



CURSO:	K3022	
GRUPO:	“GESTIONAME LAS VACACIONES”	
INTEGRANTES:	CABANAS, JUAN IGNACIO	152.160-3
	CASARINI, YAMILA	152.643-1
	JANIN, IGLESIAS	152.280-2
	PEBES, MARIA LUZ	152.451-3
AÑO:	2016	

INDICE:

• DESCRIPCION DE TABLAS-----	3
• OBSERVACIONES -----	7
• VALIDACIONES -----	8
• DESARROLLO DEL TP -----	10
• DER -----	11

Descripción de las tablas

Funcionalidades:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- descripcion NVARCHAR(50),
 - Nombre de la funcionalidad para ser mostrado en el combo box al loguear el usuario y también en ABM Rol

Roles:

- id INTEGER IDENTITY(1,1) PRIMARY KEY ,
- descripcion NVARCHAR(50),
 - Descripción del rol (Administrador, Profesional, Afiliado, y el resto de los roles que sean creados en ABM Rol)
- baja INT DEFAULT 0,
 - Campo para identificar si el rol fue dado de baja en ABM Rol

Usuarios:

- usuario VARCHAR(255) PRIMARY KEY,
 - Nombre del usuario para el login
- pass VARCHAR(255)
 - Contraseña de usuario
 - Se usó como default el SHA_256 de **w23e** para todos los usuarios generados por la migración
(`'e6b87050bfc8143fcb8db0170a4dc9ed00d904ddd3e2a4ad1b1e8dc0fdc9be7'`),
- baja INT default 0
 - Campo para el registro de los usuarios dados de baja
- fechaBaja DATETIME,
 - Fecha de la baja
- intentos INT DEFAULT 0,
 - Cantidad de intentos fallidos por el usuario, este campo sirve por si el usuario excede la cantidad de intentos fallidos y que el mismo se bloquee

Planes:

- id INTEGER PRIMARY KEY,
- precioBono INT,
 - Este campo precio fue usado para la compra de bonos, mostrar el monto dentro del form y a su vez migrarlo a la tabla compra bonos
- precioCuota INT ,
 - El precio de la cuota del plan sacado de la migración
- descripcion VARCHAR(30),
 - El nombre del plan utilizado en distintas funcionalidades (ABM afiliado, Compra bono, etc.) donde especifica el nombre y el tipo de plan
- baja INT DEFAULT 0,
 - Campo donde se chequea si se dio de baja o no al plan

Pacientes:

- id INT PRIMARY KEY IDENTITY(1,1)
 - ID del paciente utilizado como pk sobre el que se puede realizar búsquedas y se utiliza para generar el usuario
- usuario VARCHAR (255) REFERENCES GESTIONAME_LAS_VACACIONES.Usuarios(usuario)
 - Referencia a la tabla Usuarios
- nombre NVARCHAR(50) NOT NULL
- apellido NVARCHAR(50) NOT NULL
- documento INT NOT NULL
- tipoDocumento VARCHAR(100) DEFAULT 'DNI'
 - Se puso DNI como default dado que en la tabla maestra no había tipoDni
- direccion VARCHAR(100) NOT NULL
- telefono INT NOT NULL
- email VARCHAR(255)
- fechaNacimiento DATETIME NOT NULL
- sexo CHAR
- estadoCivil VARCHAR(10)
- cantFamiliares INT DEFAULT 0
- cantConsultas INT DEFAULT 0
- planes INTEGER
- baja INT DEFAULT 0
 - Campo donde se chequea si el paciente fue dado de baja o no
- fechaBaja DATE
 - Fecha en la cual se fue de baja
- CHECK (sexo IN('f','m'))
 - Se chequea que el sexo sea masculino o femenino

Profesionales:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
 - Numero también de matrícula y sirve también para generar el nombre de usuario
- usuario VARCHAR (255) REFERENCES GESTIONAME_LAS_VACACIONES.Usuarios(usuario)
 - Referencia a la tabla usuarios
- nombre NVARCHAR(50) NOT NULL
- apellido VARCHAR(50) NOT NULL
- tipoDocumento VARCHAR
- documento INT NOT NULL
- direccion VARCHAR(255) NOT NULL
- telefono INT NOT NULL
- email VARCHAR(255)
- fechaNacimiento DATETIME NOT NULL
- sexo CHAR

Especialidades:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- descripcion VARCHAR(50)
 - Nombre de la especialidad
- tipoEspecialidad VARCHAR(50)
 - Especificación de la especialidad

Agendas:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- idProfesional INT REFERENCES GESTIONAME_LAS_VACACIONES.Profesionales(id)
- idEspecialidad INT REFERENCES GESTIONAME_LAS_VACACIONES.Especialidades(id)
- fechaInicio DATETIME NOT NULL DEFAULT '2015-01-03 09:00:00.000'
 - Fecha de inicio tomada por default para la agenda
- fechaFinal DATETIME NOT NULL DEFAULT '2015-12-12 15:00:00.000'
 - Fecha de fin tomada como default para la agenda
- diaInicio INT DEFAULT 1
- diaFin INT DEFAULT 5
 - Los días están organizados del 1 al 5 (de Lunes a Viernes)
- baja INTEGER DEFAULT 0
 - Campo destinado a la lógica de periodos de la agenda
- motivo VARCHAR(255)
 - Motivo de la baja

ComprasBonos:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- idPaciente INT REFERENCES GESTIONAME_LAS_VACACIONES.Pacientes(id)
- cantidad INT NOT NULL DEFAULT 1
- monto INT NOT NULL
- fecha DATETIME NOT NULL

Modificaciones: (Tabla dedicada a la persistencia de las modificaciones de plan por usuario)

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- idPaciente INT REFERENCES GESTIONAME_LAS_VACACIONES.Pacientes(id)
- idPlan INT
- motivo VARCHAR(255)
- fecha DATETIME NOT NULL

Turnos:

- id INTEGER PRIMARY KEY IDENTITY (1,1)
- idProfesional INT REFERENCES GESTIONAME_LAS_VACACIONES.Profesionales(id)
- especialidad VARCHAR(255)
- idPaciente INT REFERENCES GESTIONAME_LAS_VACACIONES.Pacientes(id)
- idAgenda INT REFERENCES GESTIONAME_LAS_VACACIONES.Agendas(id)

- fecha DATETIME
- baja INT default 0
- tipoCancelacion INT
 - Campo utilizado para distinguir si la cancelación fue hecha por el paciente o por el profesional del turno
- motivo VARCHAR(255)
 - Motivo de la cancelación
- esConsulta INT DEFAULT 0,
 - Usado para saber si se efectivizo el turno en una consulta médica (0 NO, 1 SI)

CREATE INDEX ix1_turnos ON GESTIONAME_LAS_VACACIONES.Turnos (idPaciente)

- Este índice fue creado para agilizar las consultas de la tabla turnos

Bonos:

- id INTEGER PRIMARY KEY
- idPaciente INT REFERENCES GESTIONAME_LAS_VACACIONES.Pacientes(id)
- idPlan INT REFERENCES GESTIONAME_LAS_VACACIONES.Planes(id)
- usado INT DEFAULT 0
 - 0 si fue usado en alguna consulta, 1 si no fue usado
- idCompraBono INT REFERENCES GESTIONAME_LAS_VACACIONES.ComprasBonos(id)
- idTurno INT REFERENCES GESTIONAME_LAS_VACACIONES.Turnos(id)

ConsultasMedicas:

- id INTEGER IDENTITY(1,1) PRIMARY KEY
- idBono INT REFERENCES GESTIONAME_LAS_VACACIONES.Bonos(id)
- fecha DATETIME NOT NULL
- diagnostico VARCHAR(255)
 - Descripción del diagnóstico luego de haber sido realizada la consulta
- sintomas VARCHAR(255)
 - Descripción de los síntomas del paciente
- idTurno INT REFERENCES GESTIONAME_LAS_VACACIONES.Turnos(id)

Observaciones:

- Para la migración de los pacientes, y consideramos que los últimos dos valores identificativos son reservados para utilizarse en un mismo bloque familiar, ejemplo:

Madre: 62035000 Hija: 62035001 No familiar: 62035100

- Al no poseer tal información de la migración, se ingresaron todos los pacientes ya existentes de forma creciente, y se realizara esta consideración como una modificación del plan de negocios después de haber realizado la migración.
- Solo se podrá buscar un paciente/profesional por nombre y apellido, o código de identificación 'id', mostrando con estos valores todas las personas con el mismo nombre y apellido, y demás datos. Esto es debido a que no es muy identificativo buscar por un rango de edad, género, estado social, domicilio, etc.
- La hora actual del sistema es cargada mediante configuración y está definida dentro del rango de la agenda de del profesional y los turnos registrados en la tabla maestra, para poder de esa forma utilizar el registro de llegada como también la cancelación de un turno
- Los usuarios para afiliado y profesional son “Afiliado_<id>” y “Profesional_<id>” respectivamente. Estos pueden ser ingresados tanto en minúscula como en mayúscula, esto fue tomado de redes sociales como Twitter o Hotmail. Y la contraseña será por default w23e. Dicho nombre de usuario es primary key de usuarios, dado a que no puede haber dos pacientes con el mismo id.
- Los pacientes migrados de la maestra fueron ingresados al sistema todos como pacientes independientes sin familiares
- Por default todos los pacientes tienen como tipo de documento DNI ya que en la migración no lo especifica
- El formato de fecha a ingresar en el archivo configuración es:

AAAA-dd-MM hh:mm:ss.fff

Validaciones

- Cuando un paciente es dado de baja, el usuario asociado a dicho paciente no se puede loggear.
- No se pueden crear roles ya existentes
- No se puede habilitar un rol ya habilitado, y si se hace, muestra un loggeo de que ya fue habilitado
- El rol paciente fue cargado con las siguientes funcionalidades:
 - Pedir turno
 - Comprar bonos
 - Cancelación
- El rol profesional fue cargado con las siguientes funcionalidades:
 - Resultado de consulta
 - Cancelación
 - Alta agenda profesional
- No se puede cargar un afiliado que ya exista, y si se hace loggea un error, esta verificación se hace según su DNI
- No se puede cargar un familiar que ya exista, y si se hace se loggea un error
- No se pueden cargar más de 99 familiares
- No se pueden modificar roles o funcionalidades inexistentes, y si se hace, se loggea un error
- El paciente no puede registrar una llegada si no posee bonos, y si pasa,, se loggea un error
- Se valida que al cancelar un turno por parte de un afiliado, estos existan
- Se valida que si el profesional cancela una fecha, esta esté dentro de su intervalo en la agenda.
- Dentro de los buscadores, validamos que no se apriete el botón buscar, sin tener al menos un campo dentro de la búsqueda.

- Cuando se ingresa un usuario, sea afiliado o paciente, automáticamente todas las operaciones se hacen en referencia al paciente loggeado, no teniendo que volverse a buscar dentro de la aplicación.
- En el alta afiliado se valida valor a valor, es decir:
 - Nombre y apellido solo pueden ingresar letras
 - Tipo de documento, posee todos los tipos admitidos por el enunciado
 - Nro. de documento, cantidad de familiares y teléfono, solo admiten números
 - Mail, es necesario poner un punto y un arroba
 - Sexo y estado civil y plan tienen un combo box con todos los valores posibles.
 - Cada vez que se sale de un form ya sea por el botón salir o cerrando la ventana, el estado queda consistente.

Desarrollo del tp

- Para la elaboración del TP comenzamos pensando el dominio de nuestra aplicación diseñándolo en un DER.
- Luego de creado el script inicial y todo lo representado en el DER, realizamos la migración de la tabla Maestra proporcionada por la catedra a las tablas de nuestro modelo de negocio. Generamos la funcionalidad de “dropeo” para poder ejecutar el script sin errores. Por último, diseñamos las funcionalidades pedidas en procedures.
- En Visual Studio diseñamos **BuscadorProfesionales**, **BuscadorAfiliados**, **BuscarModificaciones**, **BuscarConsultas** y **BuscarConsultas** los cuales son usados por gran parte de las funcionalidades. Estos nos permiten hacer las búsquedas por distintos filtros, independientemente de la funcionalidad por la que fue llamada.
- Creamos las clases **Paciente**, **Profesional**, **Modificación** (para modificaciones de plan), **Funcionalidad**, **Rol**, **Turno**, **Agenda**, **Especialidad**, **Plan**, **ProfesionalesXConsulta** (Utilizados para los listados) y **Consulta**, para cosificar el resultado de las búsquedas y poder trabajarlas más cómodamente.
- A su vez se crearon **AfiliadoManager**, **RolManager**, **CancelacionManager**, **ListadosManager**, **ProfesionalManager**, **TurnosManager**, **Consultas Manager**, en los cuales se delegó la lógica importante de cada una de las entidades, así como el desarrollo de las listas y otras queries.
- Por temas de seguridad optamos por realizar que todos los pedidos complejos desde Visual Studio se hagan por medio de procedures o functions.
- Al querer seleccionar ciertos turnos según una condición el algoritmo de búsqueda era secuencial lo cual tardaba una cantidad considerable de tiempo. Para solucionar esto optamos por introducir un índice sobre el campo “idPaciente” a la tabla Turnos asumiendo el costo del espacio físico que ello requiere.
- Agregamos un trigger para cada vez que se agregue un valor en la tabla modificaciones, cambie el plan del afiliado.
- Se crearon las tablas temporales **#PacienteTemporal**, **#TemporalProfesional** y **#ConsultasTemporal** para la migración.

