

PROCEDIMIENTOS ALMACENADOS PL/SQL

PROYECTO: EXPEDIENTE MEDICO CENTRALIZADO

Nombre del Procedimiento/Función	PL_CrearAmbulancia
Descripción	PL/SQL para la creación de ambulancias de un centro Hospitalario
Parámetros de Entrada	Placa VARCHAR2 idCentroMedico INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Añadir a la base de datos una ambulancia , relacionada con su respectivo centro medico. Validar que el idCentroMedico exista Mostrar mensaje de éxito o Error en su defecto.	

Nombre del Procedimiento/Función	PL_ActualizarAmbulancia
Descripción	PL/SQL para la actualización de datos en los registros de Ambulancia.
Parámetros de Entrada	idAmbulancia INTEGER placa VARCHAR2 idCentroMedico INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Verificar la coincidencia del idAmbulancia a modificar En caso de match añadir los cambios (idAmbulancia placa idCentroMedico) Y mostrar mensaje de modificación con éxito,caso contrario mostrar mensaje de error.	

Nombre del Procedimiento/Función	PL_CrearAtencionPH
Descripción	PL/SQL para ingresar registros en la base de datos referentes a las atenciones pre-hospitalarias.
Parámetros de Entrada	Observación VARCHAR2 idParamedico INTEGER idAmbulancia INTEGER idExpediente INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idParamedico, idAmbulancia,idExpediente exista Añadir a la base de datos una atencion , relacionada con su respectivo paramédico,. Mostrar mensaje de éxito o Error en su defecto.	

Nombre del Procedimiento/Función	PL_ActualizarAtencionPH
Descripción	PL/SQL para modificar un registro almacenado en los registros de las atenciones pre-hospitalarias.
Parámetros de Entrada	IDATETIME encion INTEGER Observación VARCHAR2 IdParamedico INTEGER paramedico VARCHAR2 idAmbulancia INTEGER idExpediente INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Verificar la coincidencia del iDATETIME encion,idParamedico, idAmbulancia, idExpediente a modificar. Modificar en la base de datos de una Atencion pre-hospitalaria. Mostrar mensaje de éxito o Error en su defecto.	

Nombre del Procedimiento/Función	PL_CrearCentroMedico
Descripción	PL/SQL para añadir un nuevo centro medico
Parámetros de Entrada	Nombre VARCHAR2 dirección VARCHAR2 idTipoCentro INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idTipoCentro a relacionar exista. añadir los campos a la base de datos mostrar mensaje de éxito, caso contrario mostrar mensaje de error.	

Nombre del Procedimiento/Función	PL_ActualizarCentroMedico
Descripción	PL/SQL para modificar los registros existentes en la base de datos correspondientes a los centros medicos.
Parámetros de Entrada	idCentroMedico INTEGER nombre VARCHAR2 dirección VARCHAR2 idTipoCentro INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idCentroMedico, idTipoCentro exista En caso de match añadir los cambios (idCentroMedico nombre dirección idTipoCentro) Añadir a la base de datos los cambios del centro medico , con su respectivas relaciones. Mostrar mensaje de éxito o Error en su defecto.	

Nombre del Procedimiento/Función	PL_ AgregarCirugia
Descripción	PL/SQL para añadir un nuevo registro de cirugía con sus respectivas relaciones
Parámetros de Entrada	idIngreso INTEGER idTipoCirugia INTEGER idMedico INTEGER fechaHora DATETIME
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idTipoCirugia, idMedico a relacionar exista. añadir los campos a la base de datos mostrar mensaje de éxito, caso contrario mostrar mensaje de error.	

Nombre del Procedimiento/Función	PL_ActualizarTipoCirugia
Descripción	PL/SQL para modificar un registro almacenado.
Parámetros de Entrada	idTipoCirugia INTEGER tipoCirugia VARCHAR2
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Verificar la coincidencia del idTipoCirugia a modificar. Modificar en la base de datos de un tipo de cirugía. Mostrar mensaje de éxito o Error en su defecto.	

Nombre del Procedimiento/Función	PL_CrearConsultaExterna
Descripción	PL/SQL para ingresar registros de consulta externa.
Parámetros de Entrada	idConsultorio INTEGER idExpediente INTEGER idMedico INTEGER fechaHora 'DATETIME' síntomas VARCHAR2 diagnostico VARCHAR2 observación VARCHAR2
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idExpediente, idMedico exista Añadir a la base de datos una consulta con sus respectivas relaciones verificadas. Mostrar mensaje de éxito al agregar o Error en su defecto.	

Nombre del Procedimiento/Función	PL_ ActualizarConsultaExterna
Descripción	PL/SQL para la actualización de datos en los registros de ConsultaExterna.
Parámetros de Entrada	idConsulta INTEGER idExpediente INTEGER idMedico INTEGER fechaHora ' DATETIME ' síntomas VARCHAR2 diagnostico VARCHAR2 observación VARCHAR2
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Verificar la coincidencia del idConsulta a modificar En caso de match añadir los cambios (idConsulta idExpediente idMedico) mostrar mensaje de modificación con éxito, caso contrario mostrar mensaje de error.	

Nombre del Procedimiento/Función	PL_CrearConsultorio
Descripción	PL/SQL para la crear una nueva relación entre consultorio y piso
Parámetros de Entrada	idPiso INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Validar que el idPiso exista Añadir a la base de datos un nuevo consultorio. Mostrar mensaje de éxito al agregar o Error en su defecto.	

Nombre del Procedimiento/Función	PL_ ActualizarConsultorio
Descripción	PL/SQL para la actualización de datos en los registros de Consultorio.
Parámetros de Entrada	idConsultorio INTEGER idPiso INTEGER
Parámetros de Salida	Mensaje VARCHAR2 Resultado TINYINT
Descripción del Algoritmo Verificar la coincidencia del idConsultorio a modificar En caso de match añadir los cambios (idConsultorio idPiso) Y mostrar mensaje de modificación con éxito, caso contrario mostrar mensaje de error.	

PROCEDIMIENTOS ALMACENADOS

1. Nombre del Procedimiento/Función	crearEdificio
Descripción	Procedimiento para crear edificio.
Parámetros de Entrada	nombre: VARCHAR2 , idCentroMedico: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none">• Comprobar tipos de datos de entrada• Comprobar si existe centro médico• Si existe, insertar edificio en tabla Edificio• Retornar mensaje de resultado	

2. Nombre del Procedimiento/Función	actualizarEdificio
Descripción	Procedimiento para actualizar edificio.
Parámetros de Entrada	idEdificio: INTEGER ,, nombre: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none">• Comprobar tipos de datos de entrada• Comprobar si existe edificio• Si existe, actualizar los datos en tabla Edificio• Retornar mensaje de resultado	

3. Nombre del Procedimiento/Función	agregarPiso
Descripción	Procedimiento para agregar piso a edificio.
Parámetros de Entrada	idEdificio: INTEGER , descripcion: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none">• Comprobar tipos de datos de entrada• Comprobar si existe edificio• Si existe, insertar nuevo Piso con la descripción brindada.• Retornar mensaje de resultado	

4. Nombre del Procedimiento/Función	actualizarPiso
Descripción	Procedimiento para actualizar piso, cambio de nombre o edificio.
Parámetros de Entrada	idpiso: INTEGER , descripcion: VARCHAR2 , idEdificio: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada • Comprobar si existe edificio y el piso • Si existe, actualizar el piso con los datos nuevos. • Retornar mensaje de resultado 	

5. Nombre del Procedimiento/Función	crearEmergencia
Descripción	Procedimiento para insertar emergencia en expediente del paciente.
Parámetros de Entrada	observacion: VARCHAR2 , fechaHoraAtencion: DATETIME , idExpediente: INTEGER , idAtencion: INTEGER , idCentroMedico: INTEGER , idMedico: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada • Verificar idCentroMedico en tabla CentroMedico • Verificar idMedico en tabla Medico • Verificar idExpediente en Tabla de Expediente • Si el paciente llega en ambulancia enlazar el idAtencion de AtencionPreHospitalaria, verificar • Si todo sale bien, insertar el registro. • Retornar mensaje de resultado 	

6. Nombre del Procedimiento/Función	actualizarEmergencia
Descripción	Procedimiento para modificar registro de emergencia en la base de datos.
Parámetros de Entrada	idIngreso: INTEGER , observacion: VARCHAR2 , fechaHoraAtencion: DATETIME , idExpediente: INTEGER , idAtencion: INTEGER , idCentroMedico: INTEGER , idMedico: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada • Verificar idCentroMedico en tabla CentroMedico • Verificar idMedico en tabla Medico • Verificar idExpediente en Tabla de Expediente • Si el paciente llega en ambulancia enlazar el idAtencion de AtencionPreHospitalaria, verificar • Si no fallan las verificaciones, actualizar el registro con los datos nuevos. • Retornar mensaje de resultado 	

7. Nombre del Procedimiento/Función	actualizarTipoEnfermedad
Descripción	Procedimiento para modificar un registro en tabla TipoEnfermedad.
Parámetros de Entrada	idTipoEnfermedad: INTEGER , tipoEnfermedad: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada • Comprobar si existe idTipoEnfermedad • Si existe, actualizar el registro • Retornar mensaje de resultado 	

8. Nombre del Procedimiento/Función	crearEnfermedad
Descripción	Procedimiento para añadir una nueva enfermedad a tabla Enfermedad.
Parámetros de Entrada	enfermedad: VARCHAR2 , idTipoEnfermedad: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idTipoEnfermedad. • Si existe, insertar registro en tabla Enfermedad. • Retornar mensaje de resultado. 	

9. Nombre del Procedimiento/Función	actualizarEnfermedad
Descripción	Procedimiento para añadir una nueva enfermedad a tabla Enfermedad.
Parámetros de Entrada	idEnfermedad: INTEGER , enfermedad: VARCHAR2 , idTipoEnfermedad: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idTipoEnfermedad. • Comprobar idEnfermedad en tabla Enfermedad. • Si existe, actualizar registro en tabla Enfermedad. • Retornar mensaje de resultado. 	

10. Nombre del Procedimiento/Función	diagnosticarEnfermedad
Descripción	Procedimiento para añadir una nueva enfermedad a tabla Enfermedad diagnosticada a paciente.
Parámetros de Entrada	idEnfermedad: INTEGER , idMedico: INTEGER , fechaDiagnostico: DATETIME , idExpediente: INTEGER , idConsulta: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idEnfermedad, idMedico, idExpediente, idConsulta • Si existe, insertar registro en tabla EnfermedadConsulta con estado TRUE. • Retornar mensaje de resultado. 	

11. Nombre del Procedimiento/Función	quitarDiagnostico
Descripción	Marcar una enfermedad que padecía un paciente como curada
Parámetros de Entrada	idEnfermedad: INTEGER , idExpediente: INTEGER , idConsulta: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idEnfermedad, idExpediente, idConsulta • Si existe, actualizar registro en tabla EnfermedadConsulta con estado FALSE. • Retornar mensaje de resultado. 	

12. Nombre del Procedimiento/Función	actualizarTipoExamen
Descripción	Modificar valores de un tipo de examen.
Parámetros de Entrada	idTipoExamen: VARCHAR2 , descripcion: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idTipoExamen • Si existe, actualizar registro en la columna descripción de la tabla TipoExamen • Retornar mensaje de resultado. 	

13. Nombre del Procedimiento/Función	crearExamen
Descripción	Inserta un nuevo examen en expediente de paciente
Parámetros de Entrada	urlDocumento: VARCHAR2 , idTipo: INTEGER , idCentroMedico: INTEGER , observacion: VARCHAR2 , idExpediente: INTEGER , fecha: DATETIME ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idTipo en tabla de TipoExamen, • Comprobar existencia idCentroMedico , idExpediente en tablas correspondientes. • Si existe, actualizar registro en la columna descripción de la tabla TipoExamen • Retornar mensaje de resultado. 	

14. Nombre del Procedimiento/Función	actualizarExamen
Descripción	Inserta un nuevo examen en expediente de paciente
Parámetros de Entrada	idExamen: INTEGER , urlDocumento: VARCHAR2 , idTipo: INTEGER , idCentroMedico: INTEGER , observacion: VARCHAR2 , idExpediente: INTEGER , fecha: DATETIME
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idTipoExamen, idCentroMedico, • Si existe, actualizar registro en la tabla Examen • Retornar mensaje de resultado. 	

15. Nombre del Procedimiento/Función	crearHojaTrabajoSocial
Descripción	Inserta un nuevo registro de HojaTrabajoSocial en expediente de paciente
Parámetros de Entrada	descripcion: VARCHAR2 , idExpediente: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> Comprobar tipos de datos de entrada. Comprobar si existe idExpediente Si existe, insertar registro en la tabla HojaTrabajoSocial Retornar mensaje de resultado. 	

16. Nombre del Procedimiento/Función	actualizarHojaTrabajoSocial
Descripción	Actualizar un nuevo registro de HojaTrabajoSocial en expediente de paciente
Parámetros de Entrada	idTS: INTEGER , descripcion: VARCHAR2 , idExpediente: INTEGER , idCentroMedico: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> Comprobar tipos de datos de entrada. Comprobar si existe idExpediente, idTS, idCentroMedico Si existe, actualizar registro en la tabla HojaTrabajoSocial Retornar mensaje de resultado. 	

17. Nombre del Procedimiento/Función	crearHospitalizacion
Descripción	Insertar un nuevo registro de Hospitalización en expediente de paciente
Parámetros de Entrada	observacion: VARCHAR2 , fechaHoraIngreso: DATETIME , fechaHoraAlta: DATETIME , idPiso: INTEGER , cama: VARCHAR2 , idMedico: INTEGER , idExpediente: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> Comprobar tipos de datos de entrada. Comprobar si existe idPiso, idMedico, idExpendiente Si existe, insertar registro en la tabla Hospitalizacion Retornar mensaje de resultado. 	

18. Nombre del Procedimiento/Función	darAlta
Descripción	Dar de alta un paciente en tabla de Hospitalizacion
Parámetros de Entrada	idIngreso: INTEGER , fechaHoraAlta: DATETIME
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idIngreso • Si existe, actualizar fechaHoraAlta con el valor correspondiente. • Retornar mensaje de resultado. 	

19. Nombre del Procedimiento/Función	crearMedico
Descripción	Insertar un nuevo registro en tablaMedico
Parámetros de Entrada	pNombre: VARCHAR2 , sNombre: VARCHAR2 , pApellido: VARCHAR2 , sApellido: VARCHAR2 , direccion: VARCHAR2 , sexo: (F M), noidentidad: VARCHAR2 , idPais: INTEGER , idEspecialidad: VARCHAR2 , noColegiacion: VARCHAR2 , correo: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar validez idPais, idEspecialidad en tablas correspondiente • Comprobar si ya existe Persona con noidentidad <ul style="list-style-type: none"> a. Si existe, almacenar temporalmente el idPersona y saltar a (A) b. Si no existe continuar • Insertar un nuevo registro en tabla Persona, almacenar temporalmente el idPersona generado en la inserción. • (A) Insertar los valores en los campos correspondientes a la tabla Medico • Devolver mensaje y resultado. 	

20. Nombre del Procedimiento/Función	actualizarMedico
Descripción	Actualizar registro en tabla médico
Parámetros de Entrada	idMedico: INTEGER , direccion: VARCHAR2 , idEspecialidad: VARCHAR2 , noColegiacion: VARCHAR2 , correo: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idMedico, idEspecialidad • Si existe, actualizar registro en tabla medico • Retornar mensaje de resultado. 	

21. Nombre del Procedimiento/Función	crearPaciente
Descripción	Insertar un nuevo registro en tabla Paciente
Parámetros de Entrada	pNombre: VARCHAR2 , sNombre: VARCHAR2 , pApellido: VARCHAR2 , sApellido: VARCHAR2 , direccion: VARCHAR2 , noldentidad: VARCHAR2 , idPais: INTEGER , sexo: (F M), correo: VARCHAR2 , idTipoSangre: INTEGER , idEscolaridad: INTEGER , idOcupacion: INTEGER , idEstadoCivil: INTEGER , idAscendencia: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar validez idPais, idTipoSangre, idEscolaridad, idOcupacion, idEstadoCivil, idAscendencia en tablas correspondiente • Comprobar si ya existe Persona con noldentidad <ul style="list-style-type: none"> a. Si existe, almacenar temporalmente el idPersona y saltar a (A) b. Si no existe continuar • Insertar un nuevo registro en tabla Persona, almacenar temporalmente el idPersona generado en la inserción. • (A) Insertar los valores en los campos correspondientes a la tabla Paciente • Devolver mensaje y resultado. 	

22. Nombre del Procedimiento/Función	actualizarPaciente
Descripción	Actualizar registro en tabla Paciente
Parámetros de Entrada	idPaciente: INTEGER , direccion: VARCHAR2 , correo: VARCHAR2 , idEscolaridad: INTEGER , idOcupacion: INTEGER , idEstadoCivil: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si existe idPaciente, idEscolaridad, idOcupacion, idEstadoCivil • Si existe, actualizar registro en tabla Paciente • Retornar mensaje de resultado. 	

23. Nombre del Procedimiento/Función	crearParamedico
Descripción	Insertar un nuevo registro en tabla Paramedico
Parámetros de Entrada	pNombre: VARCHAR2 , sNombre: VARCHAR2 , pApellido: VARCHAR2 , sApellido: VARCHAR2 , direccion: VARCHAR2 , sexo: (F M), noldentidad: VARCHAR2 , correo: VARCHAR2 idPais: INTEGER , licencia; VARCHAR2 ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si ya existe Persona con noldentidad <ul style="list-style-type: none"> a. Si existe, almacenar temporalmente el idPersona y saltar a (A) b. Si no existe continuar • Insertar un nuevo registro en tabla Persona, almacenar temporalmente el idPersona generado en la inserción. • (A) Insertar los valores en los campos correspondientes a la tabla Paramedico • Devolver mensaje y resultado. 	

24. Nombre del Procedimiento/Función	actualizarParamedico
Descripción	Actualizar registro en tabla Paramedico
Parámetros de Entrada	direccion: VARCHAR2 , correo: VARCHAR2 licencia; VARCHAR2 ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Efectuar los cambios. • Retornar mensaje de resultado. 	

25. Nombre del Procedimiento/Función	crearReferencia
Descripción	Insertar un nuevo registro en tabla Referencia
Parámetros de Entrada	descripcion: VARCHAR2 , idMedico: INTEGER , idExpediente: INTEGER , idCentroMedicoRemite: INTEGER , idCentroMedicoRecibe: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si ya existe medico con idMedico, el expediente con idExpediente, referencias de los centros con idCentroMedicoRemite, idCentroMedicoRecibe <ul style="list-style-type: none"> a. Si existen, almacenar temporalmente y pasar a (A) b. Si no existe alguno mostrar mesj de error • (A) verificar que idCentroMedicoRecibe no sea igual a CentroMedicoRemite • Insertar un nuevo registro en tabla Referencia, almacenar temporalmente la información e irgenerado en la inserción. • Devolver mensaje y resultado. 	

26. Nombre del Procedimiento/Función	actualizarReferencia
Descripción	Actualizar registro en tabla Referencia
Parámetros de Entrada	idReferencia: INTEGER , descripcion: VARCHAR2 , idMedico: INTEGER , idExpediente: INTEGER , idCentroMedicoRemite: INTEGER , idCentroMedicoRecibe: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

27. Nombre del Procedimiento/Función	AgregarPersona
Descripción	Insertar un nuevo registro en tabla persona
Parámetros de Entrada	idPersona: INTEGER , telefono: VARCHAR2 , idTipoTelefono: INTEGER , idPais: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar validez idPersona, idTipoTelefono en tablas correspondiente • Comprobar si ya existe Persona con noldentidad, idTipoTelefono y idPais <ul style="list-style-type: none"> a. Si existen todos, almacenar temporalmente la información y pasar al paso (A) b. Si no existe continuar • Mostrar mensaje de error • Retener la información ingresada y comenzar el proceso para añadir el o los campos que no se encuentran registrados , en sus respectivas tablas. • (A) Insertar los valores en los campos correspondientes a la tabla telefono • Devolver mensaje y resultado. 	

28. Nombre del Procedimiento/Función	AgregarCentro
Descripción	Insertar un nuevo registro en tabla telefono
Parámetros de Entrada	idCentroMedico: INTEGER , telefono: VARCHAR2 , idTipoTelefono: INTEGER , idPais: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar si ya existe centro medico con idCentroMedico. • Validar los campos de tablas catalogo como ser idTipoTelefono, idPais (campos con scrollback select, por lo que no hay que verificar si existe) <ul style="list-style-type: none"> a. Si existen, almacenar temporalmente y pasar a (A) b. Si no existe alguno mostrar msj de error. • (A) Consecuentemente posibilitar la información para Insertar un nuevo registro en tabla telefono, almacenar temporalmente la información e Ir generado en la inserción. • Devolver mensaje y resultado. 	

29. Nombre del Procedimiento/Función	AgregarTipoTratamiento
Descripción	Insertar un nuevo registro en tabla tratamiento.
Parámetros de Entrada	idTipoTratamiento: INTEGER , tipoTratamiento: VARCHAR2 , idTratamiento: INTEGER , dosis: VARCHAR2 , INTEGER ervaloTiempo: VARCHAR2 , fechaInicio: DATETIME , duracionTratamiento: VARCHAR2 idViaSuministro: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Comprobar idTipoTratamiento • Comprobar si ya existe Persona con idtratamiento,idviasuministro en tablas correspondiente <ul style="list-style-type: none"> a. Si existen todos, almacenar temporalmente la información y pasar al paso (A) b. Si no existe continuar • Mostrar mensaje de error • Retener la información ingresada y comenzar el proceso para añadir el o los campos que no se encuentran registrados , en sus respectivas tablas. • (A) Insertar los valores en los campos correspondientes a la tabla tratamiento • Devolver mensaje y resultado. 	

30. Nombre del Procedimiento/Función	actualizarTipoTratamiento
Descripción	Actualizar registro en tabla tratamiento
Parámetros de Entrada	idTipoTratamiento: INTEGER , tipoTratamiento: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

31. Nombre del Procedimiento/Función	agregarViaSuministro
Descripción	Actualizar registro en tabla via Suministro
Parámetros de Entrada	idViaSuministro: INTEGER , viaSuministro: VARCHAR2
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

32. Nombre del Procedimiento/Función	crearTratamiento
Descripción	Agregar registro en tabla de tratamiento
Parámetros de Entrada	dosis: VARCHAR2 , INTEGER ervaloTiempo: VARCHAR2 , fechaInicio: DATETIME , duracionTratamiento: VARCHAR2 , idTipoTratamiento: INTEGER , idViaSuministro: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

33. Nombre del Procedimiento/Función	actualizarTratamiento
Descripción	Actualizar registro en tabla de tratamiento
Parámetros de Entrada	idTratamiento: INTEGER , dosis: VARCHAR2 , INTEGER ervaloTiempo: VARCHAR2 , fechaInicio: DATETIME , duracionTratamiento: VARCHAR2 , idTipoTratamiento: INTEGER , idViaSuministro: INTEGER ,
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

34. Nombre del Procedimiento/Función	recetar
Descripción	Agregar receta de Consulta a Expediente de Paciente
Parámetros de Entrada	idTratamiento: INTEGER , idConsulta: INTEGER , idMedico: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	

35. Nombre del Procedimiento/Función	borrarReceta
Descripción	Eliminar receta de Consulta a Expediente de Paciente
Parámetros de Entrada	idTratamiento: INTEGER , idConsulta: INTEGER , idMedico: INTEGER
Parámetros de Salida	mensaje: VARCHAR2 , resultado: TINYINT
Descripción del Algoritmo <ul style="list-style-type: none"> • Comprobar tipos de datos de entrada. • Verificar que exista cada registro relacionado • Efectuar los cambios o el error especificado. • Retornar mensaje de resultado. 	