

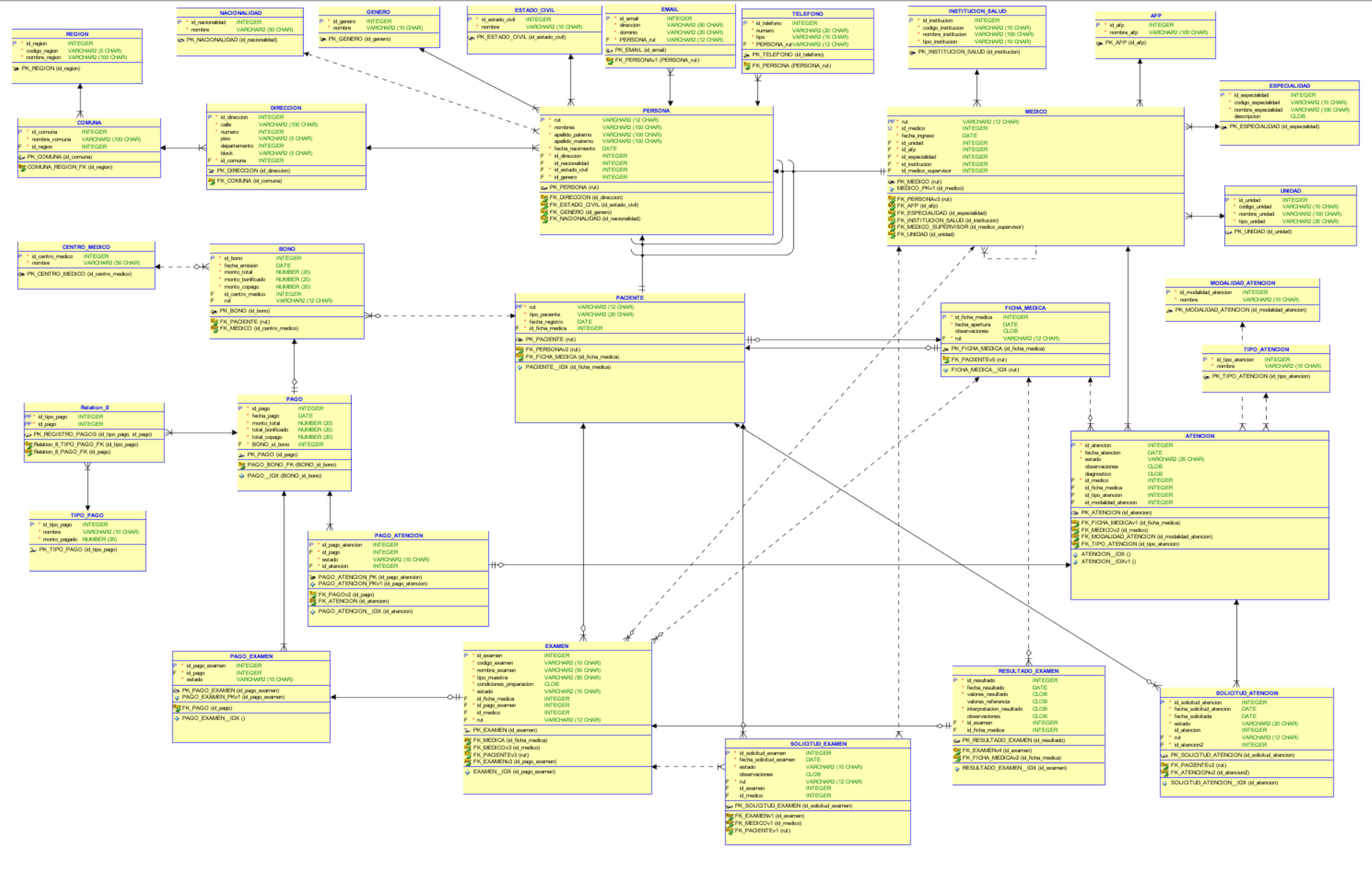
Semana 5

Modelamiento de Bases de Datos

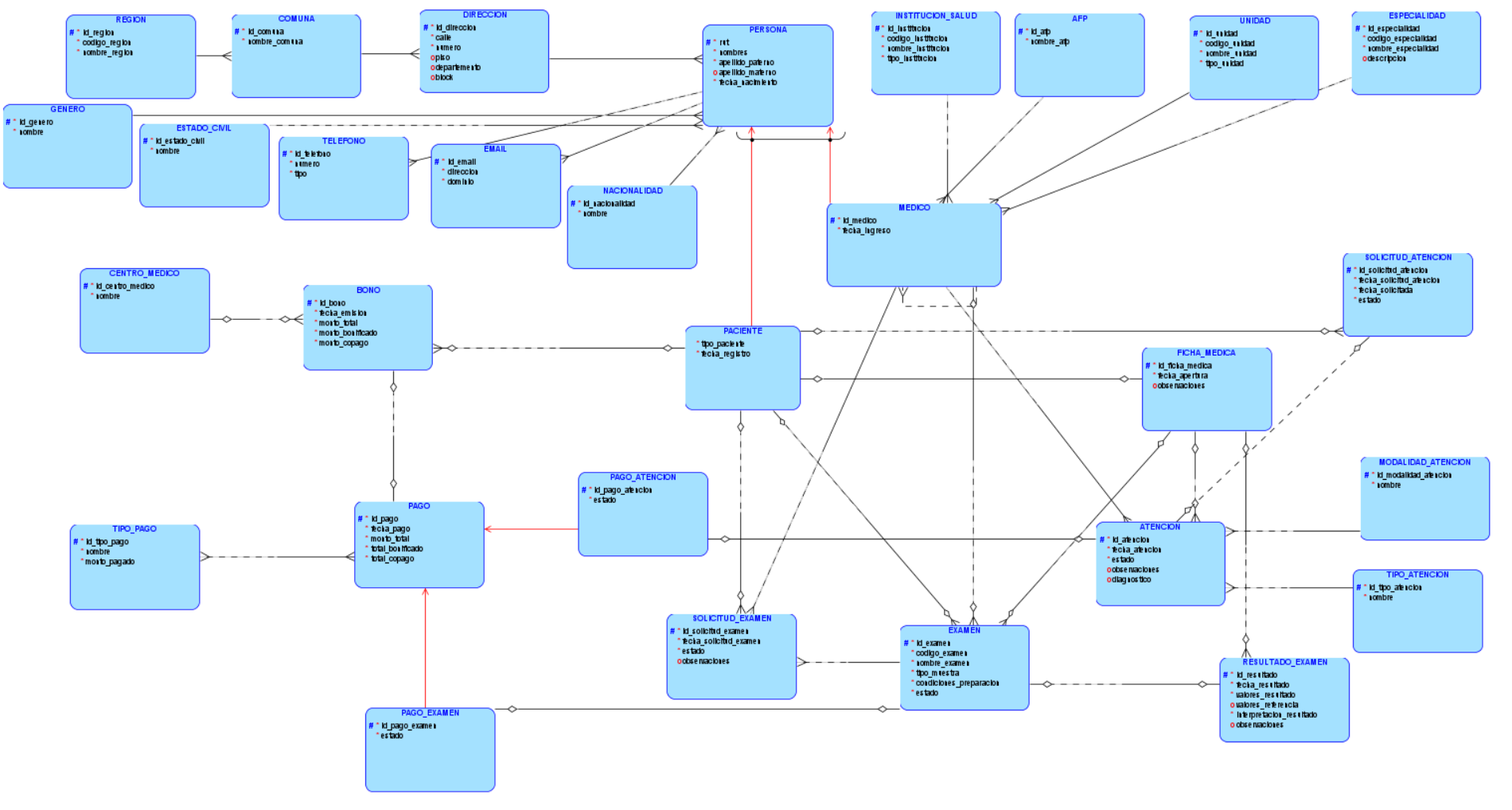
Formato de respuesta

| **Nombre estudiante:** | Nicolás Cavieres |
| --- | --- |
| **Asignatura:** | Modelamiento de Bases de Datos |
| **Profesor:** | Cristian Carreño Villarroel |
| **Fecha:** | 15/09/2025 |

**Modelo Entidad Relación-Extendido Normalizado (MER-E) en notación Barker:**



**Modelo Relacional (MR) Normalizado:**



**Script obtenido con la herramienta, a partir de tu Modelo Relacional**

|  |
| --- |
| -- Generado por Oracle SQL Developer Data Modeler 24.3.1.347.1153  -- en: 2025-09-15 23:47:30 CLST  -- sitio: Oracle Database 11g  -- tipo: Oracle Database 11g  -- predefined type, no DDL - MDSYS.SDO\_GEOMETRY  -- predefined type, no DDL - XMLTYPE  CREATE TABLE afp (  id\_afp INTEGER NOT NULL,  nombre\_afp VARCHAR2(100 CHAR) NOT NULL  );  ALTER TABLE afp ADD CONSTRAINT pk\_afp PRIMARY KEY ( id\_afp );  CREATE TABLE atencion (  id\_atencion INTEGER NOT NULL,  fecha\_atencion DATE NOT NULL,  estado VARCHAR2(20 CHAR) DEFAULT 'pendiente' NOT NULL,  observaciones CLOB,  diagnostico CLOB,  id\_medico INTEGER NOT NULL,  id\_ficha\_medica INTEGER,  id\_tipo\_atencion INTEGER,  id\_modalidad\_atencion INTEGER  );  -- Error - Index ATENCION\_\_IDX has no columns  -- Error - Index ATENCION\_\_IDXv1 has no columns  ALTER TABLE atencion ADD CONSTRAINT pk\_atencion PRIMARY KEY ( id\_atencion );  CREATE TABLE bono (  id\_bono INTEGER NOT NULL,  fecha\_emision DATE NOT NULL,  monto\_total NUMBER(20) NOT NULL,  monto\_bonificado NUMBER(20) NOT NULL,  monto\_copago NUMBER(20) NOT NULL,  id\_centro\_medico INTEGER,  rut VARCHAR2(12 CHAR)  );  ALTER TABLE bono ADD CONSTRAINT pk\_bono PRIMARY KEY ( id\_bono );  CREATE TABLE centro\_medico (  id\_centro\_medico INTEGER NOT NULL,  nombre VARCHAR2(50 CHAR) NOT NULL  );  ALTER TABLE centro\_medico ADD CONSTRAINT pk\_centro\_medico PRIMARY KEY ( id\_centro\_medico );  CREATE TABLE comuna (  id\_comuna INTEGER NOT NULL,  nombre\_comuna VARCHAR2(100 CHAR) NOT NULL,  id\_region INTEGER NOT NULL  );  ALTER TABLE comuna ADD CONSTRAINT pk\_comuna PRIMARY KEY ( id\_comuna );  CREATE TABLE direccion (  id\_direccion INTEGER NOT NULL,  calle VARCHAR2(100 CHAR) NOT NULL,  numero INTEGER NOT NULL,  piso VARCHAR2(5 CHAR),  departemento INTEGER,  block VARCHAR2(5 CHAR),  id\_comuna INTEGER NOT NULL  );  ALTER TABLE direccion ADD CONSTRAINT pk\_direccion PRIMARY KEY ( id\_direccion );  CREATE TABLE email (  id\_email INTEGER NOT NULL,  direccion VARCHAR2(50 CHAR) NOT NULL,  dominio VARCHAR2(20 CHAR) NOT NULL,  persona\_rut VARCHAR2(12 CHAR) NOT NULL  );  ALTER TABLE email ADD CONSTRAINT pk\_email PRIMARY KEY ( id\_email );  CREATE TABLE especialidad (  id\_especialidad INTEGER NOT NULL,  codigo\_especialidad VARCHAR2(10 CHAR) NOT NULL,  nombre\_especialidad VARCHAR2(100 CHAR) NOT NULL,  descripcion CLOB  );  ALTER TABLE especialidad ADD CONSTRAINT pk\_especialidad PRIMARY KEY ( id\_especialidad );  CREATE TABLE estado\_civil (  id\_estado\_civil INTEGER NOT NULL,  nombre VARCHAR2(10 CHAR) NOT NULL  );  ALTER TABLE estado\_civil ADD CONSTRAINT pk\_estado\_civil PRIMARY KEY ( id\_estado\_civil );  CREATE TABLE examen (  id\_examen INTEGER NOT NULL,  codigo\_examen VARCHAR2(10 CHAR) NOT NULL,  nombre\_examen VARCHAR2(50 CHAR) NOT NULL,  tipo\_muestra VARCHAR2(50 CHAR) NOT NULL,  condiciones\_preparacion CLOB NOT NULL,  estado VARCHAR2(10 CHAR) DEFAULT 'pendiente' NOT NULL,  id\_ficha\_medica INTEGER,  id\_pago\_examen INTEGER NOT NULL,  id\_medico INTEGER,  rut VARCHAR2(12 CHAR) NOT NULL  );  CREATE UNIQUE INDEX examen\_\_idx ON  examen (  id\_pago\_examen  ASC );  ALTER TABLE examen ADD CONSTRAINT pk\_examen PRIMARY KEY ( id\_examen );  CREATE TABLE ficha\_medica (  id\_ficha\_medica INTEGER NOT NULL,  fecha\_apertura DATE NOT NULL,  observaciones CLOB,  rut VARCHAR2(12 CHAR) NOT NULL  );  CREATE UNIQUE INDEX ficha\_medica\_\_idx ON  ficha\_medica (  rut  ASC );  ALTER TABLE ficha\_medica ADD CONSTRAINT pk\_ficha\_medica PRIMARY KEY ( id\_ficha\_medica );  CREATE TABLE genero (  id\_genero INTEGER NOT NULL,  nombre VARCHAR2(10 CHAR) NOT NULL  );  ALTER TABLE genero ADD CONSTRAINT pk\_genero PRIMARY KEY ( id\_genero );  CREATE TABLE institucion\_salud (  id\_institucion INTEGER NOT NULL,  codigo\_institucion VARCHAR2(10 CHAR) NOT NULL,  nombre\_institucion VARCHAR2(100 CHAR) NOT NULL,  tipo\_institucion VARCHAR2(10 CHAR) NOT NULL  );  ALTER TABLE institucion\_salud ADD CONSTRAINT pk\_institucion\_salud PRIMARY KEY ( id\_institucion );  CREATE TABLE medico (  rut VARCHAR2(12 CHAR) NOT NULL,  id\_medico INTEGER NOT NULL,  fecha\_ingreso DATE NOT NULL,  id\_unidad INTEGER NOT NULL,  id\_afp INTEGER NOT NULL,  id\_especialidad INTEGER NOT NULL,  id\_institucion INTEGER NOT NULL,  id\_medico\_supervisor INTEGER  );  ALTER TABLE medico ADD CONSTRAINT pk\_medico PRIMARY KEY ( rut );  ALTER TABLE medico ADD CONSTRAINT medico\_pkv1 UNIQUE ( id\_medico );  CREATE TABLE modalidad\_atencion (  id\_modalidad\_atencion INTEGER NOT NULL,  nombre VARCHAR2(10 CHAR) NOT NULL  );  ALTER TABLE modalidad\_atencion ADD CONSTRAINT pk\_modalidad\_atencion PRIMARY KEY ( id\_modalidad\_atencion );  CREATE TABLE nacionalidad (  id\_nacionalidad INTEGER NOT NULL,  nombre VARCHAR2(50 CHAR) NOT NULL  );  ALTER TABLE nacionalidad ADD CONSTRAINT pk\_nacionalidad PRIMARY KEY ( id\_nacionalidad );  CREATE TABLE paciente (  rut VARCHAR2(12 CHAR) NOT NULL,  tipo\_paciente VARCHAR2(20 CHAR) NOT NULL,  fecha\_registro DATE NOT NULL,  id\_ficha\_medica INTEGER NOT NULL  );  CREATE UNIQUE INDEX paciente\_\_idx ON  paciente (  id\_ficha\_medica  ASC );  ALTER TABLE paciente ADD CONSTRAINT pk\_paciente PRIMARY KEY ( rut );  CREATE TABLE pago (  id\_pago INTEGER NOT NULL,  fecha\_pago DATE NOT NULL,  monto\_total NUMBER(20) NOT NULL,  total\_bonificado NUMBER(20) NOT NULL,  total\_copago NUMBER(20) NOT NULL,  bono\_id\_bono INTEGER NOT NULL  );  CREATE UNIQUE INDEX pago\_\_idx ON  pago (  bono\_id\_bono  ASC );  ALTER TABLE pago ADD CONSTRAINT pk\_pago PRIMARY KEY ( id\_pago );  CREATE TABLE pago\_atencion (  id\_pago\_atencion INTEGER NOT NULL,  id\_pago INTEGER NOT NULL,  estado VARCHAR2(10 CHAR) DEFAULT 'pendiente' NOT NULL,  id\_atencion INTEGER NOT NULL  );  CREATE UNIQUE INDEX pago\_atencion\_\_idx ON  pago\_atencion (  id\_atencion  ASC );  ALTER TABLE pago\_atencion ADD CONSTRAINT pago\_atencion\_pk PRIMARY KEY ( id\_pago\_atencion );  ALTER TABLE pago\_atencion ADD CONSTRAINT pago\_atencion\_pkv1 UNIQUE ( id\_pago\_atencion );  CREATE TABLE pago\_examen (  id\_pago\_examen INTEGER NOT NULL,  id\_pago INTEGER NOT NULL,  estado VARCHAR2(10 CHAR) DEFAULT 'pendiente' NOT NULL,  id\_examen INTEGER NOT NULL  );  CREATE UNIQUE INDEX pago\_examen\_\_idx ON  pago\_examen (  id\_examen  ASC );  ALTER TABLE pago\_examen ADD CONSTRAINT pk\_pago\_examen PRIMARY KEY ( id\_pago\_examen );  ALTER TABLE pago\_examen ADD CONSTRAINT pago\_examen\_pkv1 UNIQUE ( id\_pago\_examen );  CREATE TABLE persona (  rut VARCHAR2(12 CHAR) NOT NULL,  nombres VARCHAR2(100 CHAR) NOT NULL,  apellido\_paterno VARCHAR2(100 CHAR) NOT NULL,  apellido\_materno VARCHAR2(100 CHAR),  fecha\_nacimiento DATE NOT NULL,  id\_direccion INTEGER NOT NULL,  id\_nacionalidad INTEGER,  id\_estado\_civil INTEGER NOT NULL,  id\_genero INTEGER NOT NULL  );  ALTER TABLE persona ADD CONSTRAINT pk\_persona PRIMARY KEY ( rut );  CREATE TABLE region (  id\_region INTEGER NOT NULL,  codigo\_region VARCHAR2(5 CHAR) NOT NULL,  nombre\_region VARCHAR2(100 CHAR) NOT NULL  );  ALTER TABLE region ADD CONSTRAINT pk\_region PRIMARY KEY ( id\_region );  CREATE TABLE relation\_8 (  id\_tipo\_pago INTEGER NOT NULL,  id\_pago INTEGER NOT NULL  );  ALTER TABLE relation\_8 ADD CONSTRAINT pk\_registro\_pagos PRIMARY KEY ( id\_tipo\_pago,  id\_pago );  CREATE TABLE resultado\_examen (  id\_resultado INTEGER NOT NULL,  fecha\_resultado DATE NOT NULL,  valores\_resultado CLOB NOT NULL,  valores\_referencia CLOB,  interpretacion\_resultado CLOB NOT NULL,  observaciones CLOB,  id\_examen INTEGER NOT NULL,  id\_ficha\_medica INTEGER  );  CREATE UNIQUE INDEX resultado\_examen\_\_idx ON  resultado\_examen (  id\_examen  ASC );  ALTER TABLE resultado\_examen ADD CONSTRAINT pk\_resultado\_examen PRIMARY KEY ( id\_resultado );  CREATE TABLE solicitud\_atencion (  id\_solicitud\_atencion INTEGER NOT NULL,  fecha\_solicitud\_atencion DATE NOT NULL,  fecha\_solicitada DATE NOT NULL,  estado VARCHAR2(20 CHAR) DEFAULT 'pendiente' NOT NULL,  id\_atencion INTEGER,  rut VARCHAR2(12 CHAR) NOT NULL,  id\_atencion2 INTEGER NOT NULL  );  COMMENT ON COLUMN solicitud\_atencion.fecha\_solicitud\_atencion IS  'fecha y hora de solicitud de atención';  CREATE UNIQUE INDEX solicitud\_atencion\_\_idx ON  solicitud\_atencion (  id\_atencion  ASC );  ALTER TABLE solicitud\_atencion ADD CONSTRAINT pk\_solicitud\_atencion PRIMARY KEY ( id\_solicitud\_atencion );  CREATE TABLE solicitud\_examen (  id\_solicitud\_examen INTEGER NOT NULL,  fecha\_solicitud\_examen DATE NOT NULL,  estado VARCHAR2(10 CHAR) DEFAULT 'pendiente' NOT NULL,  observaciones CLOB,  rut VARCHAR2(12 CHAR) NOT NULL,  id\_examen INTEGER,  id\_medico INTEGER  );  COMMENT ON COLUMN solicitud\_examen.fecha\_solicitud\_examen IS  'fecha y hora de la solicitud';  ALTER TABLE solicitud\_examen ADD CONSTRAINT pk\_solicitud\_examen PRIMARY KEY ( id\_solicitud\_examen );  CREATE TABLE telefono (  id\_telefono INTEGER NOT NULL,  numero VARCHAR2(20 CHAR) NOT NULL,  tipo VARCHAR2(10 CHAR) NOT NULL,  persona\_rut VARCHAR2(12 CHAR) NOT NULL  );  ALTER TABLE telefono ADD CONSTRAINT pk\_telefono PRIMARY KEY ( id\_telefono );  CREATE TABLE tipo\_atencion (  id\_tipo\_atencion INTEGER NOT NULL,  nombre VARCHAR2(10 CHAR) NOT NULL  );  ALTER TABLE tipo\_atencion ADD CONSTRAINT pk\_tipo\_atencion PRIMARY KEY ( id\_tipo\_atencion );  CREATE TABLE tipo\_pago (  id\_tipo\_pago INTEGER NOT NULL,  nombre VARCHAR2(10 CHAR) NOT NULL,  monto\_pagado NUMBER(20) NOT NULL  );  ALTER TABLE tipo\_pago ADD CONSTRAINT pk\_tipo\_pago PRIMARY KEY ( id\_tipo\_pago );  CREATE TABLE unidad (  id\_unidad INTEGER NOT NULL,  codigo\_unidad VARCHAR2(10 CHAR) NOT NULL,  nombre\_unidad VARCHAR2(100 CHAR) NOT NULL,  tipo\_unidad VARCHAR2(20 CHAR) NOT NULL  );  ALTER TABLE unidad ADD CONSTRAINT pk\_unidad PRIMARY KEY ( id\_unidad );  ALTER TABLE comuna  ADD CONSTRAINT comuna\_region\_fk FOREIGN KEY ( id\_region )  REFERENCES region ( id\_region );  ALTER TABLE medico  ADD CONSTRAINT fk\_afp FOREIGN KEY ( id\_afp )  REFERENCES afp ( id\_afp );  ALTER TABLE pago\_atencion  ADD CONSTRAINT fk\_atencion FOREIGN KEY ( id\_atencion )  REFERENCES atencion ( id\_atencion );  ALTER TABLE solicitud\_atencion  ADD CONSTRAINT fk\_atencionv2 FOREIGN KEY ( id\_atencion2 )  REFERENCES atencion ( id\_atencion );  ALTER TABLE direccion  ADD CONSTRAINT fk\_comuna FOREIGN KEY ( id\_comuna )  REFERENCES comuna ( id\_comuna );  ALTER TABLE persona  ADD CONSTRAINT fk\_direccion FOREIGN KEY ( id\_direccion )  REFERENCES direccion ( id\_direccion );  ALTER TABLE medico  ADD CONSTRAINT fk\_especialidad FOREIGN KEY ( id\_especialidad )  REFERENCES especialidad ( id\_especialidad );  ALTER TABLE persona  ADD CONSTRAINT fk\_estado\_civil FOREIGN KEY ( id\_estado\_civil )  REFERENCES estado\_civil ( id\_estado\_civil );  ALTER TABLE pago\_examen  ADD CONSTRAINT fk\_examen FOREIGN KEY ( id\_examen )  REFERENCES examen ( id\_examen );  ALTER TABLE solicitud\_examen  ADD CONSTRAINT fk\_examenv1 FOREIGN KEY ( id\_examen )  REFERENCES examen ( id\_examen );  ALTER TABLE examen  ADD CONSTRAINT fk\_examenv3 FOREIGN KEY ( id\_pago\_examen )  REFERENCES pago\_examen ( id\_pago\_examen );  ALTER TABLE resultado\_examen  ADD CONSTRAINT fk\_examenv4 FOREIGN KEY ( id\_examen )  REFERENCES examen ( id\_examen );  ALTER TABLE paciente  ADD CONSTRAINT fk\_ficha\_medica FOREIGN KEY ( id\_ficha\_medica )  REFERENCES ficha\_medica ( id\_ficha\_medica );  ALTER TABLE atencion  ADD CONSTRAINT fk\_ficha\_medicav1 FOREIGN KEY ( id\_ficha\_medica )  REFERENCES ficha\_medica ( id\_ficha\_medica );  ALTER TABLE resultado\_examen  ADD CONSTRAINT fk\_ficha\_medicav2 FOREIGN KEY ( id\_ficha\_medica )  REFERENCES ficha\_medica ( id\_ficha\_medica );  ALTER TABLE persona  ADD CONSTRAINT fk\_genero FOREIGN KEY ( id\_genero )  REFERENCES genero ( id\_genero );  ALTER TABLE medico  ADD CONSTRAINT fk\_institucion\_salud FOREIGN KEY ( id\_institucion )  REFERENCES institucion\_salud ( id\_institucion );  ALTER TABLE examen  ADD CONSTRAINT fk\_medica FOREIGN KEY ( id\_ficha\_medica )  REFERENCES ficha\_medica ( id\_ficha\_medica );  ALTER TABLE bono  ADD CONSTRAINT fk\_medico FOREIGN KEY ( id\_centro\_medico )  REFERENCES centro\_medico ( id\_centro\_medico );  ALTER TABLE medico  ADD CONSTRAINT fk\_medico\_supervisor FOREIGN KEY ( id\_medico\_supervisor )  REFERENCES medico ( id\_medico );  ALTER TABLE solicitud\_examen  ADD CONSTRAINT fk\_medicov1 FOREIGN KEY ( id\_medico )  REFERENCES medico ( id\_medico );  ALTER TABLE atencion  ADD CONSTRAINT fk\_medicov2 FOREIGN KEY ( id\_medico )  REFERENCES medico ( id\_medico );  ALTER TABLE examen  ADD CONSTRAINT fk\_medicov3 FOREIGN KEY ( id\_medico )  REFERENCES medico ( id\_medico );  ALTER TABLE atencion  ADD CONSTRAINT fk\_modalidad\_atencion FOREIGN KEY ( id\_modalidad\_atencion )  REFERENCES modalidad\_atencion ( id\_modalidad\_atencion );  ALTER TABLE persona  ADD CONSTRAINT fk\_nacionalidad FOREIGN KEY ( id\_nacionalidad )  REFERENCES nacionalidad ( id\_nacionalidad );  ALTER TABLE bono  ADD CONSTRAINT fk\_paciente FOREIGN KEY ( rut )  REFERENCES paciente ( rut );  ALTER TABLE solicitud\_examen  ADD CONSTRAINT fk\_pacientev1 FOREIGN KEY ( rut )  REFERENCES paciente ( rut );  ALTER TABLE solicitud\_atencion  ADD CONSTRAINT fk\_pacientev2 FOREIGN KEY ( rut )  REFERENCES paciente ( rut );  ALTER TABLE examen  ADD CONSTRAINT fk\_pacientev3 FOREIGN KEY ( rut )  REFERENCES paciente ( rut );  ALTER TABLE ficha\_medica  ADD CONSTRAINT fk\_pacientev5 FOREIGN KEY ( rut )  REFERENCES paciente ( rut );  ALTER TABLE pago\_examen  ADD CONSTRAINT fk\_pago FOREIGN KEY ( id\_pago )  REFERENCES pago ( id\_pago );  ALTER TABLE pago\_atencion  ADD CONSTRAINT fk\_pagov2 FOREIGN KEY ( id\_pago )  REFERENCES pago ( id\_pago );  ALTER TABLE telefono  ADD CONSTRAINT fk\_persona FOREIGN KEY ( persona\_rut )  REFERENCES persona ( rut );  ALTER TABLE email  ADD CONSTRAINT fk\_personav1 FOREIGN KEY ( persona\_rut )  REFERENCES persona ( rut );  ALTER TABLE paciente  ADD CONSTRAINT fk\_personav2 FOREIGN KEY ( rut )  REFERENCES persona ( rut );  ALTER TABLE medico  ADD CONSTRAINT fk\_personav3 FOREIGN KEY ( rut )  REFERENCES persona ( rut );  ALTER TABLE atencion  ADD CONSTRAINT fk\_tipo\_atencion FOREIGN KEY ( id\_tipo\_atencion )  REFERENCES tipo\_atencion ( id\_tipo\_atencion );  ALTER TABLE medico  ADD CONSTRAINT fk\_unidad FOREIGN KEY ( id\_unidad )  REFERENCES unidad ( id\_unidad );  ALTER TABLE pago  ADD CONSTRAINT pago\_bono\_fk FOREIGN KEY ( bono\_id\_bono )  REFERENCES bono ( id\_bono );  ALTER TABLE relation\_8  ADD CONSTRAINT relation\_8\_pago\_fk FOREIGN KEY ( id\_pago )  REFERENCES pago ( id\_pago );  ALTER TABLE relation\_8  ADD CONSTRAINT relation\_8\_tipo\_pago\_fk FOREIGN KEY ( id\_tipo\_pago )  REFERENCES tipo\_pago ( id\_tipo\_pago );  CREATE OR REPLACE TRIGGER fknto\_atencion BEFORE  UPDATE OF id\_ficha\_medica ON atencion  FOR EACH ROW  BEGIN  IF :old.id\_ficha\_medica IS NOT NULL THEN  raise\_application\_error(-20225, 'Non Transferable FK constraint FK\_FICHA\_MEDICAv1 on table ATENCION is violated');  END IF;  END;  /  CREATE OR REPLACE TRIGGER fknto\_bono BEFORE  UPDATE OF rut, id\_centro\_medico ON bono  FOR EACH ROW  BEGIN  IF :old.rut IS NOT NULL THEN  raise\_application\_error(-20225, 'Non Transferable FK constraint FK\_PACIENTE on table BONO is violated');  END IF;  IF :old.id\_centro\_medico IS NOT NULL THEN  raise\_application\_error(-20225, 'Non Transferable FK constraint FK\_MEDICO on table BONO is violated');  END IF;  END;  /  CREATE OR REPLACE TRIGGER fknto\_examen BEFORE  UPDATE OF id\_medico, id\_ficha\_medica ON examen  FOR EACH ROW  BEGIN  IF :old.id\_medico IS NOT NULL THEN  raise\_application\_error(-20225, 'Non Transferable FK constraint FK\_MEDICOv3 on table EXAMEN is violated');  END IF;  IF :old.id\_ficha\_medica IS NOT NULL THEN  raise\_application\_error(-20225, 'Non Transferable FK constraint FK\_MEDICA on table EXAMEN is violated');  END IF;  END;  /  CREATE OR REPLACE TRIGGER fkntm\_examen BEFORE  UPDATE OF rut, id\_pago\_examen ON examen  BEGIN  raise\_application\_error(-20225, 'Non Transferable FK constraint on table EXAMEN is violated');  END;  /  CREATE OR REPLACE TRIGGER fkntm\_ficha\_medica BEFORE  UPDATE OF rut ON ficha\_medica  BEGIN  raise\_application\_error(-20225, 'Non Transferable FK constraint on table FICHA\_MEDICA is violated');  END;  /  CREATE OR REPLACE TRIGGER fkntm\_paciente BEFORE  UPDATE OF id\_ficha\_medica ON paciente  BEGIN  raise\_application\_error(-20225, 'Non Transferable FK constraint on table PACIENTE is violated');  END;  /  CREATE OR REPLACE TRIGGER fkntm\_pago BEFORE  UPDATE OF bono\_id\_bono ON pago  BEGIN  raise\_application\_error(-20225, 'Non Transferable FK constraint on table PAGO is violated');  END;  /  CREATE OR REPLACE TRIGGER fkntm\_pago\_atencion BEFORE  UPDATE OF id\_atencion ON pago\_atencion  BEGIN  raise\_application\_error(-20225, 'Non Transferable FK constraint on table PAGO\_ATENCION is violated');  END;  /  -- Informe de Resumen de Oracle SQL Developer Data Modeler:  --  -- CREATE TABLE 30  -- CREATE INDEX 8  -- ALTER TABLE 74  -- CREATE VIEW 0  -- ALTER VIEW 0  -- CREATE PACKAGE 0  -- CREATE PACKAGE BODY 0  -- CREATE PROCEDURE 0  -- CREATE FUNCTION 0  -- CREATE TRIGGER 13  -- ALTER TRIGGER 0  -- CREATE COLLECTION TYPE 0  -- CREATE STRUCTURED TYPE 0  -- CREATE STRUCTURED TYPE BODY 0  -- CREATE CLUSTER 0  -- CREATE CONTEXT 0  -- CREATE DATABASE 0  -- CREATE DIMENSION 0  -- CREATE DIRECTORY 0  -- CREATE DISK GROUP 0  -- CREATE ROLE 0  -- CREATE ROLLBACK SEGMENT 0  -- CREATE SEQUENCE 0  -- CREATE MATERIALIZED VIEW 0  -- CREATE MATERIALIZED VIEW LOG 0  -- CREATE SYNONYM 0  -- CREATE TABLESPACE 0  -- CREATE USER 0  --  -- DROP TABLESPACE 0  -- DROP DATABASE 0  --  -- REDACTION POLICY 0  --  -- ORDS DROP SCHEMA 0  -- ORDS ENABLE SCHEMA 0  -- ORDS ENABLE OBJECT 0  --  -- ERRORS 0  -- WARNINGS 0 |



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.