



USAC

TRICENTENARIA

Universidad de San Carlos de Guatemala

MANUAL TÉCNICO:

Proyecto: Practica No. 2

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1 SECCIÓN "A"

Universidad De San Carlos De Guatemala
Centro Universitario De Occidente
División De Ciencias De La Ingeniería



Hospital

Manual Técnico

Astrid Gabriela Martínez Castillo

Carne: 201731318

INDICE

1. INTRODUCCION 1

2. OBJETIVOS 2

2.1 Objetivos Específicos 2

2.2 Objetivos Generales 2

3. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 2

4. REQUERIMIENTOS TECNICOS 2

5. ANÁLISIS DE LOS REQUERIMIENTOS 3

7. DIAGRAMAS DE CLASES 4-9

INTRODUCCION

Este documento describe cada una de las herramientas que se utilizaron para el desarrollo de un programa orientado a un hospital para el registro de movimientos y empleados, destacando tanto aspectos relacionados con el hardware como el software esperando sirva de referencia para especificar la creación de la práctica, así como también los la documentación necesaria.

2. OBJETIVOS

2.1. Objetivos Específicos

- Construcción de algoritmos para los requerimientos de la actividad.
- Ampliar el conocimiento de estructuras del control de JAVA.
- Ampliar el conocimiento de procedimientos en JAVA.
- Implementación desarrollo web

2.2. Objetivos Generales

- Familiarizar al estudiante con el lenguaje JAVA.
- Aplicar conceptos recibidos en clase magistral y laboratorio.
- Elaborar la lógica para la solución del problema planteado.

4.1. Requerimientos Mínimos de Software y Hardware

- **Windows:** Windows 10 (8u51 y superiores) Windows 8.x (escritorio) Windows 7 SP1 Windows Vista SP2 Windows Server 2008 R2 SP1 (64 bits) Windows Server 2012 y 2012 R2 (64 bits).
- **Memoria RAM:** 4 GB.
- **Espacio en disco:** 124 MB para JRE; 2 MB para Java Update.
- **Procesador Mínimo:** Pentium 2 a 266 MHz
- **Linux**
 - Oracle Linux 5.5+1
 - Oracle Linux 6.x (32 bits), 6.x (64 bits)2
 - Oracle Linux 7.x (64 bits)2 (8u20 y superiores)
 - Ubuntu Linux 12.04 LTS, 13.x
 - Ubuntu Linux 14.x (8u25 y superiores)
 - Ubuntu Linux 15.04 (8u45 y superiores)
 - Ubuntu Linux 15.10 (8u65 y superiores)
 - Exploradores: Firefox

3. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

- Java JDK. 8
- Netbeans 8.2
- Linux Ubuntu 18.04
- **JasperRepors**

4. CONFIGURACIÓN

No hay una configuración general la aplicación en si viene configurada y la utilización es intuitiva para el usuario o programador que quiera manejar el software.

5. ANÁLISIS DE LOS REQUERIMIENTOS

- **IDENTIFICACIÓN DE PROBLEMAS, OPORTUNIDADES Y**

- **OBJETIVOS.**

Sistema de control de empleados y clientes

- **DETERMINACIÓN DE LOS REQUERIMIENTOS DE INFORMACIÓN.**

Un hospital privado necesita del manejo de clientes, medicinas, empleados entre otros es por esto que surge la necesidad de desarrollar un software adecuado para el uso.

El sistema permitira el registro de clientes, registro y despido de empleados, registro de medicina entre otras cosas.

6. MAPEO FÍSICO DE LA BASE DE DATOS

```
CREATE DATABASE hospitalPractica;  
USE hospitalPractica;
```

```
CREATE TABLE AreasHospital(  
    nombreArea CHAR(23) NOT NULL,  
    idAreaHospital INTEGER NOT NULL AUTO_INCREMENT,  
    CONSTRAINT PK_ID_AREA PRIMARY KEY(idAreasHospital)  
);
```

```
CREATE TABLE Empleado(  
    nombres CHAR(23) NOT NULL,  
    apellidos CHAR(23) NOT NULL,  
    cuiEmpleado CHAR(13) NOT NULL,  
    salario INTEGER() NOT NULL,  
    rango CHAR(23) NOT NULL,  
    CONSTRAINT PK_CUI_EMPLEADO PRIMARY KEY(cuiEmpleado)  
);
```

```
CREATE TABLE Contratar(  
    fechaInicio DATE,  
    fechaFinal Date default NULL,  
    cuiEmpleado CHAR(13) NOT NULL,  
    rango CHAR(23) NOT NULL,  
    nombreArea INTEGER NOT NULL,  
    IGSS BOOLEAN DEFAULT TRUE,  
    IRTRA boolean DEFAULT TRUE,  
    CONSTRAINT FK_CUI_CEMPLADO FOREIGN KEY(cuiEmpleado)  
        REFERENCES Empleado(cuiEmpleado),  
    CONSTRAINT FK_AREA_HOSPITAL FOREIGN KEY(nombreArea)  
        REFERENCES AreasHospital(nombreArea)  
);
```

```
CREATE TABLE DetalleContrato(  
    diasVacaciones INTEGER NOT NULL,  
    IGSS FLOAT NOT NULL,  
    IRTRA FLOAT NOT NULL  
);
```

```
CREATE TABLE rango(  
    nombreArea INTEGER NOT NULL,  
    rango CHAR(23) NOT NULL,  
    sueldo INTEGER NOT NULL,  
    CONSTRAINT PK_RANGO PRIMARY KEY(rango),  
    CONSTRAINT FK_AREA_HOSPITALR FOREIGN KEY(nombreArea)  
        REFERENCES AreasHospital(nombreArea)  
);
```

```
CREATE TABLE VACACIONES(  
    cuiEmpleado CHAR(13) NOT NULL,  
    reclamadas BOOLEAN NOT NULL DEFAULT FALSE,  
    fechaInicioVacaciones DATE NOT NULL,  
    fechaFinalVacaciones DATE NOT NULL,  
    fechaAsignacion DATE NOT NULL,  
    CONSTRAINT FK_CUI_CEMPLADO FOREIGN KEY(cuiEmpleado)  
        REFERENCES Empleado(cuiEmpleado)  
);
```

```
CREATE TABLE Aumento(  
    cuiEmpleado CHAR(13) NOT NULL,  
    aumentoSalario INTEGER NOT NULL,  
    fechaAumento DATE NOT NULL,  
    CONSTRAINT FK_CUI_CEMPLADO FOREIGN KEY(cuiEmpleado)  
        REFERENCES Empleado(cuiEmpleado)  
);
```

```
CREATE TABLE Cobrar(  
    cuiEmpleado CHAR(13) NOT NULL,  
    rango CHAR(23) NOT NULL,  
    fechaPago DATE NOT NULL,  
    CONSTRAINT FK_CUI_CEMPLADO FOREIGN KEY(cuiEmpleado)  
        REFERENCES Empleado(cuiEmpleado),  
    CONSTRAINT FK_RANGO FOREIGN KEY(rango)  
        REFERENCES Salario(rango)  
);
```

```
CREATE TABLE Cliente(  
    nombres VARCHAR(23) NOT NULL,  
    apellidos VARCHAR(30) NOT NULL,  
    cuiCliente VARCHAR(30) NOT NULL,  
    telefono CHAR(13) ,
```



```
        correoElectronico VARCHAR(50),  
        CONSTRAINT PK_CUI_CLIENTE PRIMARY KEY(cuiCliente)  
    );
```

```
CREATE TABLE Factura(  
    idFactura INTEGER NOT NULL AUTO_INCREMENT,  
    cuiCliente VARCHAR(30) NOT NULL,  
    idArea INTEGER NOT NULL,  
    monto FLOAT NOT NULL,  
    fecha DATE NOT NULL,  
    CONSTRAINT PK_ID_FACTURA PRIMARY KEY(idFactura),  
    CONSTRAINT FK_CUI_CLIENTE FOREIGN KEY(cuiCliente)  
        REFERENCES Cliente(cuiCliente)  
);
```

```
CREATE TABLE Medicina(  
    nombreProducto VARCHAR(60) NOT NULL,  
    descripcion VARCHAR(100) NOT NULL,  
    precio FLOAT NOT NULL,  
    costo FLOAT NOT NULL,  
    existenciaMinima INTEGER NOT NULL,  
    CONSTRAINT PK_NOMBRE_PRODUCTO PRIMARY KEY(nombreProducto)  
);
```

```
CREATE TABLE Verder(  
    idFactura INTEGER NOT NULL,  
    nombreProducto VARCHAR(60) NOT NULL,  
    cuiVendedor CHAR(13) NOT NULL,  
    fecha DATE NOT NULL,  
    cantidad INTEGER NOT NULL,  
    CONSTRAINT FK_ID_FACTURA FOREIGN KEY(idFactura)  
        REFERENCES Factura(idFactura),  
    CONSTRAINT FK_NOMBRE_PRODUCTO FOREIGN KEY(nombreProducto)  
        REFERENCES Medicamento(nombreProducto)  
    CONSTRAINT FK_CUI_VENDEDOR FOREIGN KEY(cuiVendedor)  
        REFERENCES Empleado(cuiVendedor)  
);
```

```
CREATE TABLE InventarioFarmacia (  
    nombreProducto VARCHAR(60) NOT NULL,  
    existencia INTEGER NOT NULL,  
    CONSTRAINT FK_NOMBRE_PROD FOREIGN KEY(nombreProducto)
```

```

REFERENCES Medicina(nombreProducto)
);

CREATE TABLE Atender(
    cuiCliente VARCHAR(30) NOT NULL,
    cuiEmpleado CHAR(13) NOT NULL,
    rango CHAR(23) NOT NULL,
    fechaAsignacion DATE NOT NULL,
    CONSTRAINT FK_CUI_CLIENTE FOREIGN KEY(cuiCliente)
        REFERENCES Cliente(cuiCliente),
    CONSTRAINT FK_CUI_CEMPLADO FOREIGN KEY(cuiEmpleado)
        REFERENCES Empleado(cuiEmpleado)
);

CREATE TABLE Habitacion(
    NoHabitacion INTEGER NOT NULL,
    costoMantenimiento FLOAT NOT NULL,
    disponibilidad BOOLEAN NOT NULL,
    CONSTRAINT PK_NO_HABITACION PRIMARY KEY(NoHabitacion)
);

CREATE TABLE PacienteInternado(
    cuiCliente VARCHAR(30) NOT NULL,
    fechaIngreso DATE NOT NULL,
    fechaEgreso DATE,
    NoHabitacion INTEGER NOT NULL,
    CONSTRAINT FK_CUI_CLIENTEI FOREIGN KEY(cuiCliente)
        REFERENCES Cliente(cuiCliente),
    CONSTRAINT FK_NO_HABITACIONP FOREIGN KEY(NoHabitacion)
        REFERENCES Habitacion(NoHabitacion)
);

CREATE TABLE ServiciosHospital(
    idAreaHospital INTEGER NOT NULL,
    nombreServicio VARCHAR(100) NOT NULL,
    precioServicio INTEGER NOT NULL,
    CONSTRAINT PK_NOMBRE_SERVICIO PRIMARY KEY(nombreServicio)
);

CREATE TABLE ServicioEspecial(
    nombreServicioEspecial VARCHAR(100) NOT NULL,
    precioServicio INTEGER NOT NULL,

```

```

        CONSTRAINT PK_NOMBRE_SERVICIO_ESPECIAL PRIMARY
        KEY(nombreServicioEspecial)
    );

CREATE TABLE MedicoEspecialista(
    nombreServicioEspecial VARCHAR(100) NOT NULL,
    nombre VARCHAR(40) NOT NULL,
    apellido VARCHAR(40) NOT NULL,
    cuiEspecialista CHAR(13) NOT NULL,
    pago INTEGER NOT NULL,
    CONSTRAINT PK_CUI_ESPECIALISTA PRIMARY KEY(cuiEspecialista),
    CONSTRAINT FK_NOMBRE_SERVICIO_ESPECIAL KEY(nombreServicioEspecial)
        REFERENCES ServicioEspecial(nombreServicioEspecial)
);

CREATE TABLE Adquirir(
    nombreServicio VARCHAR(100) NOT NULL,
    cuiCliente VARCHAR(100) NOT NULL,
    fechaServicio DATETIME NOT NULL,
    estadoPago BOOLEAN NOT NULL DEFAULT FALSE,
    CONSTRAINT FK_CUI_CLIENTE FOREIGN KEY(cuiCliente)
        REFERENCES Cliente(cuiCliente),
    CONSTRAINT FK_NOMBRE_SERVICIO FOREIGN KEY(nombreServicio)
        REFERENCES ServiciosHospital(nombreServicio)
);

CREATE TABLE Pagar(
    nombreServicio VARCHAR(100) NOT NULL,
    cuiCliente VARCHAR(100) NOT NULL,
    idFactura INTEGER NOT NULL,
    fechaPago DATE NOT NULL,
    CONSTRAINT FK_ID_FACTURA FOREIGN KEY(idFactura)
        REFERENCES Factura(idFactura),
    CONSTRAINT FK_CUI_CLIENTE FOREIGN KEY(cuiCliente)
        REFERENCES Cliente(cuiCliente)
);

```


DIAGRAMA DE CLASES

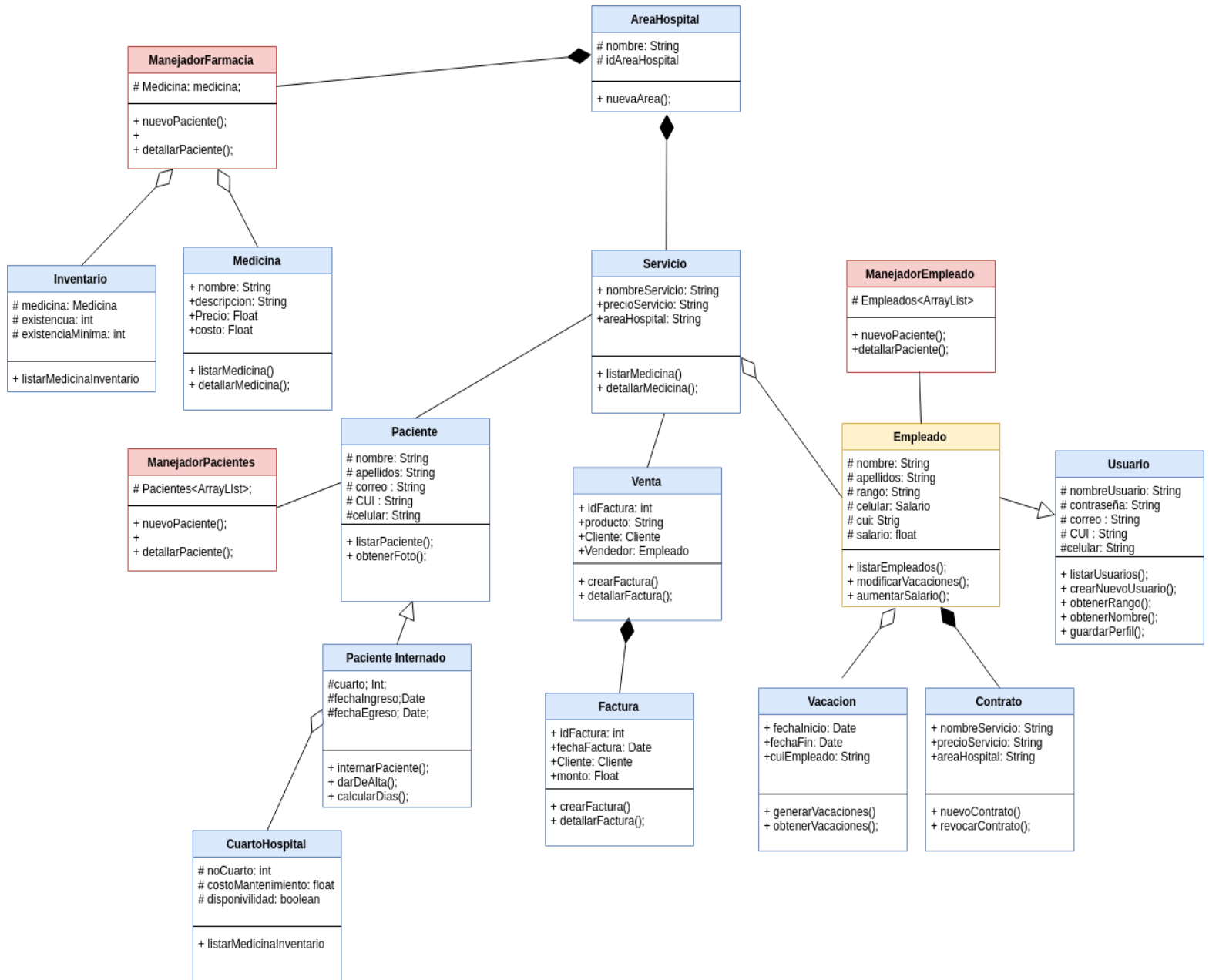


DIAGRAMA DE TABLAS

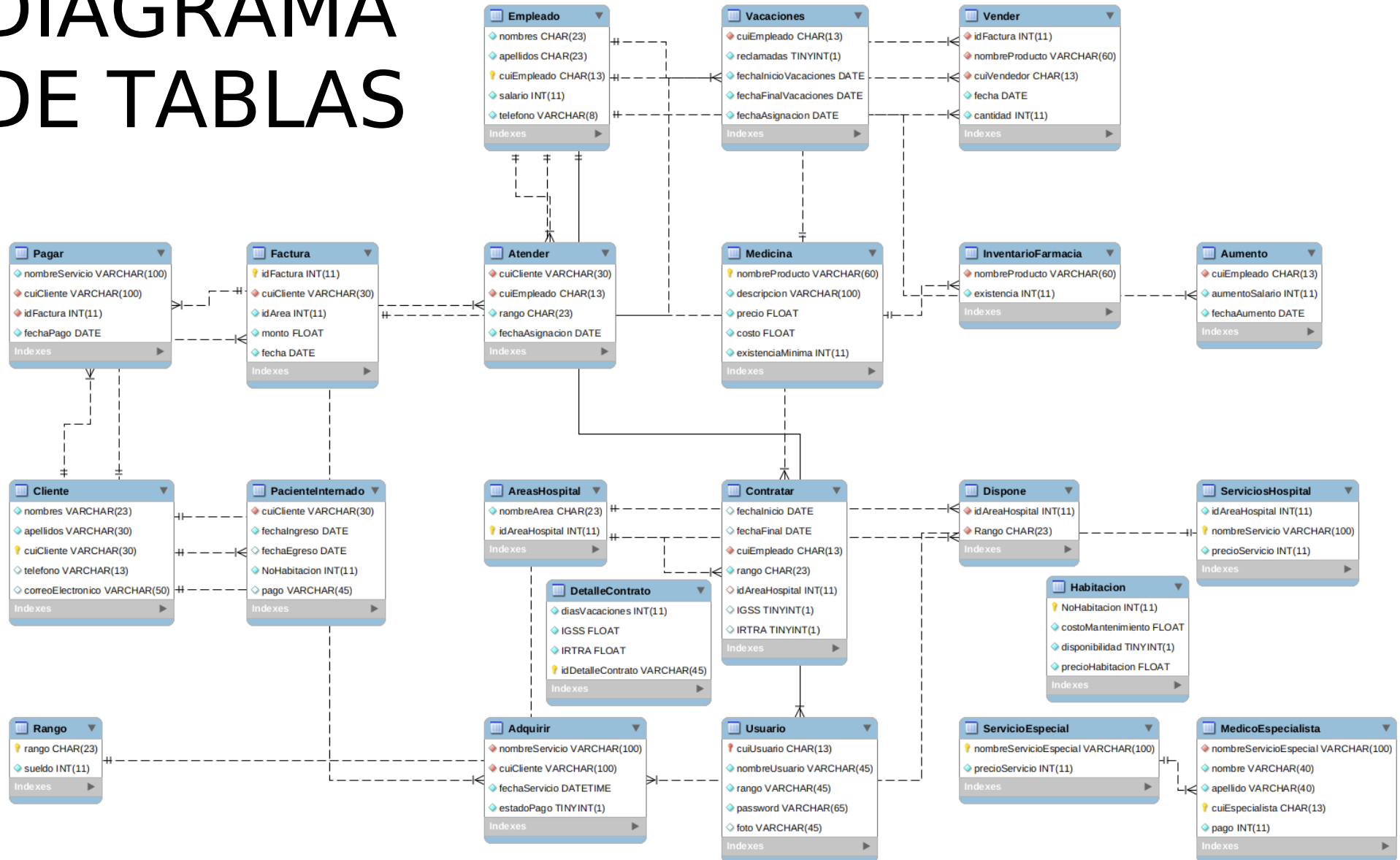


DIAGRAMA DE E/R

