|  |  |
| --- | --- |
| 1. **Nombre del** **Procedimiento/Función** | crearEdificio |
| Descripción | Procedimiento para crear edificio. |
| Parámetros de Entrada | nombre: 'char',  idCentroMedico: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada * Comprobar si existe centro médico * Si existe, insertar edificio en tabla Edificio * Retornar mensaje de resultado | |

**PROCEDIMIENTOS ALMACENADOS**

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarEdificio |
| Descripción | Procedimiento para actualizar edificio. |
| Parámetros de Entrada | idEdificio:int,,  nombre: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada * Comprobar si existe edificio * Si existe, actualizar los datos en tabla Edificio * Retornar mensaje de resultado | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | agregarPiso |
| Descripción | Procedimiento para agregar piso a edificio. |
| Parámetros de Entrada | idEdificio:int,  descripcion: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada * Comprobar si existe edificio * Si existe, insertar nuevo Piso con la descripción brindada. * Retornar mensaje de resultado | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarPiso |
| Descripción | Procedimiento para actualizar piso, cambio de nombre o edificio. |
| Parámetros de Entrada | idpiso:int,  descripcion: 'char',  idEdificio: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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 | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearEmergencia |
| Descripción | Procedimiento para insertar emergencia en expediente del paciente. |
| Parámetros de Entrada | observacion: 'char',  fechaHoraAtencion: 'date',  idExpediente: int,  idAtencion: int,  idCentroMedico: int,  idMedico: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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 | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarEmergencia |
| Descripción | Procedimiento para modificar registro de emergencia en la base de datos. |
| Parámetros de Entrada | idIngreso: int,  observacion: 'char',  fechaHoraAtencion: 'date',  idExpediente: int,  idAtencion: int,  idCentroMedico: int,  idMedico: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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 | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarTipoEnfermedad |
| Descripción | Procedimiento para modificar un registro en tabla TipoEnfermedad. |
| Parámetros de Entrada | idTipoEnfermedad: int,  tipoEnfermedad: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada * Comprobar si existe idTipoEnfermedad * Si existe, actualizar el registro * Retornar mensaje de resultado | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearEnfermedad |
| Descripción | Procedimiento para añadir una nueva enfermedad a tabla Enfermedad. |
| Parámetros de Entrada | enfermedad: 'char',  idTipoEnfermedad: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idTipoEnfermedad. * Si existe, insertar registro en tabla Enfermedad. * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarEnfermedad |
| Descripción | Procedimiento para añadir una nueva enfermedad a tabla Enfermedad. |
| Parámetros de Entrada | idEnfermedad: int,  enfermedad: 'char',  idTipoEnfermedad: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | diagnosticarEnfermedad |
| Descripción | Procedimiento para añadir una nueva enfermedad a tabla Enfermedad diagnosticada a paciente. |
| Parámetros de Entrada | idEnfermedad: int,  idMedico: int,  fechaDiagnostico: 'date',  idExpediente: int,  idConsulta: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | quitarDiagnostico |
| Descripción | Marcar una enfermedad que padecia un paciente como curada |
| Parámetros de Entrada | idEnfermedad: int,  idExpediente: int,  idConsulta: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarTipoExamen |
| Descripción | Modificar valores de un tipo de examen. |
| Parámetros de Entrada | idTipoExamen:'char',  descripcion: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearExamen |
| Descripción | Inserta un nuevo examen en expediente de paciente |
| Parámetros de Entrada | urlDocumento: 'char',  idTipo: int,  idCentroMedico: int,  observacion: 'char',  idExpediente: int,  fecha: 'date', |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * 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. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarExamen |
| Descripción | Inserta un nuevo examen en expediente de paciente |
| Parámetros de Entrada | idExamen: int,  urlDocumento: 'char',  idTipo: int,  idCentroMedico: int,  observacion: 'char',  idExpediente: int,  fecha: 'date' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idTipoExamen, idCentroMedico, * Si existe, actualizar registro en la tabla Examen * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearHojaTrabajoSocial |
| Descripción | Inserta un nuevo registro de HojaTrabajoSocial en expediente de paciente |
| Parámetros de Entrada | descripcion: 'char',  idExpediente: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idExpediente * Si existe, insertar registro en la tabla HojaTrabajoSocial * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarHojaTrabajoSocial |
| Descripción | Actualizar un nuevo registro de HojaTrabajoSocial en expediente de paciente |
| Parámetros de Entrada | idTS:int,  descripcion:'char',  idExpediente:int,  idCentroMedico:int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idExpediente, idTS, idCentroMedico * Si existe, actualizar registro en la tabla HojaTrabajoSocial * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearHospitalizacion |
| Descripción | Insertar un nuevo registro de Hospitalización en expediente de paciente |
| Parámetros de Entrada | observacion: 'char',  fechaHoraIngreso: 'date',  fechaHoraAlta: 'date',  idPiso: int,  cama: 'char',  idMedico: int,  idExpediente: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idPiso, idMedico, idExpendiente * Si existe, insertar registro en la tabla Hospitalizacion * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | darAlta |
| Descripción | Dar de alta un paciente en tabla de Hospitalizacion |
| Parámetros de Entrada | idIngreso: int,  fechaHoraAlta: 'date' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idIngreso * Si existe, actualizar fechaHoraAlta con el valor correspondiente. * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearMedico |
| Descripción | Insertar un nuevo registro en tablaMedico |
| Parámetros de Entrada | pNombre: 'char',  sNombre: 'char',  pApellido: 'char',  sApellido: 'char',  direccion: 'char',  sexo: (F|M),  noIdentidad: 'char',  idPais: int,  idEspecialidad; 'char',  noColegiacion: 'char',  correo: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar validez idPais, idEspecialidad en tablas correspondiente * Comprobar si ya existe Persona con noIdentidad   1. Si existe, almacenar temporalmente el idPersona y saltar a **(A)**   2. 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 correspodientes a la tabla Medico * Devolver mensaje y resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarMedico |
| Descripción | Actualizar registro en tabla médico |
| Parámetros de Entrada | idMedico: int,  direccion: 'char',  idEspecialidad; 'char',  noColegiacion: 'char',  correo: 'char' |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idMedico, idEspecialidad * Si existe, actualizar registro en tabla medico * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | crearPaciente |
| Descripción | Insertar un nuevo registro en tabla Paciente |
| Parámetros de Entrada | pNombre: 'char',  sNombre: 'char',  pApellido: 'char',  sApellido: 'char',  direccion: 'char',  noIdentidad: 'char',  idPais: int,  sexo: (F|M),  correo: 'char',  idTipoSangre: int,  idEscolaridad: int,  idOcupacion: int,  idEstadoCivil: int,  idAscendencia: int |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar validez idPais, idTipoSangre, idEscolaridad, idOcupacion, idEstadoCivil, idAscendencia en tablas correspondiente * Comprobar si ya existe Persona con noIdentidad   1. Si existe, almacenar temporalmente el idPersona y saltar a **(A)**   2. 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 correspodientes a la tabla Paciente * Devolver mensaje y resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarPaciente |
| Descripción | Actualizar registro en tabla Paciente |
| Parámetros de Entrada | idPaciente: int,  direccion: 'char',  correo: 'char',  idEscolaridad: int,  idOcupacion: int,  idEstadoCivil: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idPaciente, idEscolaridad, idOcupacion, idEstadoCivil * Si existe, actualizar registro en tabla Paciente * Retornar mensaje de resultado. | |

|  |  |
| --- | --- |
| 1. **Nombre del Procedimiento/Función** | actualizarPaciente |
| Descripción | Actualizar registro en tabla Paciente |
| Parámetros de Entrada | idPaciente: int,  direccion: 'char',  correo: 'char',  idEscolaridad: int,  idOcupacion: int,  idEstadoCivil: int, |
| Parámetros de Salida | mensaje: 'char',  resultado: bool |
| Descripción del Algoritmo   * Comprobar tipos de datos de entrada. * Comprobar si existe idPaciente, idEscolaridad, idOcupacion, idEstadoCivil * Si existe, actualizar registro en tabla Paciente * Retornar mensaje de resultado. | |