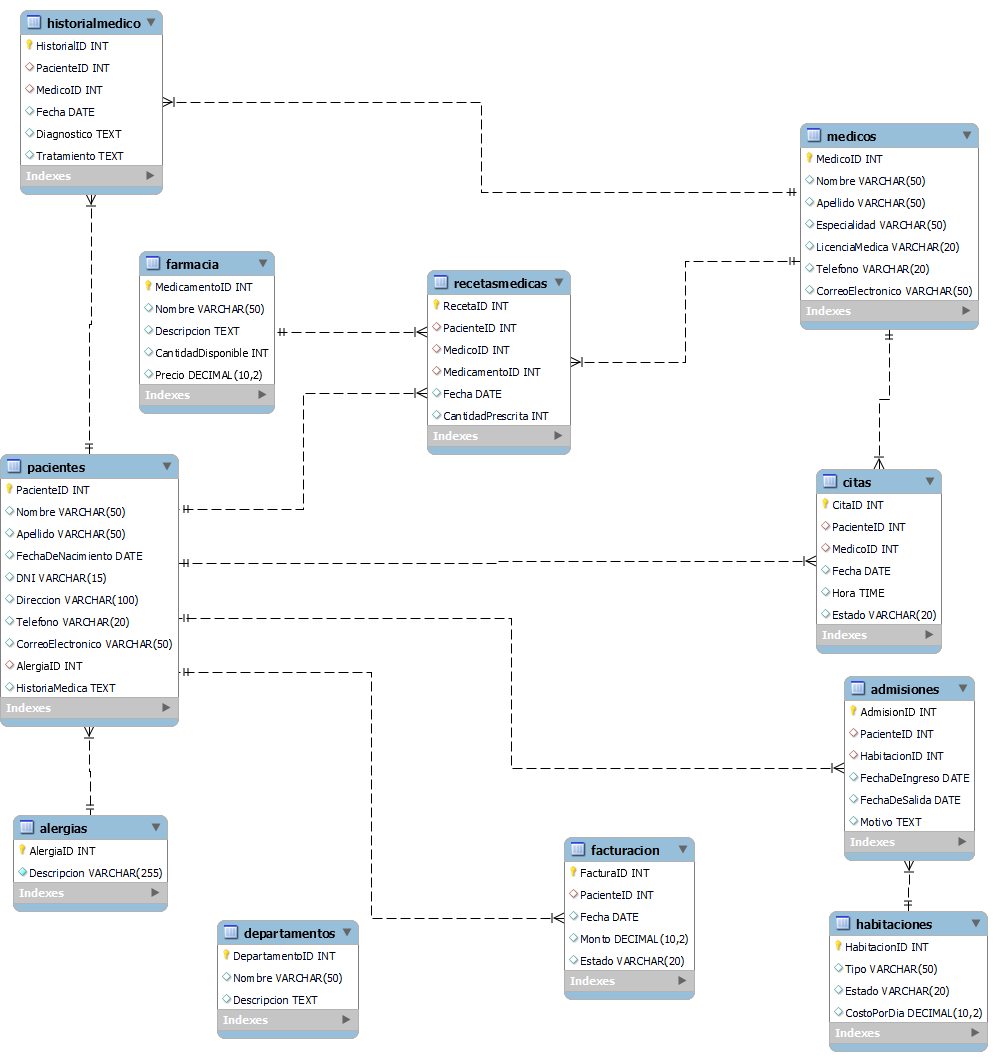
CREACIÓN DE UNA APLICACIÓN DE GESTIÓN DE HOSPITAL

Este ejercicio se guardará en el repositorio DAWES-VARIOS en un proyecto llamado “hospital”

Se pide implementar una aplicación de gestión de un hospital el cual posee el siguiente esquema de BBDD (descargaos el fichero de [aquí](https://calasanzescolapiossalamanca-my.sharepoint.com/:f:/g/personal/kikesainz_calasanzescolapiossalamancao365_educamos_com/EmUEz0wLkExDmPORxgf3p3cBq8B7k9Jz7wFsqzDB95DfNw?e=uGq5Gh)):

Se pide implementar una aplicación usando Java SE que permita llevar a cabo las siguientes operaciones en la BBDD:

Tabla EstadoCitas

* CREATE: Permitirá crear un “Estado Cita” nueva.
* READ: Permitirá listar los tipos de estados de citas. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar solamente la descripción.

Tabla EstadoFacturacion

* CREATE: Permitirá crear un “Estado Facturación” nueva.
* READ: Permitirá listar los tipos de estados de facturación. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar solamente la descripción.

Tabla Pacientes

* CREATE: Permitirá crear un paciente nuevo.
* READ: Permitirá listar los pacientes. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del paciente (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**

TABLA Médicos

* CREATE: Permitirá crear un médico nuevo.
* READ: Permitirá listar los médicos. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del paciente (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**

TABLA Citas

* CREATE: Permitirá crear una cita nueva.
* READ: Permitirá listar los médicos. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del paciente (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla Departamentos

* CREATE: Permitirá crear un departamento nuevo.
* READ: Permitirá listar los departamentos. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del departamento (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**
* DELETE: Permitirá borar un departamento. En este caso no haremos un borrado físico, sino que pondremos el campo activo a cero.

Tabla HistorialMedico

* CREATE: Permitirá crear un Historial Médico nuevo.
* READ: Permitirá listar los Historiales médicos. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del Historial médico de un paciente (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla Facturación

* CREATE: Permitirá crear un Historial Médico nuevo.
* READ: Permitirá listar los Historiales médicos. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo del paciente (salvo la clave primaria), **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla Farmacia

* CREATE: Permitirá crear registro en la tabla Farmacia, es decir, introducir un medicamento.
* READ: Permitirá listar los medicamentos incluidos en la tabla Farmacia. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo de la tabla Farmacia, **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla RecetasMedicas

* CREATE: Permitirá crear registro en la tabla RecetasMedicas, es decir, introducir una receta para un paciente.
* READ: Permitirá listar las recetas incluidos en la tabla RecetasMedicas. Se podrá filtrar por cualquiera de los campos.
* UPDATE: Permitirá actualizar cualquier campo de la tabla Farmacia, **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla Habitaciones

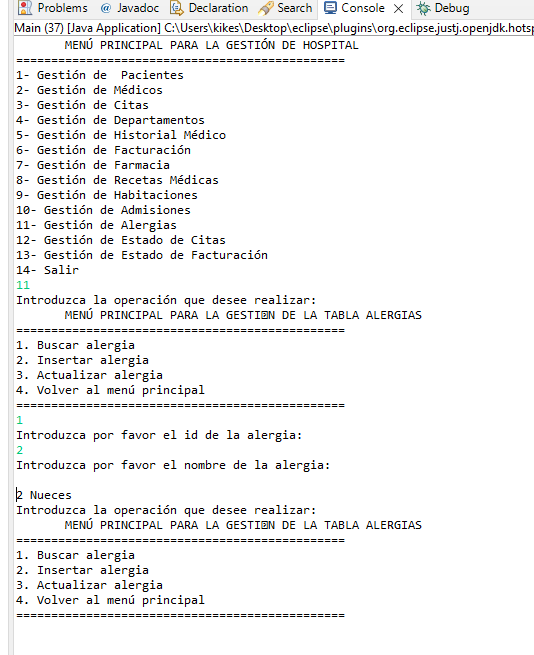
* CREATE: Permitirá crear registro en la tabla Habitaciones, es decir, introducir una nueva habitación en el hospital.
* READ: Permitirá listar las habitaciones registradas en el hospital
* UPDATE: Permitirá actualizar cualquier campo de la tabla Habitaciones, **si el usuario deja un campo en blanco no se debe de actualizar.**

Tabla Admisiones

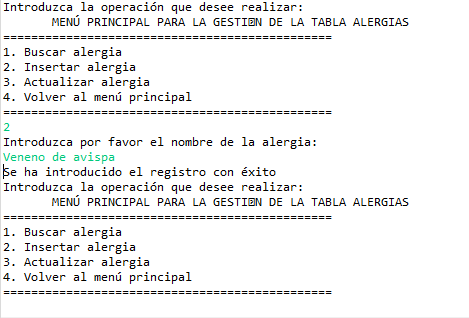
* CREATE: Permitirá crear registro en la tabla Admisiones, es decir, registrar un ingreso de un paciente.
* READ: Permitirá listar las habitaciones registradas en el hospital
* UPDATE: Permitirá actualizar cualquier campo de la tabla Habitaciones, **si el usuario deja un campo en blanco no se debe de actualizar.**

Ejemplo de uso con el CRUD de la tabla alergias:

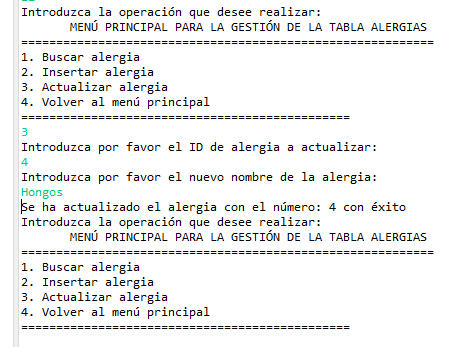
**BÚSQUEDA**



**INSERCIÓN**



**ACTUALIZACIÓN**



Requisitos:

• Se debe seguir el patrón MVC en el que las vistas (menús), los controladores y los modelos estén capas (paquetes) diferentes. Si no se sigue este patrón no se aprobará la práctica.

• Para la conexión a la BBDD se debe crear en un paquete llamado “utils” una clase y un método estático que nos devuelva un objeto Connection. (No hacerlo de esta forma restará 3 puntos.

Tenéis la semilla del proyecto de la cual podéis partir [aquí](https://github.com/kikesainz/hospitalInicio.git).