

**Gestión de Datos**

**Trabajo Práctico 2° Cuatrimestre 2016**

Clínica – FRBA

Integrantes:

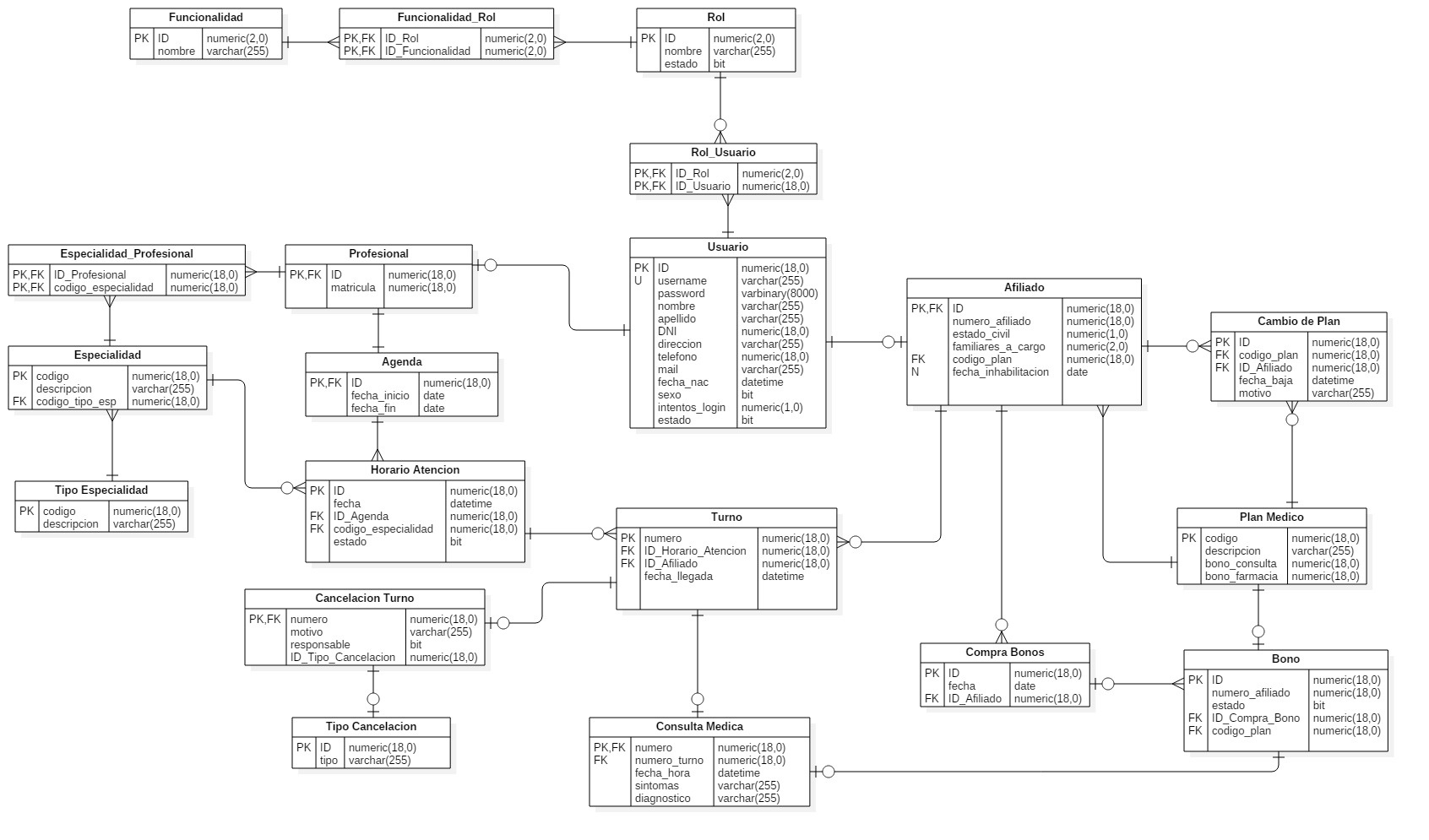
* Minian, Matias – 1473748
* Alice, Pablo – 1498587
* Fernandez, Federico – 1420525
* Montañez, Víctor - 2496641

**Índice**

DER ………………………………………………………………………………… 2

Aclaraciones Generales ...................................................... 3

DER:



Entidades:

Usuario:

* Entidad que representa a un afiliado o a un profesional.
* El username debe ser único, la password debe estar cifrada y el estado representa si está habilitado o no.
* Relaciones: Se relaciona con Profesional o Afiliado dependiendo de que tipo de usuario sea y con Rol Usuario para representar todos los roles que un usuario puede tener.
* PK: ID.

Rol Usuario:

* Relaciona cada usuario con sus roles.
* Relaciones: Se relaciona con Usuario y Rol.
* PK: ID\_Rol, ID\_Usuario.
* FK: ID\_Rol, ID\_Usuario.

Rol:

* Entidad que representa a un rol.
* Tiene un estado que muestra si está habilitado o no.
* Relaciones: Se relaciona con Rol\_Usuario donde están todos los roles que cada usuario tiene y con Funcionalidad Rol para representar las funcionalidades que tiene un rol.
* PK: ID.

Funcionalidad Rol:

* Relaciona cada rol con sus funcionalidades.
* Relaciones: Se relaciona con Rol y Funcionalidad.
* PK: ID\_Rol, ID\_Funcionalidad.
* FK: ID\_Rol, ID\_Funcionalidad.

Funcionalidad:

* Entidad que representa las funcionalides que un usuario puede tener dado el rol que el mismo posee.
* Relaciones: Se relaciona con Funcionalidad Rol donde se detallan las funcionalidades que cada rol tiene.
* PK: ID.

Afiliado:

* Entidad que representa datos particulares de un afiliado.
* El número de afiliado tiene una raíz en común para todo el grupo familiar y la fecha inhabilitación determina cuando se dió la baja lógica de ese afiliado.
* Relaciones: Se relaciona con Usuario, con Compra Bonos cuando se realizan compras de bonos, con Plan Médico donde tiene los datos concernientes a su plan y con Cambio de Plan donde se detallan todos los planes que tuvo anteriormente.
* PK: ID.
* FK: ID, codigo\_plan.

Cambio de Plan:

* Entidad que detalla todos los planes anteriores que un usuario tuvo.
* Relaciones: Se relaciona con Afiliado el cual hizo ese cambio de plan y con Plan Médico para saber que plan es.
* PK: ID.
* FK: codigo\_plan, ID\_Afiliado.

Plan Médico:

* Esta entidad representa los planes médicos que un afiliado puediera tener.
* Relaciones: Se relaciones con Cambio de Plan, con Afiliado para saber que plan tiene cada afiliado y con Bono para conocer los precios del bono y a que plan está asociado el bono comprado.
* PK: codigo.

Bono:

* Esta entidad representa los bonos que un afiliado puede comprar.
* El numero afiliado posee el numero de un afiliado para poder utilizar ese bono por todo el grupo familiar. El estado representa si el bono ya fue usado o esta disponible.
* Relaciones: Se relaciona con Compra Bonos para detallar todos los bonos adquiridos en una sola compra, con Plan Médico y con Consulta Médica para asociar un bono a una conuslta.
* PK: ID.
* FK: ID\_Compra\_Bono, codigo\_plan

Compra Bonos:

* Esta entidad representa una compra de bonos, pudiendo ser uno o más.
* Relaciones: Se relaciona con Afiliado para saber el afiliado que hizo la compra y con Bono para detallar cada bono comprado.
* PK: ID.
* FK: ID\_Afiliado.

|  |  |
| --- | --- |
| **Stored Procedure** | **Descripción** |
| obtenerFuncionalidades() | Obtener todas las funcionalidades. |
| obtenerRoles() | Obtener los roles con sus funcionalidades. |
| crearRol(nombreRol, Table[idFuncionalidad]) | Crear un rol con sus funcionalidades. |
| modificarRol(idRol, nombreRol, Table[idFuncionalidad]) | Modificar un rol. |
| habilitarRol(idRol) | Habilitar un rol. Pasar su estado a 1. |
| eliminarRol(idRol) | Baja lógica de un rol. |
| login(username, password) | Login de un usuario. Deshabilitar si llega a los 3 intentos fallidos. |
| obtenerRolesDeUsuario(idUsuario) | Obtener todos los roles de un usuario. |
| obtenerCantidadDeBonosDisponiblesPorAfiliado (idAfiliado) | Obtener la cantidad de bonos que un usuario tiene disponibles para su uso. |
| obtenerRaizAfiliado() | Obtener raíz para un nuevo afiliado. |
| crearAfiliado(username, password, nombre, apellido, dni, direccion, telefono, mail, fechaNac, sexo, estadoCivil, familiaresACargo, idPlanMedico, numDeAfiliado) | Crear afiliado con raíz para su número obtenida anteriormente. |
| modificarAfiliado(password, direccion, telefono, mail, sexo, estadoCivil, familiaresACargo, idPlanMedico, motivo) | Modificar un afiliado. |
| habilitarAfiliado(idAfiliado) | Habilitar afiliado. Pasar su estado a 1. |
| eliminarAfiliado(idAfiliado) | Baja lógica de un afiliado. |
| obtenerPlanesMedicos() | Obtener todos los planes médicos. |
| obtenerProfesionalPorDNI(dni) | Obtener profesional dado su DNI. |
| registrarAgenda(idProfesional, mesInicio, mesFin, Table[dia, horarioInicio, horarioFin, codigoEspecialidad]) | Registrar agenda de un profesional detallando los horarios que trabajará cada día. |
| obtenerBonosPorNumeroAfiliado(nroAfiliado) | Obtener los precios de los bonos para un afiliado dependiendo del plan que tenga. |
| comprarBonos(idAfiliado, nroAfiliado, codigoPlan, cantidad) | Realizar una compra de bonos por parte de un afiliado. |
| obtenerEspecialidades() | Obtener todas las especialidades. |
| obtenerProfesionalesPorEspecialidad(codigoEspecialidad) | Obtener profesionales de determinada especialidad. |
| obtenerHorariosDisponiblesParaFecha(idProfesional, codigoEspecialidad, fecha) | Obtener todos los horarios disponibles de un profesional para determinada fecha y especialidad. |
| registrarTurno(idAfiliado, idHorario) | Reservar un turno con un profesional en un determinado horario. |
| obtenerProfesionalesDelDiaPor(nombre, apellido, codEspecialidad, fecha) | Obtener los profesionales que atiendan ese día. |
| obtenerTurnosDeProfesional(idProfesional, fecha) | Obtener todos los turnos reservados que el profesional tiene para ese día. |
| registrarLlegada(idAfiliado, numeroTurno, fecha) | Registrar la llegada de un afiliado a la clínica. |
| registrarAtencionMedica(numeroTurno, fecha, sintomas, diagnostico) | Registrar datos relacionados con la consulta médica. |
| obtenerTurnosDeAfiliado(idAfiliado) | Obtener todos los turnos que un afiliado haya pedido. |
| obtenerTiposDeCancelacion() | Obtener tipos de cancelación. |
| cancelarTurnoPorAfiliado(idAfiliado, numeroTurno, tipoCancelacion, motivoCancelacion) | Cancelar turno de un afiliado. Debe realizarse con un día de antelación |
| cancelarDiaPorProfesional(idProfesional, fecha,tipoCancelacion,motivo) | Cancelar día de atención de un profesional. |
| cancelarTurnosPorProfesional(idProfesional, fechaInicial, fechaFinal,tipoCancelacion,motivo) | Se cancelan todos los turnos que esten dentro de ese rango de fechas. |
| topEspecialidadesMasCancelaciones() | Top 5 de especialidades con mas cancelaciones. |
| topProfesionalesMasConsultadosPorPlan() | Top 5 profesionales mas consultados por plan. |
| topProfesionalesMenosHoras(codigoPlan, codigoEspecialidad) | Top 5 profesionales con menos horas trabajadas. |
| topAfiliadosMasBonos() | Top 5 afiliados con mayor cantidad de bonos comprados. |
| topEspecialidadesMasBonosUsados() | Top 5 especialidades con mas bonos usados. |

Aclaraciones Generales:

* Para los antiguos usuarios tomamos el DNI como el username.
* Utilizamos tablas temporales para distribuir las funcionalidades que le corresponden a cada tipo de usuario para luego poder insertarlas en la tabla funcionalidades\_rol.
* Determinamos 4 tipos de cancelación posibles que se pueden elegir a la hora de cancelar un turno ya sea por parte de un afiliado como de un profesional.
* Determinamos que funcionalidades tiene cada rol, creando algunos roles por defecto para usuarios y administradores.