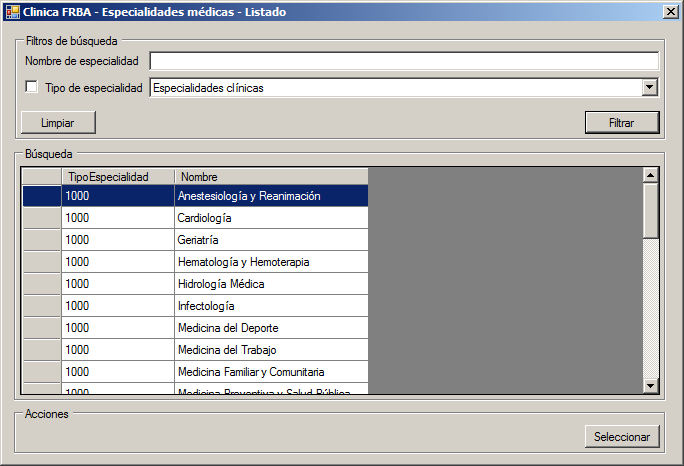
Si no tiene especialidades médicas no permite darlo de alta:



Si intenta agregar una especialidad que ya existe se muestra un mensaje:



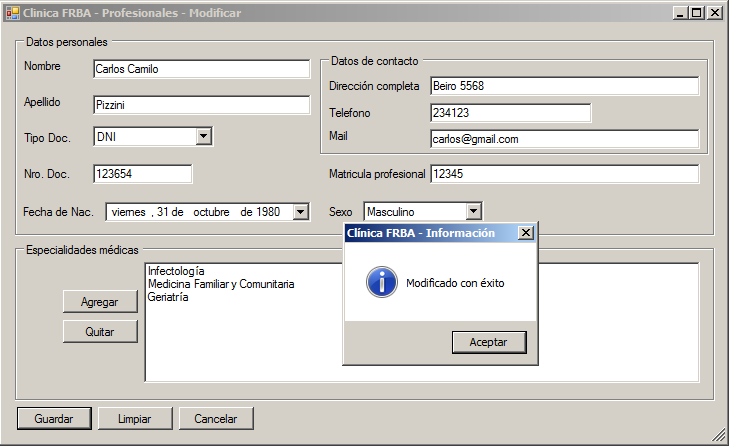
Para agregar una especialidad se usa la pantalla de listado de especialidades médicas en “modo selección”, donde figura un botón seleccionar y se ocultan los botones de alta, modificar y baja.



Una vez dado de alta se vuelve al listado de profesionales.

### Gestión de profesionales -> Profesionales -> Botón Modificar

Seleccionando un profesional se puede acceder a la pantalla de modificación y cambiar sus datos:



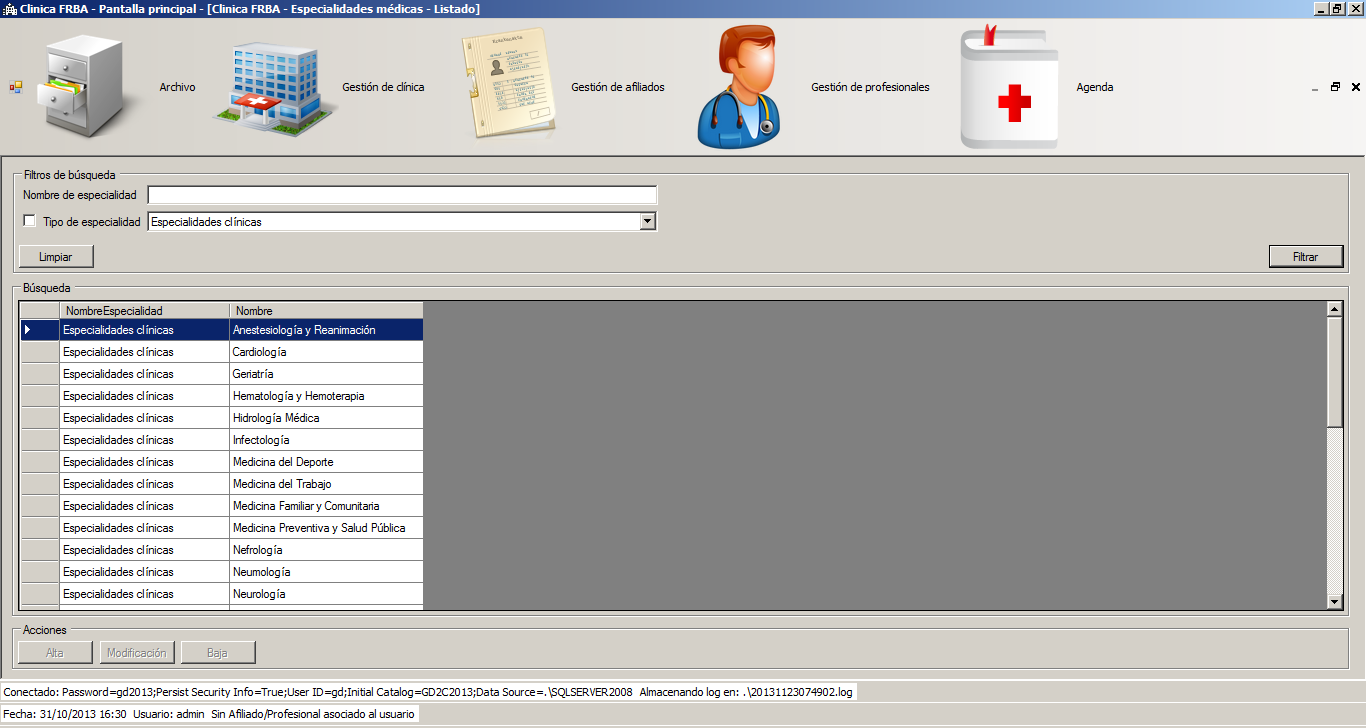
### Gestión de profesionales -> Profesionales -> Botón Baja

De misma forma que la baja de afiliados, se consulta si quiere eliminar al profesional

ABM de Especialidades médicas

### Gestión de profesionales -> Especialidades médicas

Desde esta pantalla se pueden listar las especialidades médicas de la clínica y filtrar por su tipo si se marca el checkbox de “Tipo de especialidad” o por su nombre



### Gestión de profesionales -> Especialidades médicas -> Botón alta

No se implementa porque está fuera del alcance del trabajo práctico

### Gestión de profesionales -> Especialidades médicas -> Botón modificar

No se implementa porque está fuera del alcance del trabajo práctico

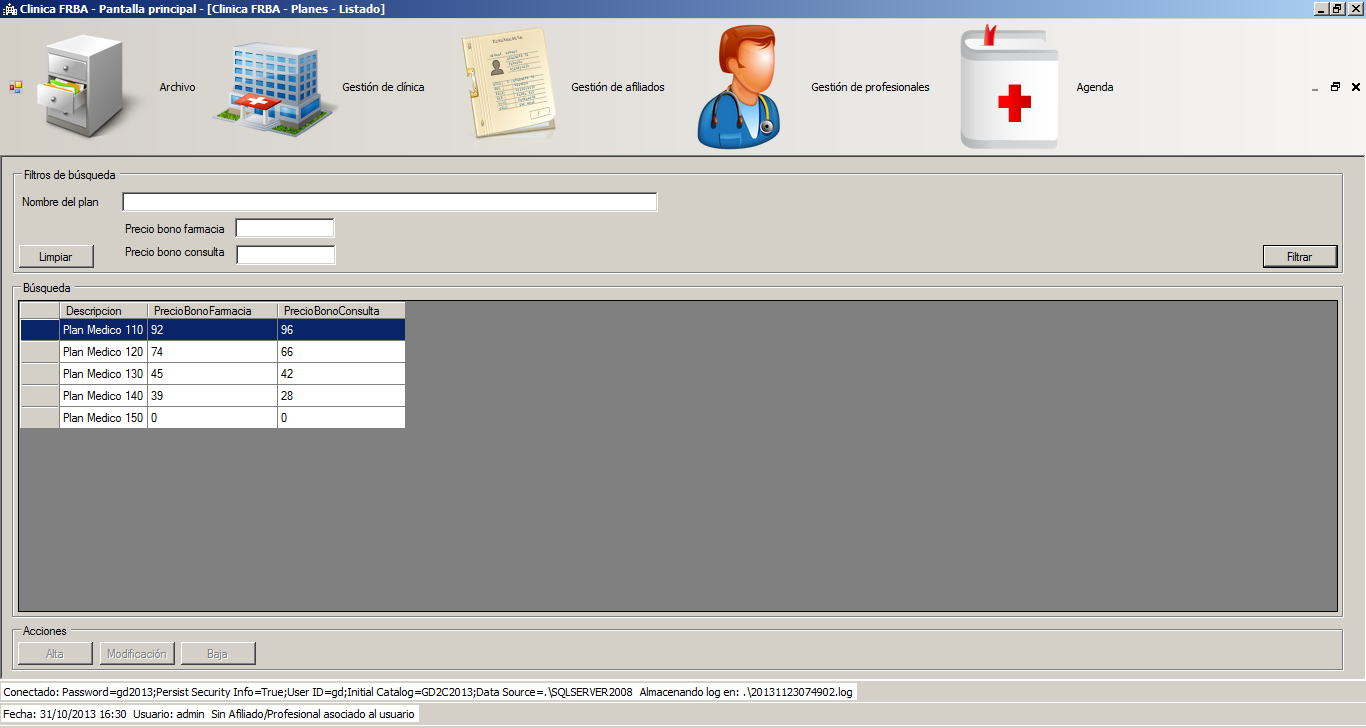
### Gestión de profesionales -> Especialidades médicas -> Botón baja

No se implementa porque está fuera del alcance del trabajo práctico

ABM de planes

### Gestión de clínica -> Planes

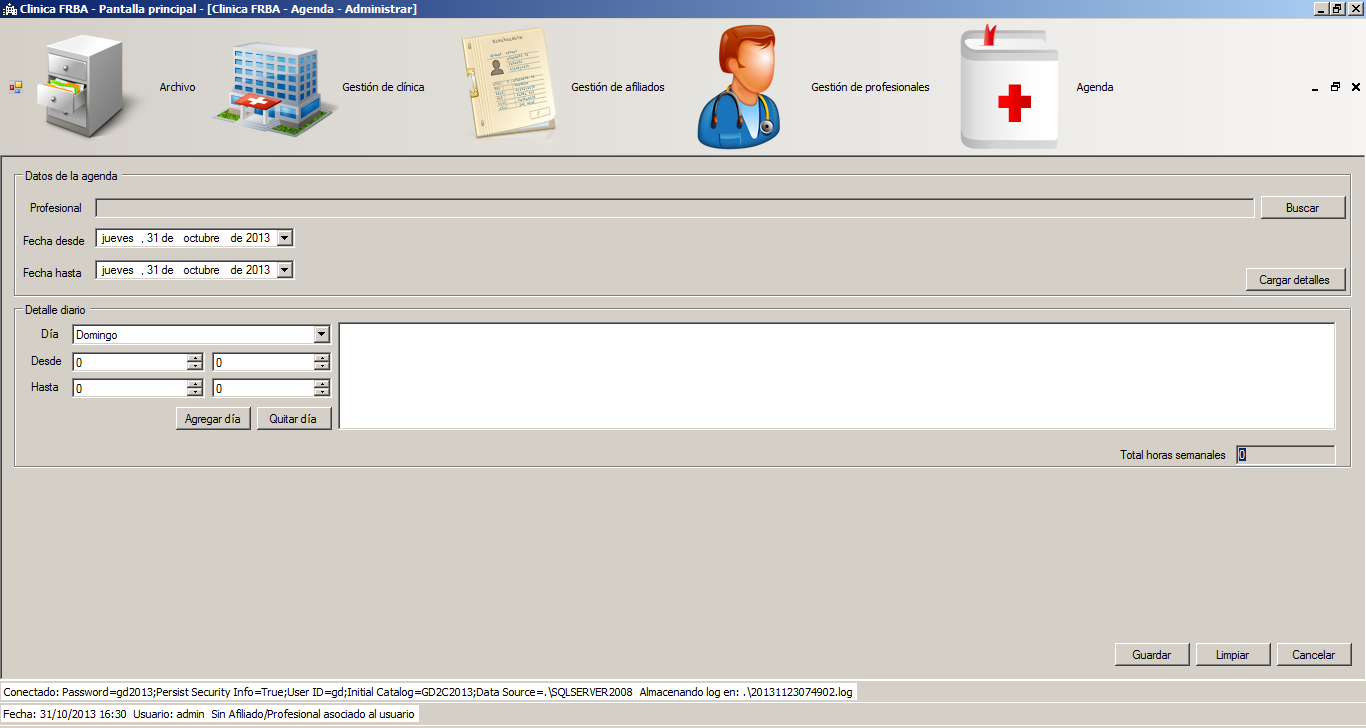
Desde esta pantalla de pueden listar los planes de la clínica:



Registrar Agenda del médico

### Agenda -> Registrar

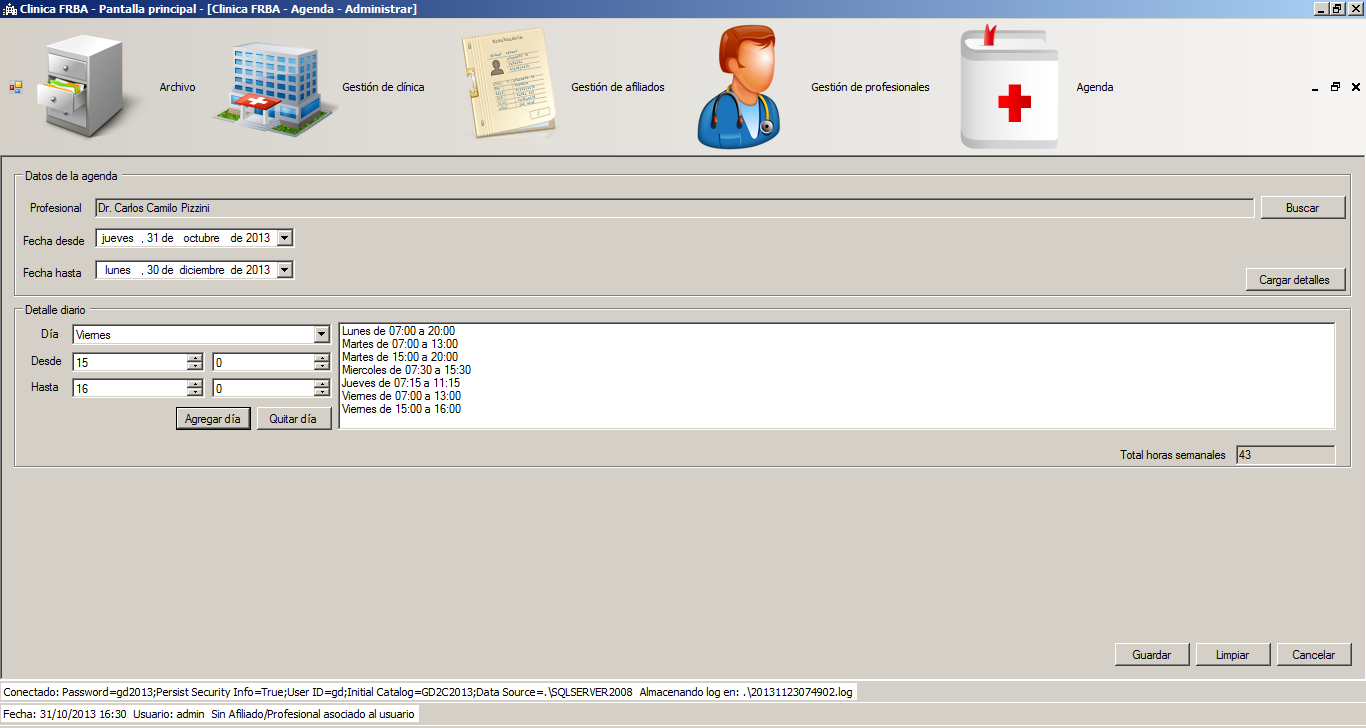
Primero se debe seleccionar el profesional a cargarle su agenda. Si el usuario tiene asociado un profesional, este campo ya estará cargado.



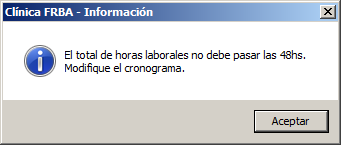
Luego se debe elegir el rango de fechas a asignarle horarios. Hay un límite de 120 días de rango.

Una vez definido el rango se procede a cargarle los días uno por uno. A cada día de la semana le corresponde una hora de inicio límite y una hora de fin límite en los que la clínica está abierta. Al seleccionar un día, estos límites se cargan en los controles de “Desde” y “Hasta”.

Para cada día se da la posibilidad de usar minutos para cargar las horas y usar horarios partidos por día, es decir que en un día puedo cargar de 08:00 a 11:00 y de 15:00 a 17:00.



Debajo de la lista de días se visualiza la suma de horas total por semana que no debe superar las 48, sino aparece este cartel:

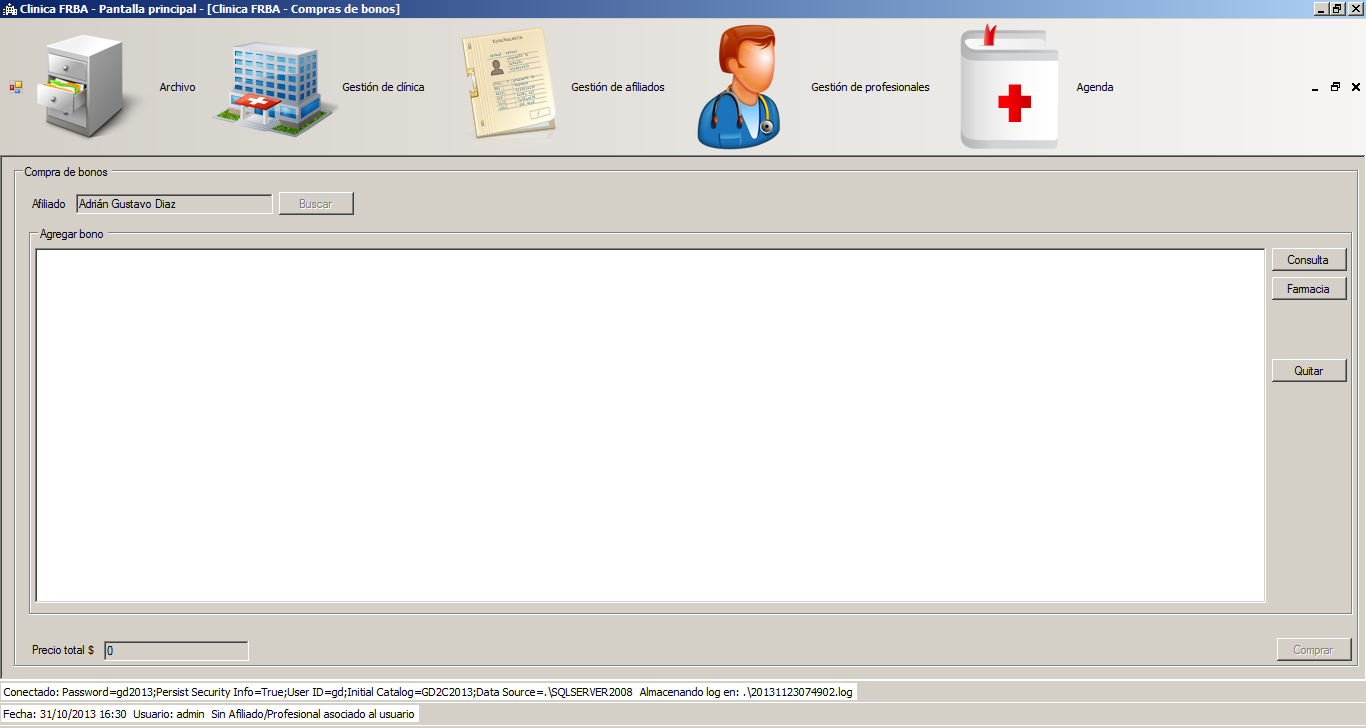


Para los días que no son laborables, se pueden dar de baja usando el módulo de cancelaciones para profesionales.

Compra de bonos

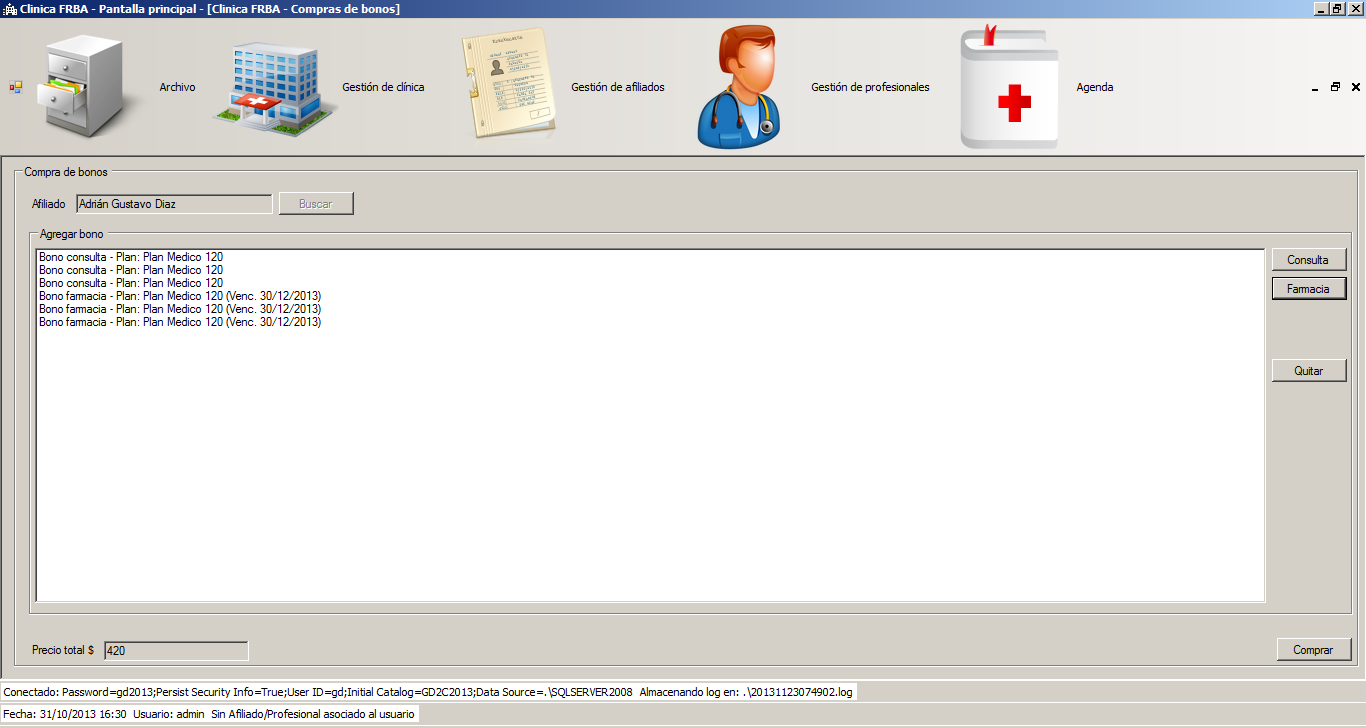
### Gestión de afiliados -> Compra de bonos

Para la compra de bonos primero hay que seleccionar al afiliado que realiza la compra si es que el rol es administrativo.



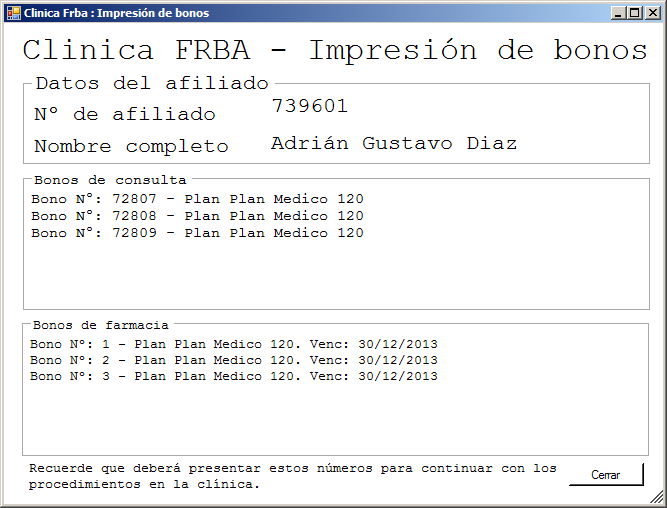
En caso de que el afiliado haya iniciado sesión, ya estará su nombre cargado en la pantalla.

Mediante los botones “Consulta” y “Farmacia” agrega los bonos a la lista de compra:



Debajo de la lista se muestra el precio total de los bonos en base a su plan médico asociado.

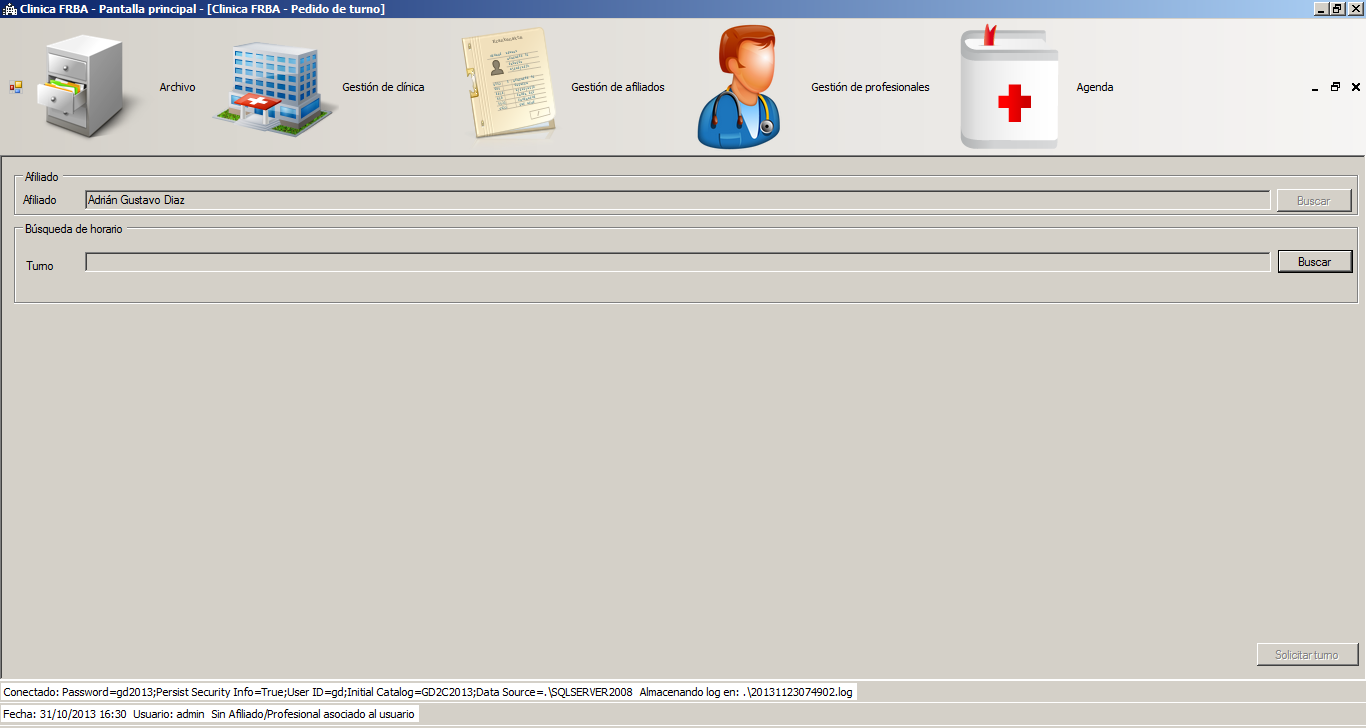
Cuando realiza la compra mediante el botón “Comprar” se imprime por pantalla el detalle de su compra:



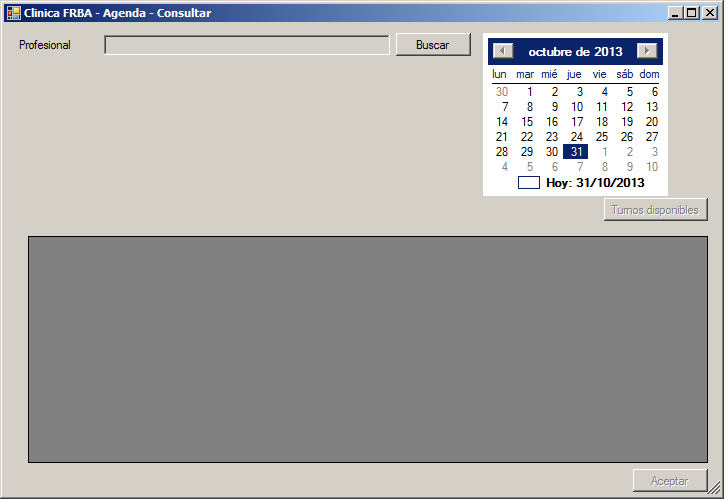
Pedir Turno

### Gestión de afiliados -> Pedir turno

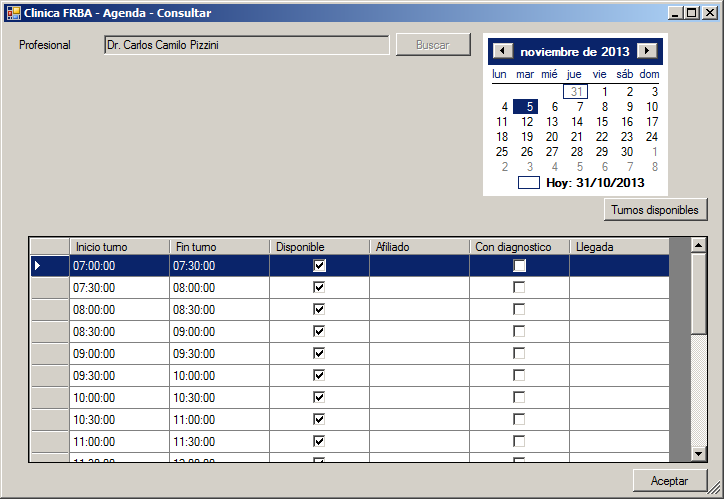
Desde esta pantalla se busca al afiliado que solicita el turno. En caso de que el afiliado sea el que inicio sesión, su nombre ya estará cargado.



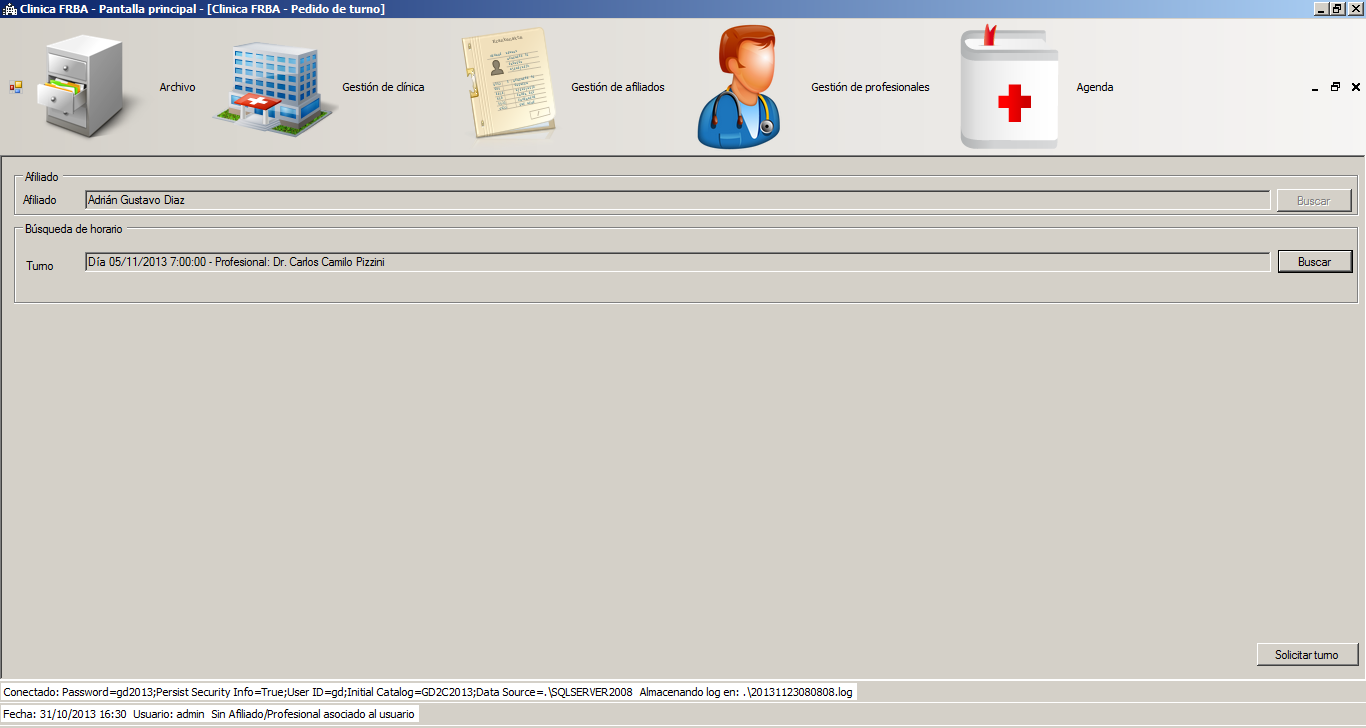
Cuando el afiliado ya está cargado se busca el turno:



Primero debe cargar al profesional y luego se le habilitaran las fechas posibles en el calendario. Al hacer clic en Turnos disponibles se muestran las posibilidades para el día seleccionado



Se selecciona una y se acepta para volver a la pantalla anterior:

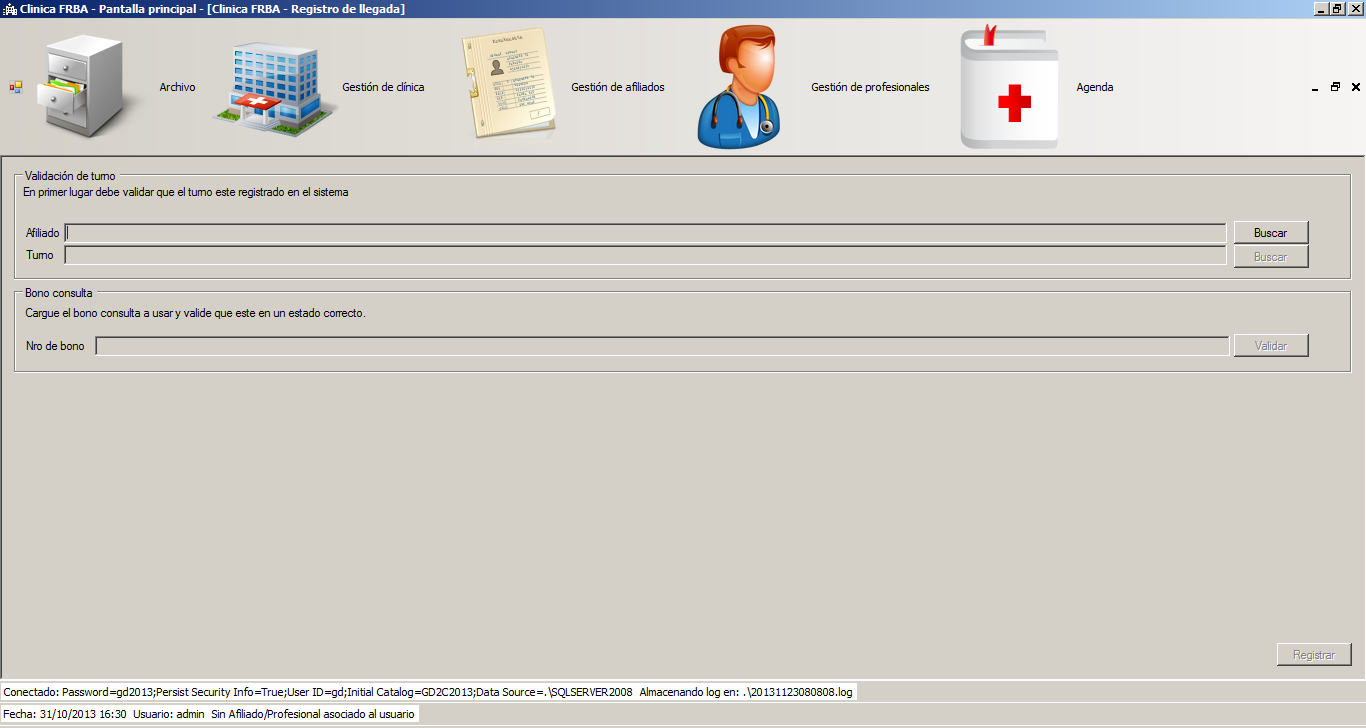


Al solicitar el turno se muestra un mensaje al usuario para confirmarle la decisión.

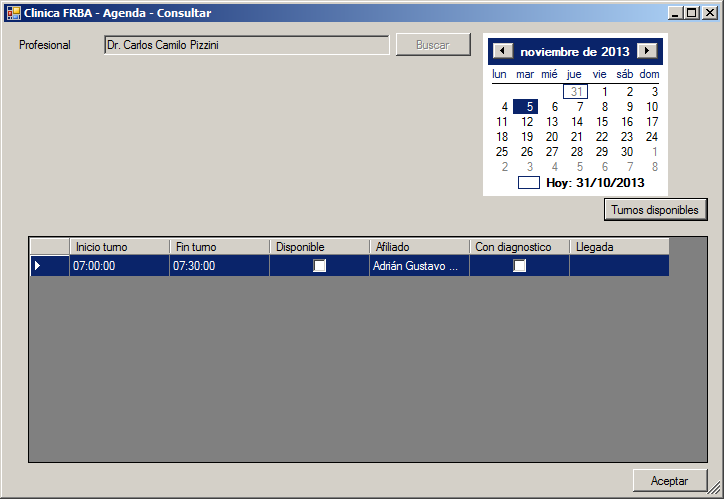
Registro de llegada para atención médica

### Gestión de afiliados -> Registro de llegada

Desde esta pantalla el administrativo carga el afiliado que llega a la recepción:



Luego con el afiliado cargado, busca el turno para verificar que realmente existe:



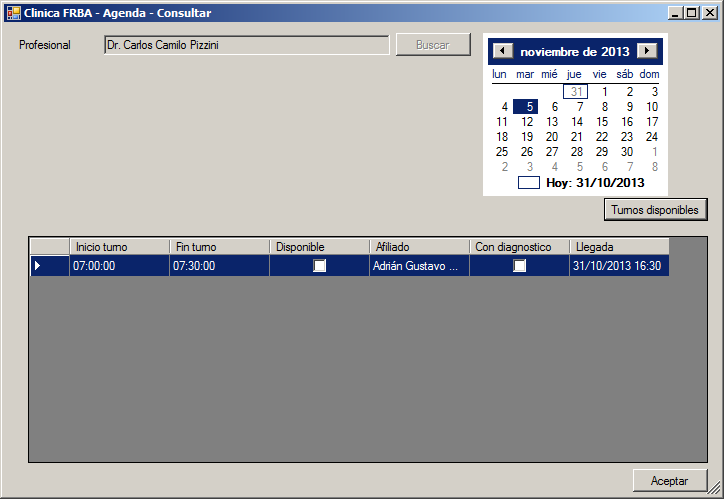
Le solicita un bono de consulta y se lo valida. Una vez validado se hace clic en registrar para finalizar.

Registrar resultado para atención médica

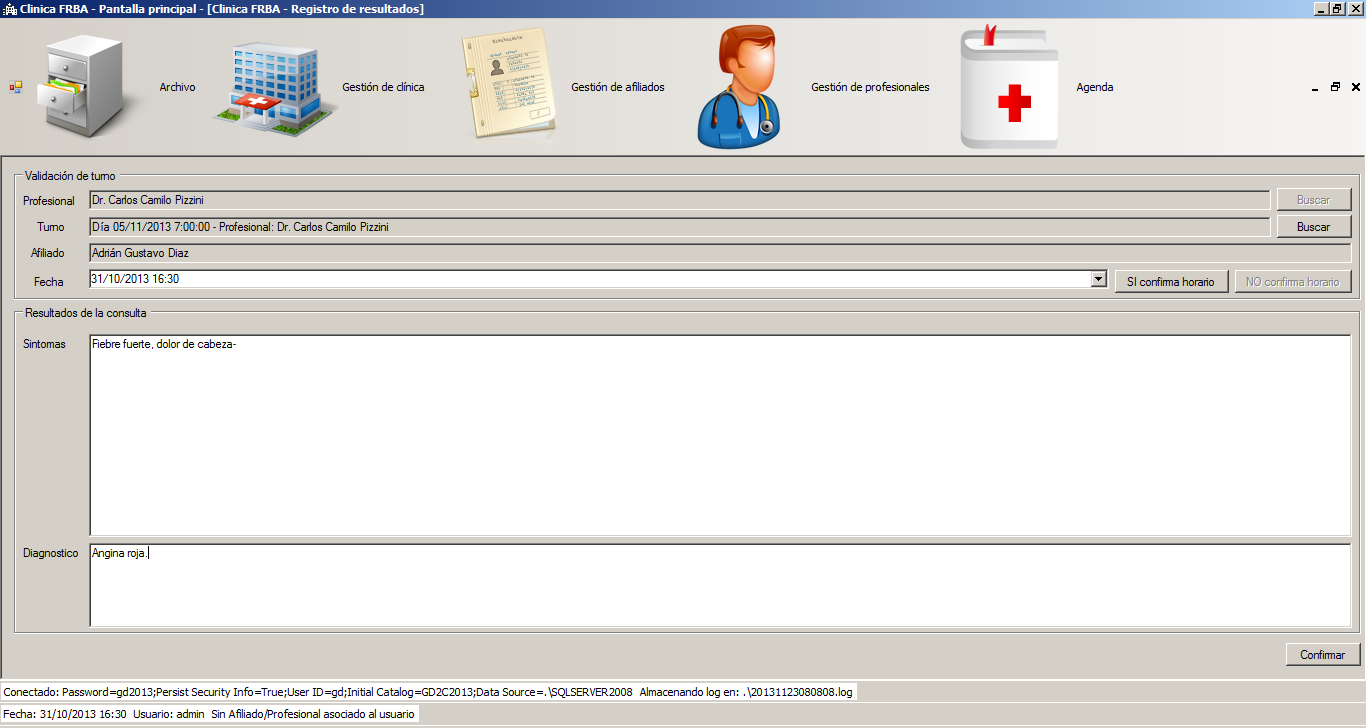
### Gestión de afiliados -> Registro de resultados

Desde esta pantalla se carga el profesional que está realizando el diagnostico. En caso de que el usuario que inicio sesión tenga asignado un profesional, este campo ya estará cargado:

Luego se busca el turno:



Confirma el horario de ingreso al consultorio y carga los datos del diagnóstico:



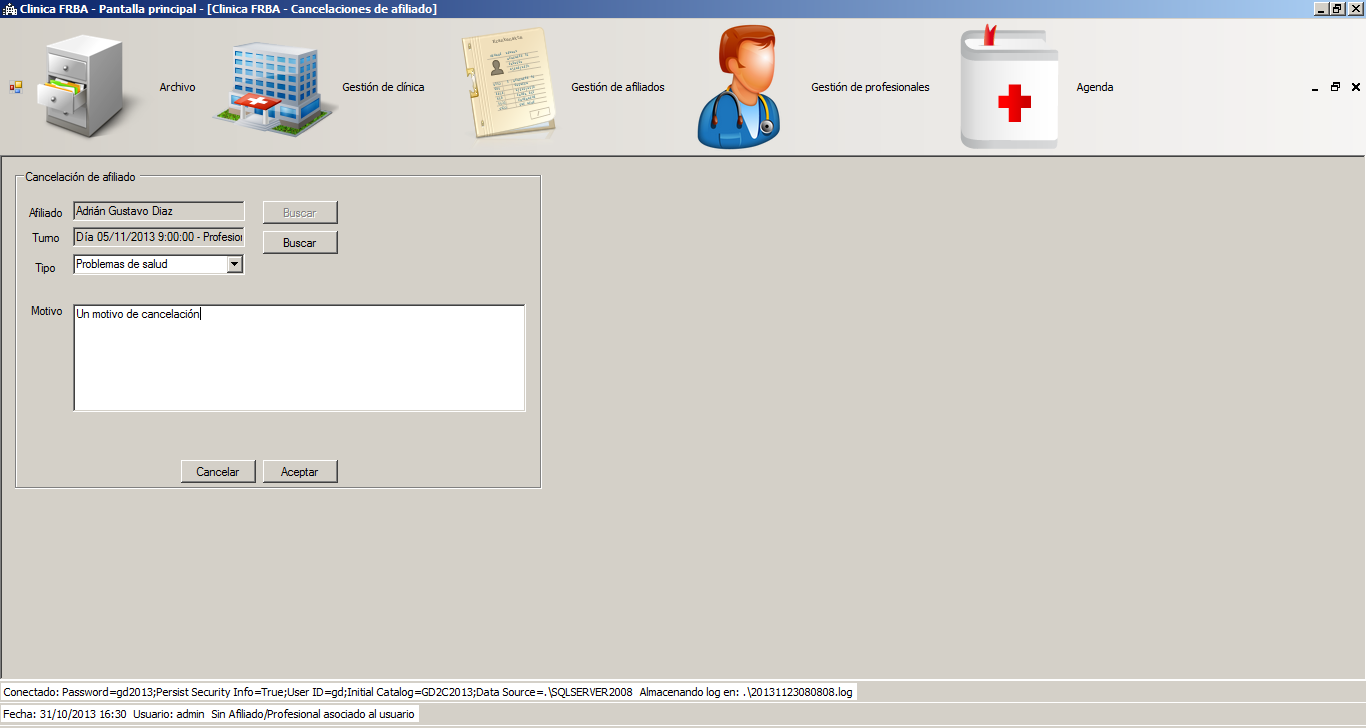
Al confirmar se le consulta si quiere realizar recetas.

Cancelar atención médica

### Gestión de clínica-> Cancelaciones -> Afiliado

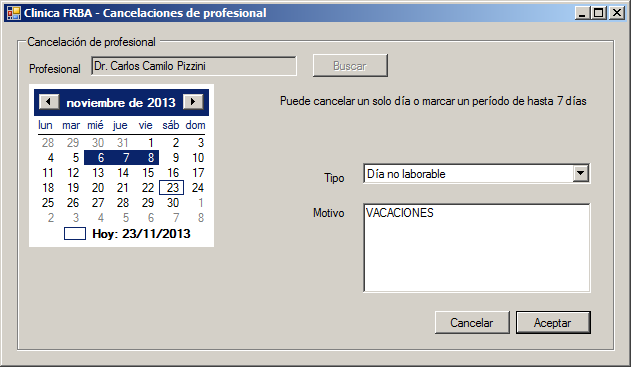
Desde esta pantalla se busca al afiliado que realiza la cancelación. Si el usuario está asociado a un afiliado, este campo ya estará cargado.

Luego deberá buscar un turno a cancelar y agregar el motivo:



### Gestión de clínica-> Cancelaciones -> Profesional

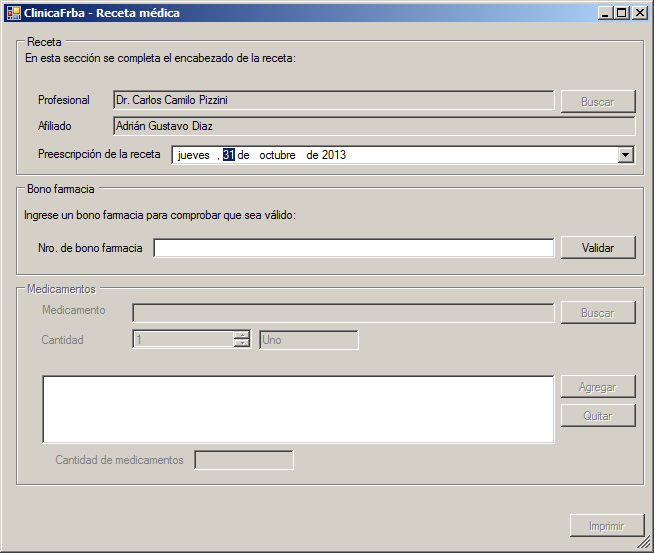
Desde esta pantalla el profesional puede cancelar un día o un rango de días:



Al cancelar los días, se dan de baja los turnos y no se asignan a otros profesionales.

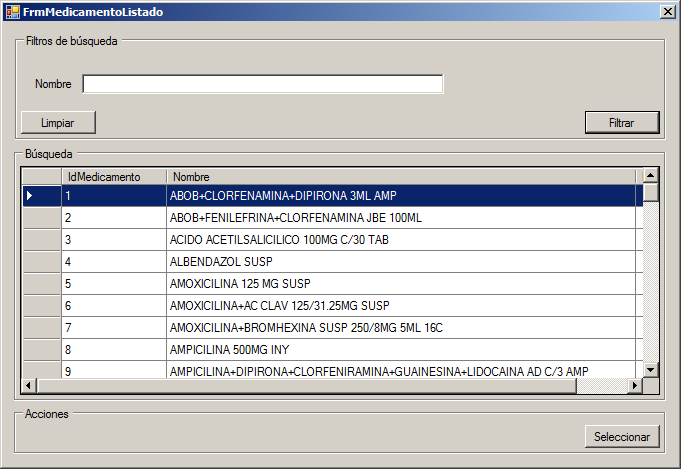
Generar receta

Al registrar resultados de la atención médica se pregunta si quiere registrar recetas.

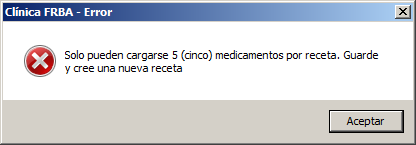


Desde esta pantalla debe cargar un bono farmacia y validarlo para que le permita cargar medicamentos.

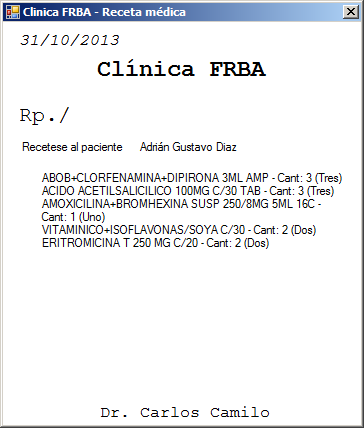
Por cada medicamento se lo busca usando la pantalla de listado de medicamentos:



Al seleccionarlo se le carga la cantidad y se lo agrega a la lista. Si quiere agregar más de cinco medicamentos aparecerá la pantalla:



Cuando están todos los medicamentos cargados se procede a imprimir la receta:



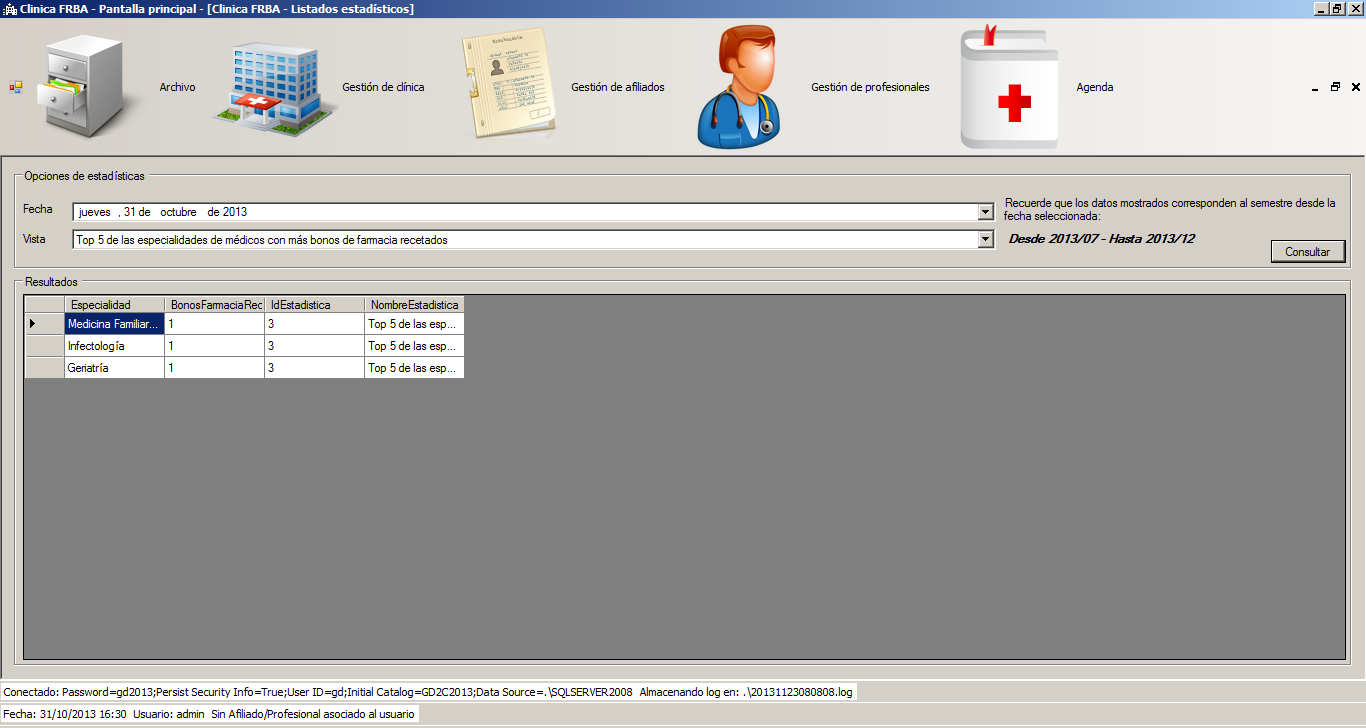
Listados estadísticos

### Archivo -> Estadísticas

Se accede desde el menú archivo y desde la lista de estadísticas se permite elegir cual mostrar:



Cada vez que se elige una fecha, se hace un cálculo para determinar el inicio y fin del semestre en el que se encuentra y esas fechas se usan para invocar al stored procedure correspondiente a la estadística elegida.



## Conclusiones

Al realizar el trabajo nos basamos en dos fundamentos:

* Aplicar los conocimientos adquiridos en materias anteriores como Diseño de sistemas, al realizar una arquitectura separada en capas evitando repetir código, separar responsabilidades y tener abstracción al ocultar el acceso a la base de datos a la capa de presentación por ejemplo.
* Usar los conocimientos dados por la materia, es decir, realizar una aplicación que cumpla con las mejores prácticas a nivel modelo de datos: no hay queries sueltas en el código de la aplicación, siempre se invocan stored procedures. Hay uso de transacciones para asegurar consistencia, constraints para verificar los ingresos de datos provenientes desde la aplicación y agregamos los índices necesarios para agilizar las consultas.

Logramos un buen trabajo en equipo y pudimos cumplir con el desarrollo dentro de las fechas que estimábamos.