ACORDARSE DE PASAR A PDF







NEXTGDD

**Índice:**

* **ARCHIVO DE CONFIGURACION....................................................... 1**
* **CONSIDERACIONES ASUMIDAS....................................................... 2**
* **DER………………………………………………………………………. .3**
* **ACLARACIONES SOBRE EL MODELO*.....................................3***
* **IMAGEN …………………………………………………………..5**
* **MIGRACION……….............................................................................. ….6**
* **APLICACION DESKTOP........................................................................ .7**
* **LOGIN Y SEGURIDAD………………………………………….7**
* **ABM…………………….………………………………………….8**
* **USUARIOS………………………………………………………….8**

**ARCHIVO DE CONFIGURACION**

El archivo de configuración se encuentra en la raíz del proyecto y se llama App.config.

En este archivo se configura el *conectionString* con la base de datos y además se configura como

fecha de inicio del sistema el 07/11/2016.

**CONSIDERACIONES ASUMIDAS:**

● Cuando un usuario ingresa al sistema bajo el rol administrador, para poder utilizar todas

las funcionalidades, el id\_persona del usuario debe estar relacionado con la ENTIDAD

AFILIADO y PROFESIONAL, ya que, por ejemplo, para pedir un turno o registrar agenda

se necesita el id\_afiliado o el id\_matricula respectivamente.

● Las funcionalidades en el sistema son fijas, dependiendo a un rol del sistema se habilitan

o deshabilitan. Por ejemplo, un administrador posee todas las funcionalidades habilitadas.

● Los bonos que compra un afiliado pueden ser usados por familiares del mismo plan.

● Una persona (afiliado, médico, etc.) puede no tener un usuario en el sistema.

● Una persona puede ser afiliado, profesional o administrativo a la vez.

● Una persona puede no ser afiliado ni profesional (ej. usuario administrador)

● El numero de documento por sí sólo no es único por persona; depende del tipo.

● En el sistema anterior se compro un bono cuando el campo

Bono\_Consulta\_Fecha\_Impresion y Compra\_Bono\_Fecha no son nulos.

● En el sistema anterior se utilizo un bono cuando el campo

Bono\_Consulta\_Fecha\_Impresion es no nulo y Compra\_Bono\_Fecha es nulo.

● Existe la misma cantidad de turnos, bonos\_consulta y consultas realizadas, por ende, todos los

turno y bonos\_consultas que se encuentran en el sistema fueron utilizados.

●Ninguno de los profesionales en la tabla maestra posee rangos horarios de atención, es por eso que deben crearse agendas para cada uno.

●Para facilidad del usuario, se identifican a las consultas y turnos con la fecha.

**DER**

**ACLARACIONES SOBRE EL MODELO**

En el DER que entregamos anteriormente se agregaron atributos en las entidades ya existentes y nuevas entidades que fueron necesarias para la simplificación de la Solución, mencionamos unas modificaciones :

● #Usuario → login\_fallidos y habilitado

Es un contador de las veces que el usuario quiere ingresar al sistema ingresando la contraseña de forma incorrecta. Habilitado es un campo bit que permite saber si el usuario fue dado de baja del sistema (baja lógica).

● #Persona → id\_persona y atributos como nombre, apellido, nro\_doc y tipo\_de\_doc, etc.

Esta entidad nos ayuda a que no haya redundancia de atributos en afiliado y profesional, asi como relacionar estos con la tabla Usuario.

● #Afiliado → grupo\_afiliado, integrante\_grupo, nro\_afiliado

Estos atributos nos ayudan a distinguir el grupo familiar y a obtener el nro\_afiliado

● #Afiliado → activo, fecha\_baja\_logica

Activo es un campo bit que permite saber si el afiliado fue dado de baja del sistema (baja lógica). Si así lo fuere, no existe posibilidad de volverlo a cambiar.

● #Afiliado → nro\_consulta

Para facilitar la migración movimos este atributo de la entidad Bono\_Consulta a esta entidad , que nos dice cuantas consultas realizo el afiliado.

● #Estado\_Civil

Añadimos esta entidad para manejar enteros en vez de string.

● #Compra\_Bono

Añadimos esta entidad para registrar en el sistema las compras de un afiliado.

●#Tipo\_Cancelacion

Añadimos esta entidad para manejar enteros en vez de string.

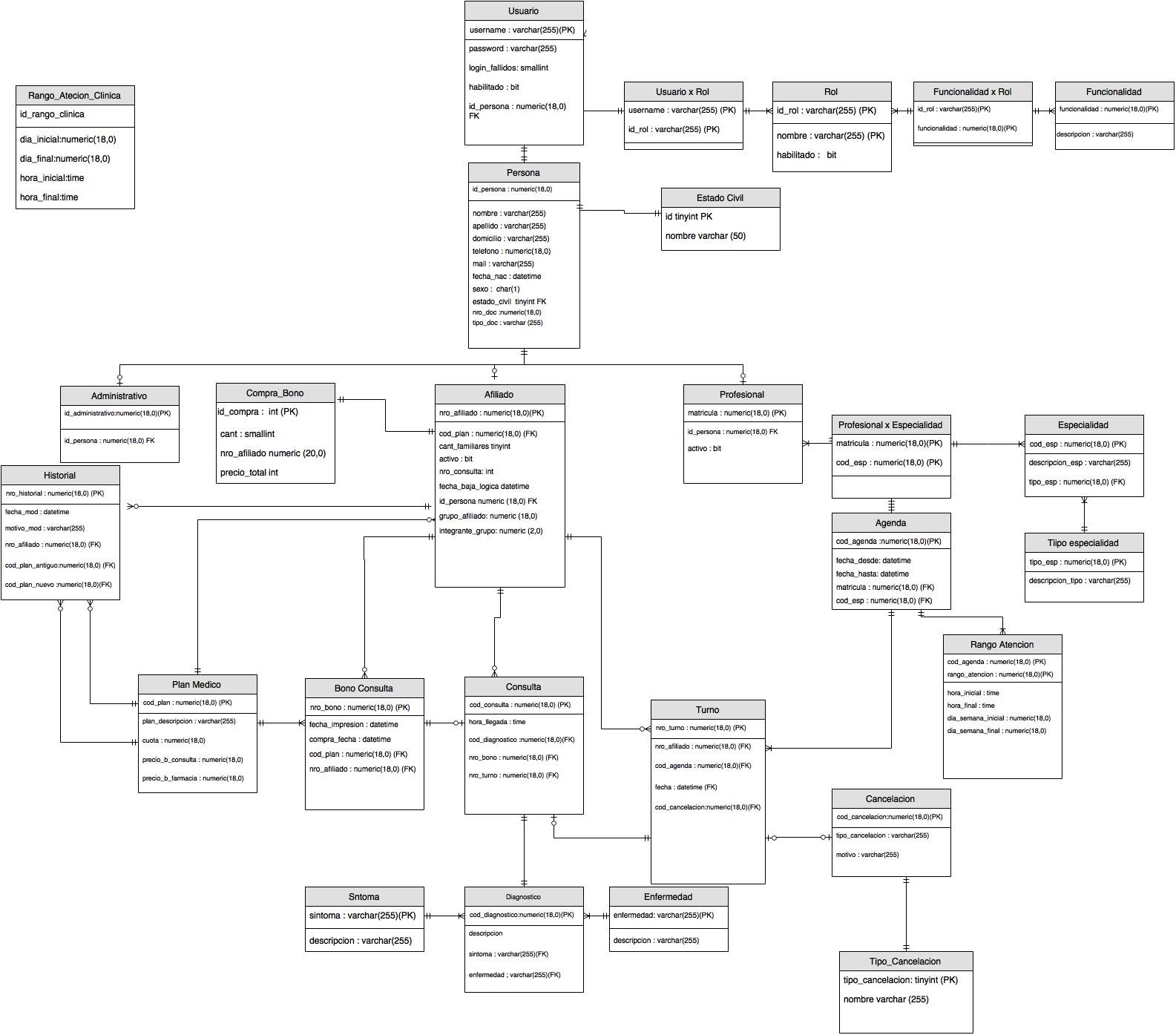
●#Agenda → fecha\_desde, fecha\_hasta

Atributos agregados para establecer el rango de fechas en donde la agenda es valida.

●#Rango\_atencion\_Clinica

Esta entidad nos ayuda a persistir el rango horario del sistema.

**IMAGEN :**



**MIGRACION**

Nuestra estrategia del proceso de migración se divide en varias etapas. Primeramente, creamos vistas auxiliares sobre la tabla Maestra para la tabla Persona del dominio que se migrará. Luego, realizamos consultas sobre estas vistas para insertar los datos en nuestro modelo. Esto nos permitió reducir la duplicación de datos.

Observamos que los números de documento en la tabla maestra son únicos. Aprovechamos esto para facilitar el proceso de migración, asignando el DNI como identificador único para las personas que aparecen en la tabla Maestra y un identificador numérico autogenerado para todas las personas que se ingresen posteriormente a la base de datos. Tomamos esta decisión porque el número de documento en la Argentina no es necesariamente único para cada persona.

Al no haber datos sobre los grupos de familia en la tabla maestra, decidimos que inicialmente cada afiliado esté en su propio grupo familiar con el plan que tenía anteriormente. Luego, sus familiares podrán unirse a ese grupo si lo desean.

Notamos que hay turnos en la tabla maestra que no deberían existir según los horarios de atención de la clínica. Para mitigar esto, migramos todos los turnos (asegurando de no poder agregar más turnos inválidos a posteriori). Es decir, todo turno en el futuro que caiga domingo, lunes antes de las 08:00, etc. se considera como inválido.

Algunas consideraciones correspondientes a la migración :

● No se dropea el esquema si existe porque no nos interesa que se borren todos los datos, sino que solamente los que vamos a utilizar. Además para eliminar el esquema necesitaríamos eliminar todo lo que tenía cargado previamente, cosa que es muy complicado distinguir.

● En varios lugares se usa SET IDENTITY\_INSERT NEXTGDD.NombreTabla ON. Lo que hace esto es permitir insertar valores en un campo IDENTITY. Cuando se inserte un nuevo campo el IDENTITY va a seguir funcionando de forma autoincremental. Gracias a esto pudimos relacionar entidades.

● Utilizamos el atributo matricula Primary key del profesional como Identity durante la migracion , una vez terminada la carga de datos de los profesionales utilizamos SET IDENTITY\_INSERT NEXTGDD.Profesional ON para permitir insertar datos manualmente.

● Decidimos que a todos los usuarios se les setea el tipo de documento como *‘DNI’* por default.

●Decidimos que a todas las personas se les setea el sexo como ‘X’, para facilitar la migración.

● Inicialmente, el código de un grupo familiar creado en la migración es el mismo que el

identificador único de la persona que lo conforma, que llamamos id\_persona.

● Inicialmente, el nro\_afiliado creado en la migración es el mismo que el

identificador único de la persona que lo conforma, que llamamos id\_persona , seguido de 01 .

Para forzar esto hicimos un casteo de los dos valores.

●Consideramos que se efectua la compra de un bono cuando los campos Compra\_Bono\_Fecha y Bono\_Consulta\_Fecha\_Impresion tengan la misma fecha y todos los demás campos en sean nulos.

●Consideramos la solicitud de turno cuando los campos Compra\_Bono\_Fecha, Bono\_Consulta\_Fecha\_Impresion, Consulta\_Sintoma y Consulta\_Enfermedades sean nulos, y todo lo demás contenga datos.

●Consideramos que cuando el campo Compra\_Bono\_Fecha sea nulo y todos los demás campos tengan datos se refiere a que se  efectivizó la consulta, registrando la utilización del bono y el diagnóstico del médico.

●El campo Turno\_Fecha de la tabla maestra lo consideramos en la migración como la fecha en que se solicito el turno , luego el campo fecha en nuestra entidad Turno corresponde a la fecha de la consulta.

**Aplicación Desktop**

**Conexión a la base de datos**

Se utiliza una clase estática que contiene todos los métodos para realizar los querys. Cada método instancia una conexión con el mismo string (obtenido a partir del App Config); sin embargo, cada instancia representa la misma conexión física, ya que el “connection pool” de .NET hace que se reúsen las conexiones.

**Login y Seguridad**

Mediante el login se puede iniciar sesión y utilizar las distintas funcionalidades del sistema.

Utilizamos una clase singleton para guardar la información de la sesión del usuario.

Usamos la función HASHBYTES bajo el algoritmo SHA2\_256 para encriptar la contraseña del usuario. Por defecto no asociamos a ningún profesional o afiliado con un usuario, salvo los que entregamos para la corrección.

Por default cuando se da de alta a un afiliado se crea el respectivo usuario mediante un stored Procedure para que este pueda ingresar al sistema en el futuro.

● username = ‘nro\_documento’

● contraseña = ‘nro\_afiliado’

Una vez que se loguea correctamente el usuario, si este tiene mas de un rol, se le mostrara en una pantalla con que rol desea ingresar al sistema, en caso de que tenga asignado un solo rol ingresara automáticamente a la pantalla principal del sistema (en donde están todas las funcionalidades del sistema)

**ABM**

Todas las funcionalidades del sistema están representadas mediante un combobox en una pantalla principal donde el sistema nos pregunta que funcionalidad deseamos ejecutar.

Los ABM que no se desarrollan simplemente no hacen nada.

**REGISTRO DE AGENDA**

Se comienza ingresando el profesional y las especialidades (las cuales se filtran por las que ese profesional posee). Se verifica si ya tiene una agenda con rangos horarios establecidos (ya que los profesionales de la tabla maestra poseen una, pero sin rangos de atención), y en caso de tenerla se emite un mensaje de error. Luego se cargan los días de atención de lunes a sábado (dado que el domingo no está dentro del rango de atención de la clínica), filtrando el “día hasta” para que solo pueda elegirse a partir del “día desde”. Según el rango horario en el que entra el “día hasta”, se cargan los horarios, y se filtra el “horario hasta” igual que con el día. Luego se ingresan en la tabla de rango de atención, permitiendo que se puedan borrar. Y finalmente se da la opción de elegir el rango de fechas. Una vez ingresada la agenda, se verifica si ya existe una, y en ese caso se actualizan los rangos; de lo contrario, se agrega en la base de datos, fijándose cuál es el identificador para luego ingresar cada rango horario correspondiente a esa agenda.

**PEDIDO DE TURNO**

Primero se verifica si el número de afiliado ingresado es correcto. Una vez que se selecciona la especialidad, se filtran los profesionales. Al seleccionarse el profesional, se filtran las fechas según el rango de fechas en el que atiende (de no tener un rango, no se aplica el filtro). Cuando se selecciona una fecha, se cargan los rangos horarios para ese día, si es que tiene, y sino se cargan los horarios dentro del rango horario de la clínica (por ejemplo, si se selecciono un sábado, solo se pueden elegir horarios entre las 10 y las 15, y si se selecciona un domingo, no hay horarios para seleccionar). Una vez que se ingresa el turno, se valida que no haya otro turno en el mismo horario y, si bien se aplicó el filtro, se verifica nuevamente que esté dentro del rango de fechas del profesional y dentro del rango horario. De no ser así, no se registra el nuevo turno, y se emite un mensaje de error para que el administrativo modifique los campos.

**REGISTRO DE LLEGADA**

Primero se selecciona el profesional (dando la opción de que se pueda filtrar por especialidad), y se cargan todos los turnos (a los que se identifica con la fecha, ya que puede ser que el afiliado no se acuerde del número de turno) de ese profesional en la fecha actual. Cuando se selecciona una fecha, se autocompletan los campos de fecha y hora, pudiéndose modificar si el afiliado llegó antes (en caso de llegar más tarde, se emite mensaje de error y no se permite registrar la consulta). Luego, teniendo el profesional, la especialidad y la fecha del turno (los cuales identifican a un único turno), se obtiene el numero de afiliado y se cargan todos los bonos disponibles, es decir, todos los que no se registraron en consultas anteriores (excepto aquellas canceladas) y las pertenecientes a familiares del mismo plan. Una vez seleccionado el bono, se registra la llegada, y se emite un número de consulta.

**REGISTRO DE RESULTADO DE ATENCIÓN MÉDICA**

Se cargan todas las consultas (identificadas por fecha) del profesional logueado, correspondientes a la fecha actual (ya que se asume que registra el resultado el mismo día que ocurre la consulta) y que no posean un diagnóstico; se permite la opción de filtrar por afiliado (dentro de los afiliados atendidos por ese profesional). Una vez seleccionada la consulta, se autocompletan los campos de fecha y hora de atención, pudiendo modificarse, en caso de no haberse efectuado en esa hora exacta. Luego se permite que el profesional seleccione una enfermedad y un síntoma (cargándose todas aquellas presentes en la base de datos), y que ingrese una descripción (la cual no es obligatoria). Una vez ingresado el diagnóstico, se registra en la base de datos, vinculándolo con la consulta correspondiente.

**LISTADO ESTADÍSTICO**

El usuario selecciona uno de los 5 posibles listados y un semestre (habiendo primero y segundo semestre de cada año registrado en la base de datos). Según el listado elegido, se activa una opción de filtro. Se decidió implementar así ya que tres de los listados poseían un único filtro, y los otros dos ninguno. Luego se muestran los resultados en una tabla, mostrando la cantidad de columnas correspondientes, y permitiendo limpiar la tabla, de forma de cargar un nuevo listado. No se muestra la opción de filtro en la tabla, ya que se repetiría para cada fila.

**USUARIO**

**Usuario AFILIADO @username = 'profesional', @password = 'w23e'**

**Usuario PROFESIONAL @username = 'afiliado', @password = 'w23e'**