

TEMA

“ARQUITECTURA DE UN SOFTWARE”



DOCENTE: CALEB LOPEZ

TECNOLOGIA Y DESARROLLO DE SOFTWARE

Enfoque del tema

La Arquitectura del Software, es el diseño funcional de un software, y se empieza a estructurar mediante la creación de un prototipo el cual servirá como apoyo en el desarrollo del SI.

Se debe valorar el consumo de memoria, la legibilidad del código y el análisis de la lógica empresarial.



SUBTEMAS

- ▶ TIPOS DE ARQUITECTURA DEL SOFTWARE
- ▶ HERRAMIENTAS TECNOLÓGICAS
- ▶ DEPURACIÓN DE CÓDIGO



DOCENTE: CALEB LOPEZ

TECNOLOGIA Y DESARROLLO DE SOFTWARE

Tipos de arquitectura del software:



DOCENTE: CALEB LOPEZ

TECNOLOGIA Y DESARROLLO DE SOFTWARE

Cuando hablamos de diseño de una base de datos, hacemos uso de diferentes tipos de diagramas, mas cuando hablamos de código o implementación de un lenguaje de programación nos referimos a un modelo diferente, por lo que en esta clase lo dividiremos en esas 2 secciones:

1. MODELO DE BASES DATOS

2. MODELO DE DISEÑO DE LA LOGICA DEL CODIGO

1. MODELO DE BASES DE DATOS:

- Conceptual: En este se encuentra una manera mas real de comprender lo que se hará de manera más lógica.
- Lógico: simplemente es la misma representación conceptual, pero aplicando una estructura de organización de datos, en el se pueden plantear las funciones de las mismas clases o tablas.
- Dentro de las bases de datos el modelo más utilizado es el relacional, el cual utiliza diferentes relaciones con los datos recopilados y extraídos mediante abstracción.

Tipos de diagramas lógicos y conceptuales más comunes:

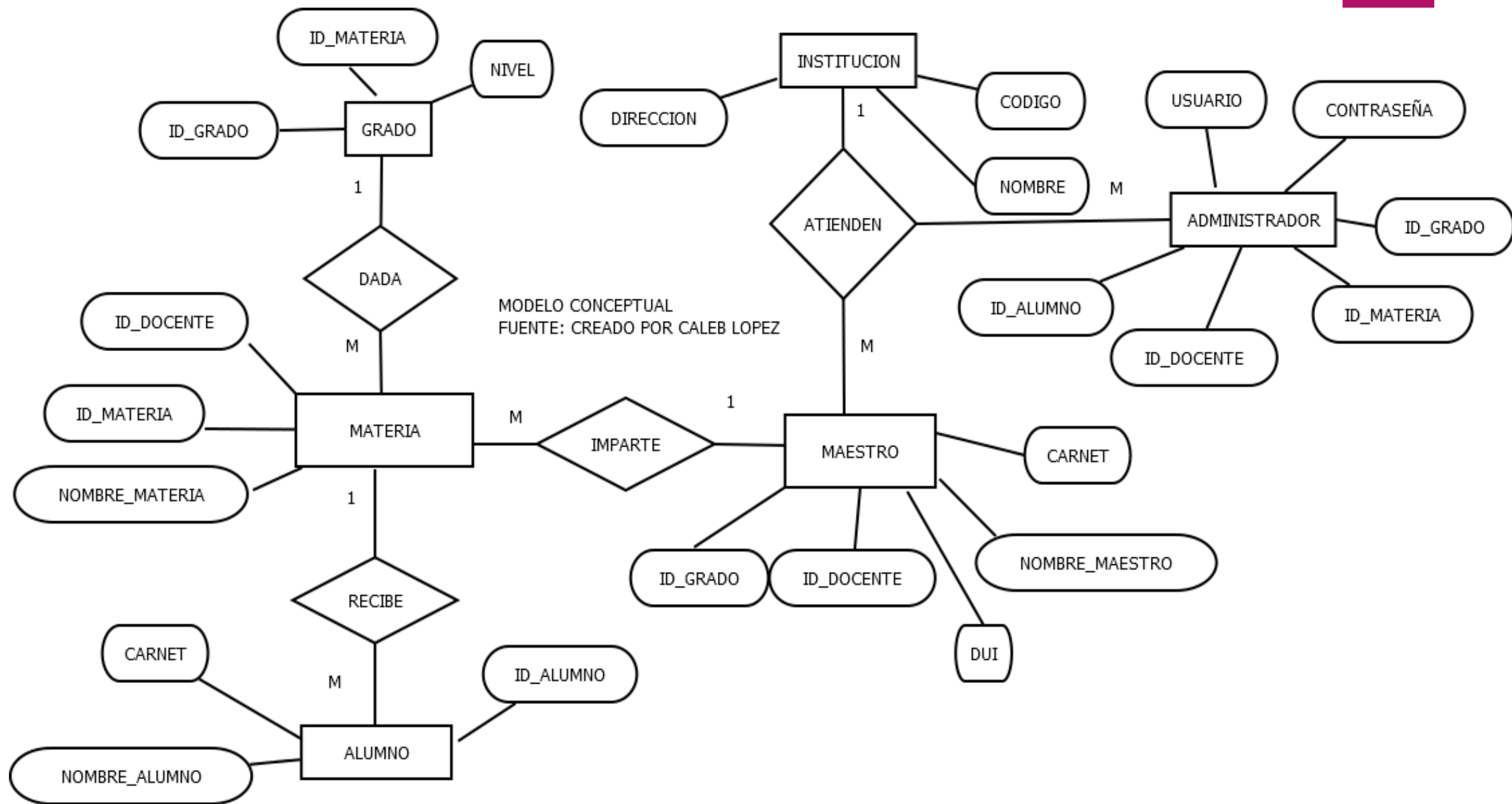
1. Diagrama entidad relacion:

- ▶ Entidades: Las entidades representan las clases.
- ▶ Relacion: utilizamos o asociamos una entidad con la otra mediante una palabra la cual tiene una relacion.
- ▶ Cardinalidad: usamos los diferentes tipos; 1:m, m:m, 1: *, *:1, 1:1.
- ▶ Atributos: son aquellas características de la entidad clase.

Estructura:

- Rombo – relacion
- Rectángulo – entidad
- Elipse – atributos
- Línea – asociación





2. Diagrama de clases:

- ▶ Clase: es la abstracción de un objeto real
- ▶ Métodos: Actividad o función dentro de una clase.
- ▶ Atributos: Son todas las características del objeto o la clase.
- ▶ Cardinalidad: usamos los diferentes tipos; 1:m, m:m, 1: *, *:1, 1:1.

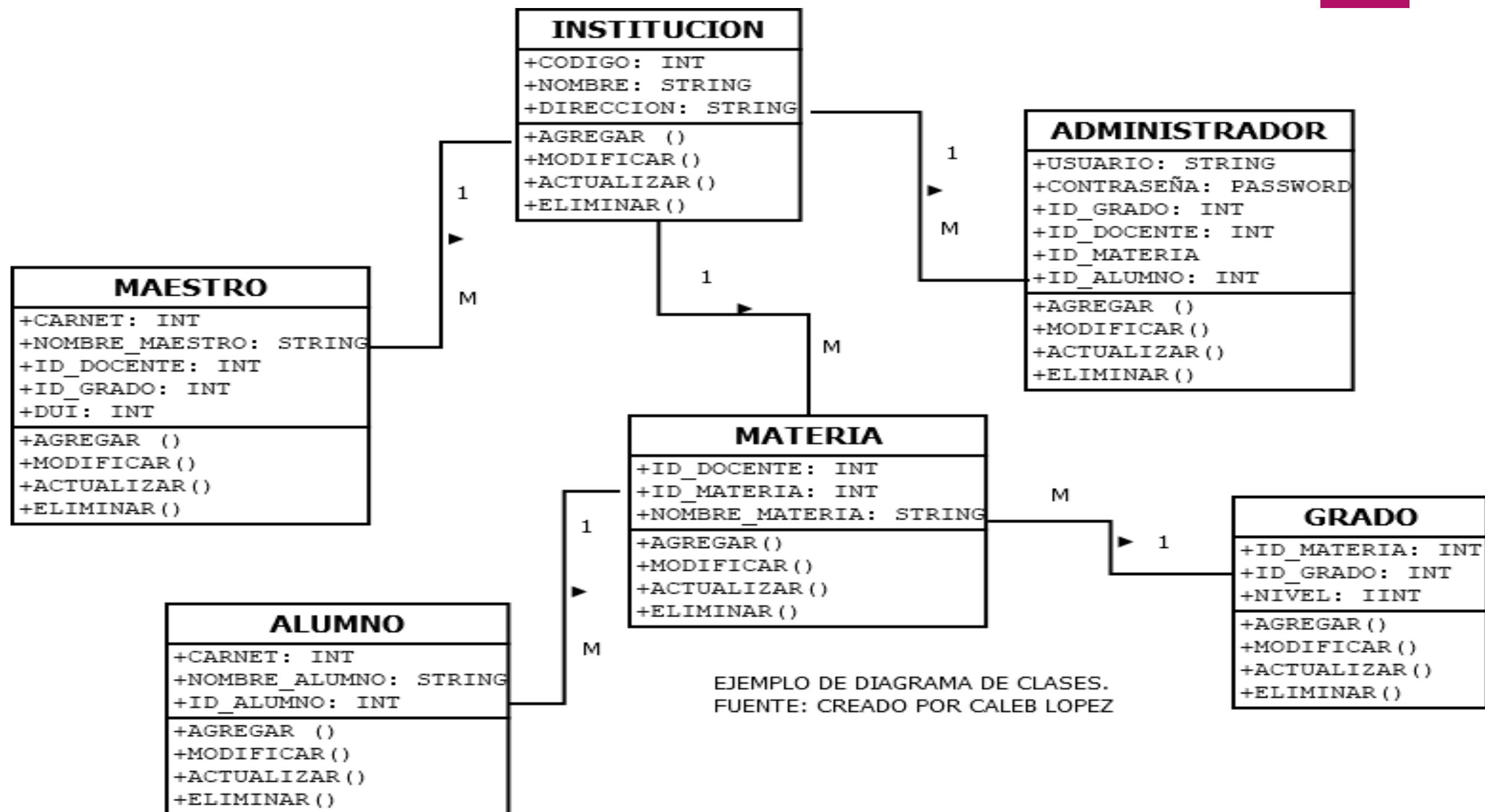


Diagrama de clases a base de código SQL:

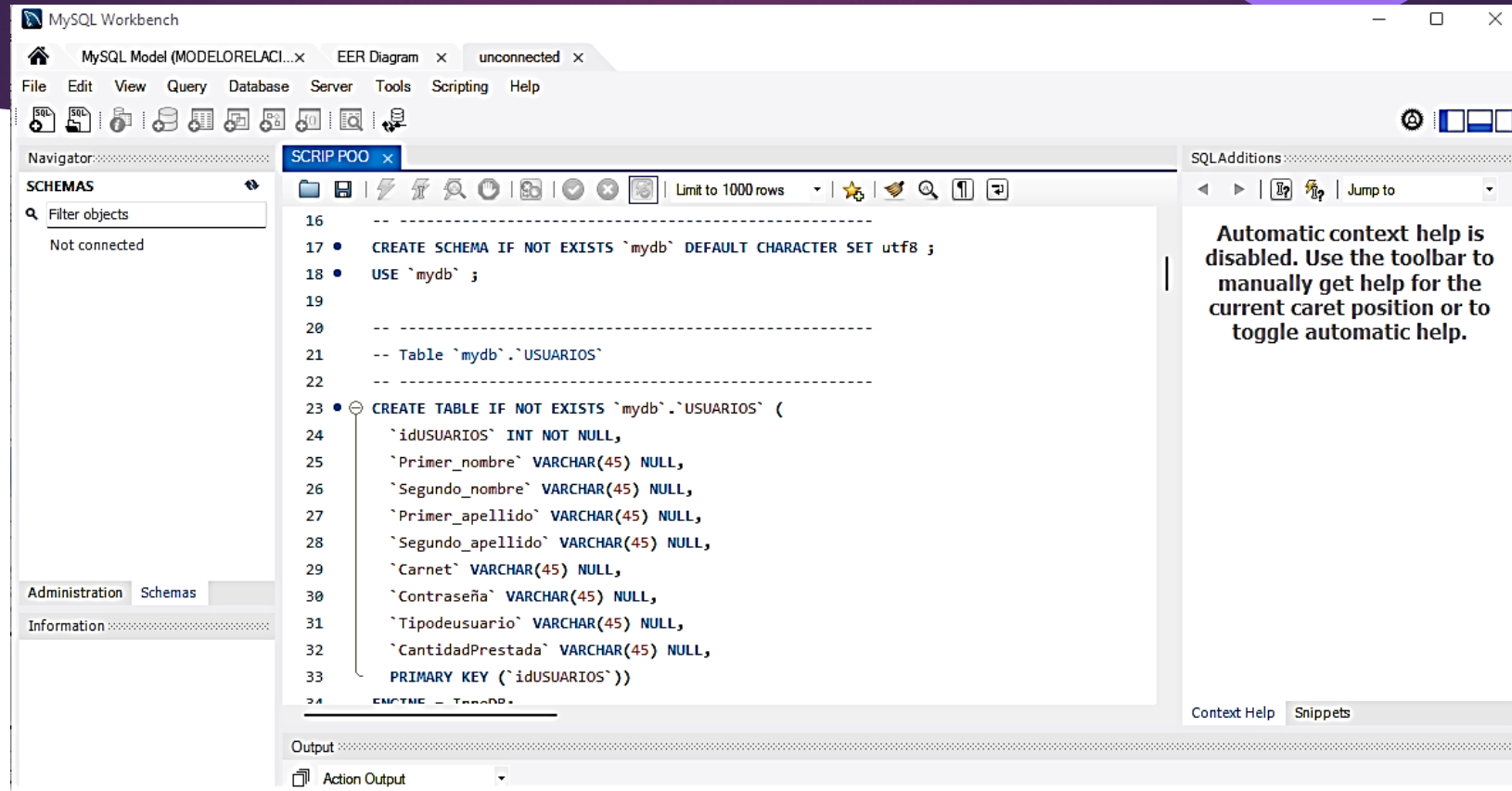


Diagrama relacional con código.
Fuente: maquinasde.com (Caleb López)

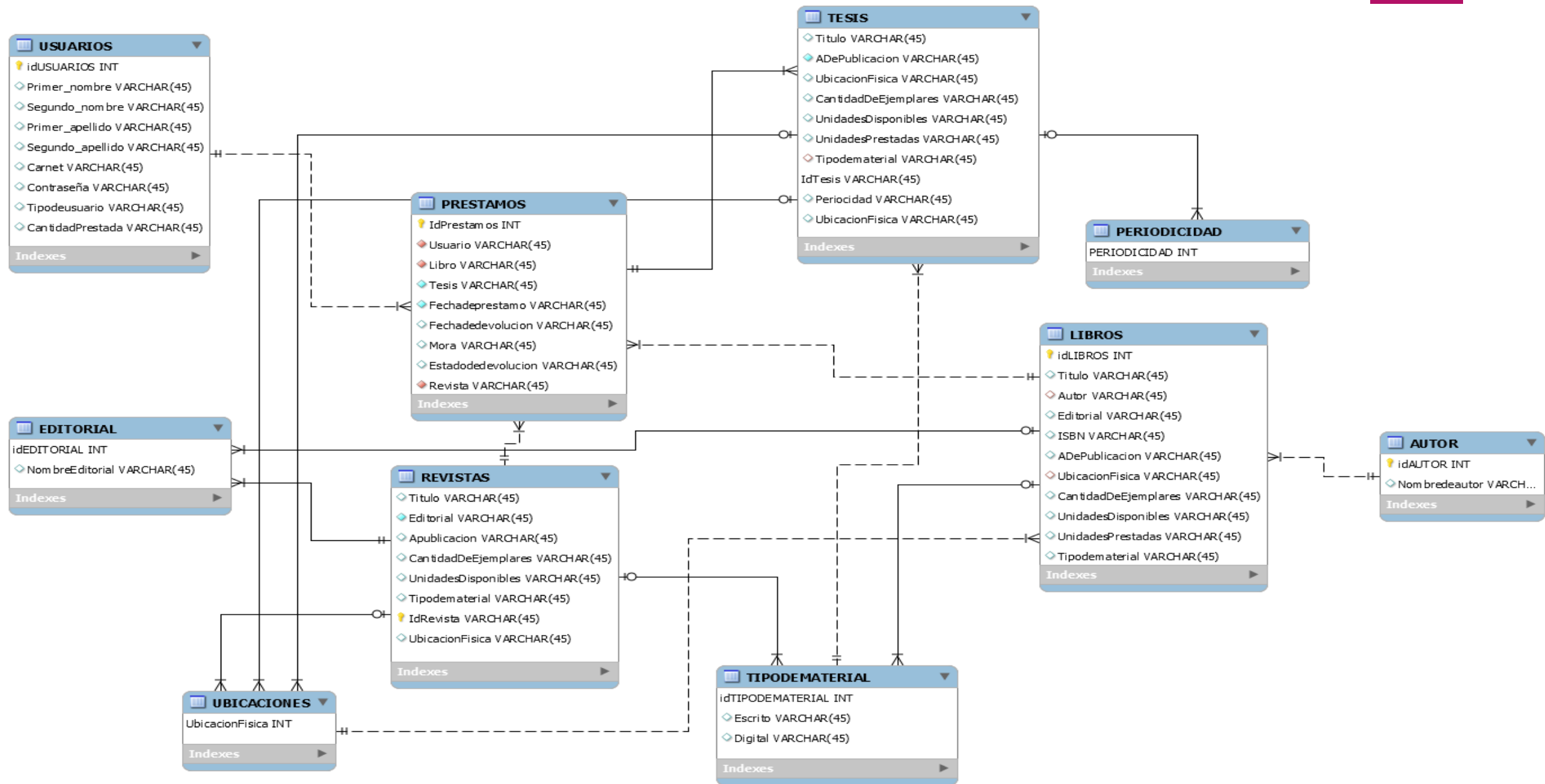


Diagrama Relacional
Fuente: maquinasde.com (Caleb López)

Código en Plantuml

```
EJEMPLODECASODEUSO.wsd  ejemplodiagramadeclasses.wsd 1
C: > Users > MINEDUCYT > OneDrive - Ministerio de Educación, Gobierno de El Salvador > Desk
1  @startuml
2  class Alumno{
3      Nombre
4      apellido
5      edad
6  }
7  }
8  Alumno "0..*" - "1..*" Materia
9  (Alumno, Materia) .. metodos
10
11 class Materia{
12     Nombredelamateria
13     id_alumno
14     id_grado
15 }
16 class metodos {
17     eliminar()
18     cancelar()
19     modificar()
20     actualizar()
21 }
22 Materia "0..*" - "1..*" grado
23 (Materia, grado) .. metodos
24 class grado {
25     id_grado
26 }
27 @enduml
28
```

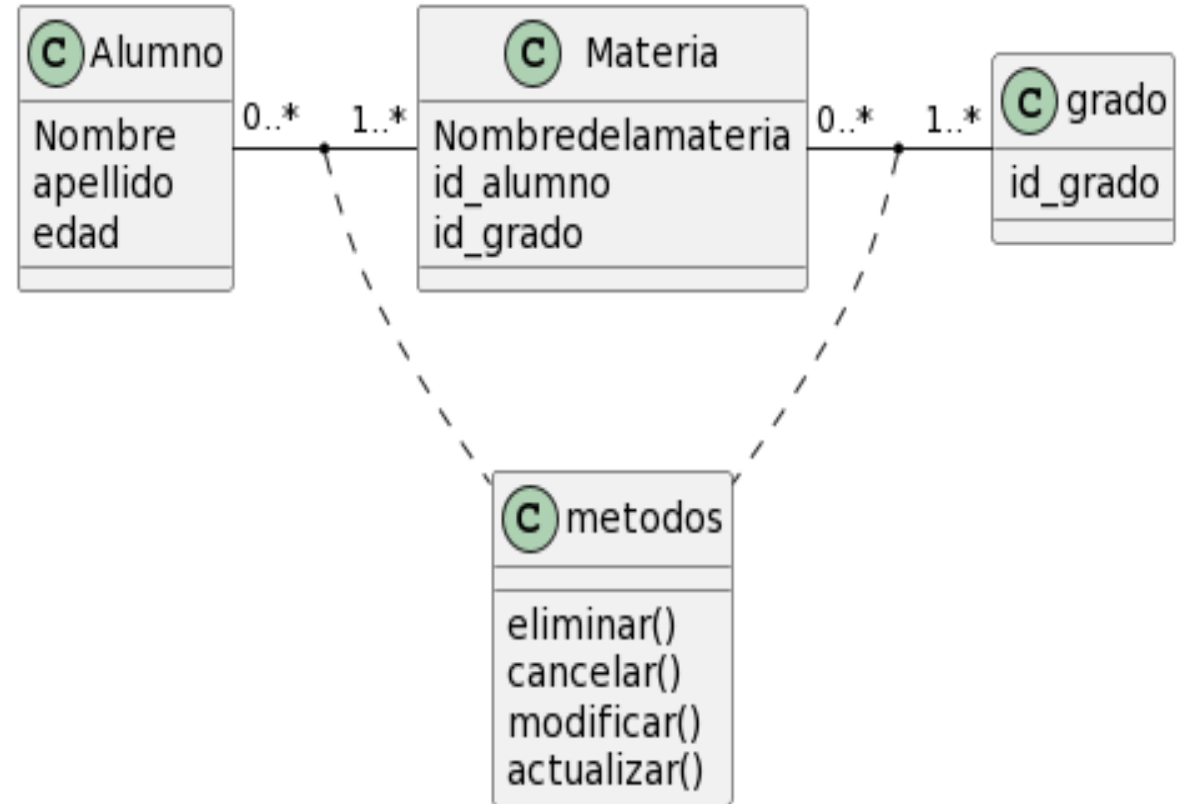


Diagrama de Clases y Código
Fuente: maquinasde.com (Caleb López)

Diagrama de caso de uso:

Actor: Persona quien realiza la acción, se identifica por un dibujo.

Acción: Flujo de la actividad, son los casos de uso, se representa por una elipse.

Sistema : ambiente en que se dan las acciones o el caso de uso y el flujo de la actividad, se dibuja como un rectángulo.

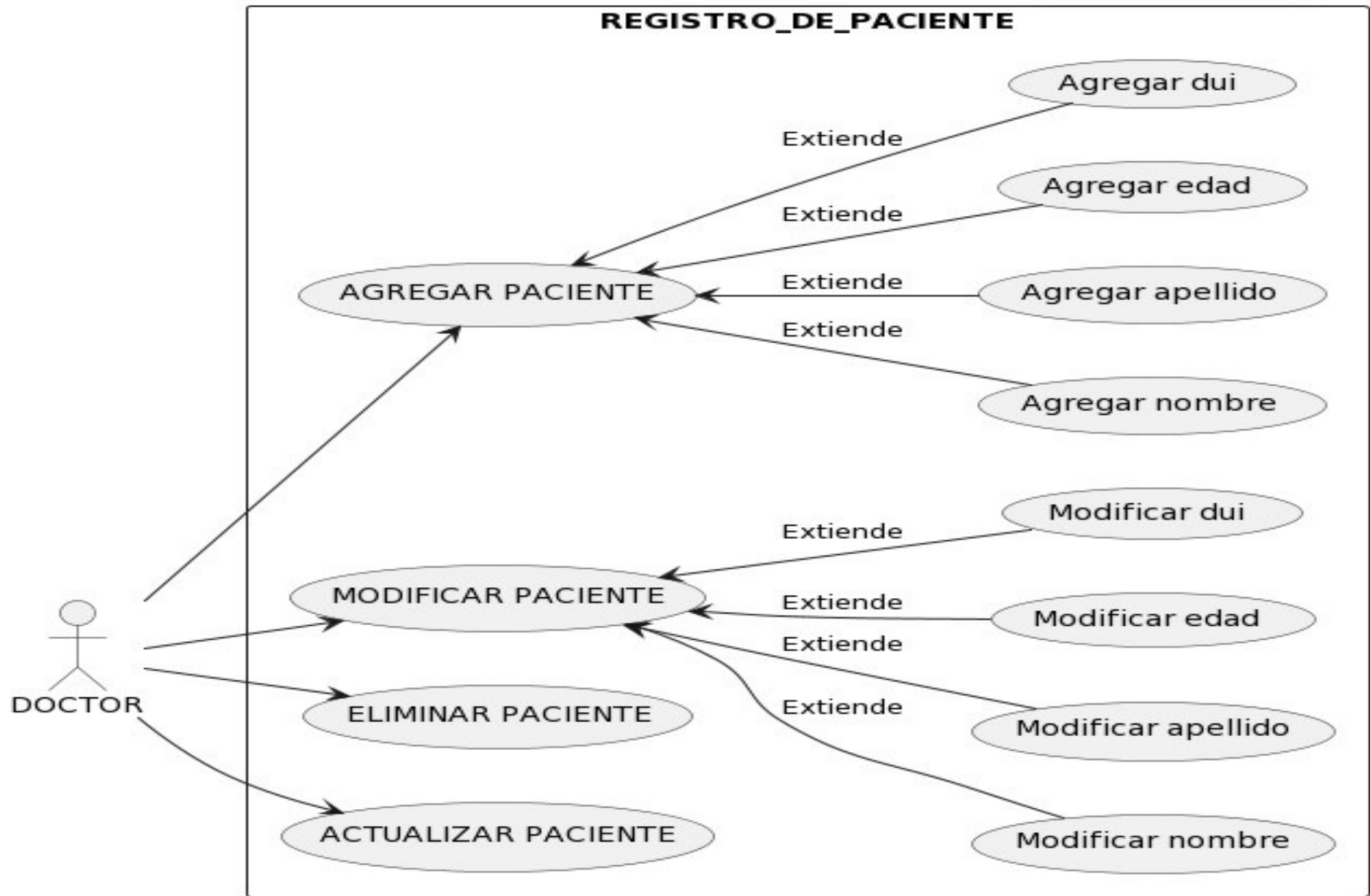
Paquete: su uso es opcional, y significa agrupación de casos de uso.

EJEMPLODECASODEUSO.wsd X

C: > Users > MINEDUCYT > OneDrive - Ministerio de Educación

```
1  @startuml
2  left to right direction
3  actor "DOCTOR" as fc
4  rectangle REGISTRO_DE_PACIENTE {
5      usecase "AGREGAR PACIENTE" as UC1
6      usecase "MODIFICAR PACIENTE" as UC2
7      usecase "ELIMINAR PACIENTE" as UC3
8      usecase "ACTUALIZAR PACIENTE" as UC4
9
10     (UC1)<--(Agregar nombre):Extiende
11     (UC1)<--(Agregar apellido):Extiende
12     (UC1)<--(Agregar edad):Extiende
13     (UC1)<--(Agregar dui):Extiende
14
15     (UC2)<--(Modificar nombre):Extiende
16     (UC2)<--(Modificar apellido):Extiende
17     (UC2)<--(Modificar edad):Extiende
18     (UC2)<--(Modificar dui):Extiende
19
20 }
21
22 fc --> UC1
23 fc --> UC2
24 fc --> UC3
25 fc --> UC4
26 @enduml
```

Ejemplo de caso de uso con
código de Plantuml

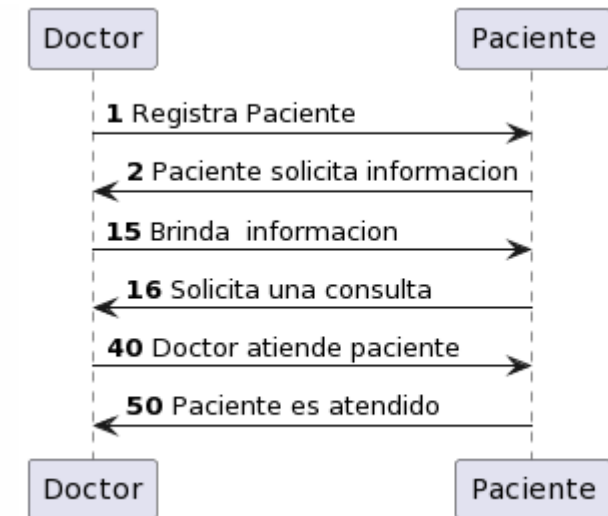


Ejemplo de diagrama de caso de uso con código.
Fuente: maquinasde.com (Caleb López)

Diagrama de secuencia:

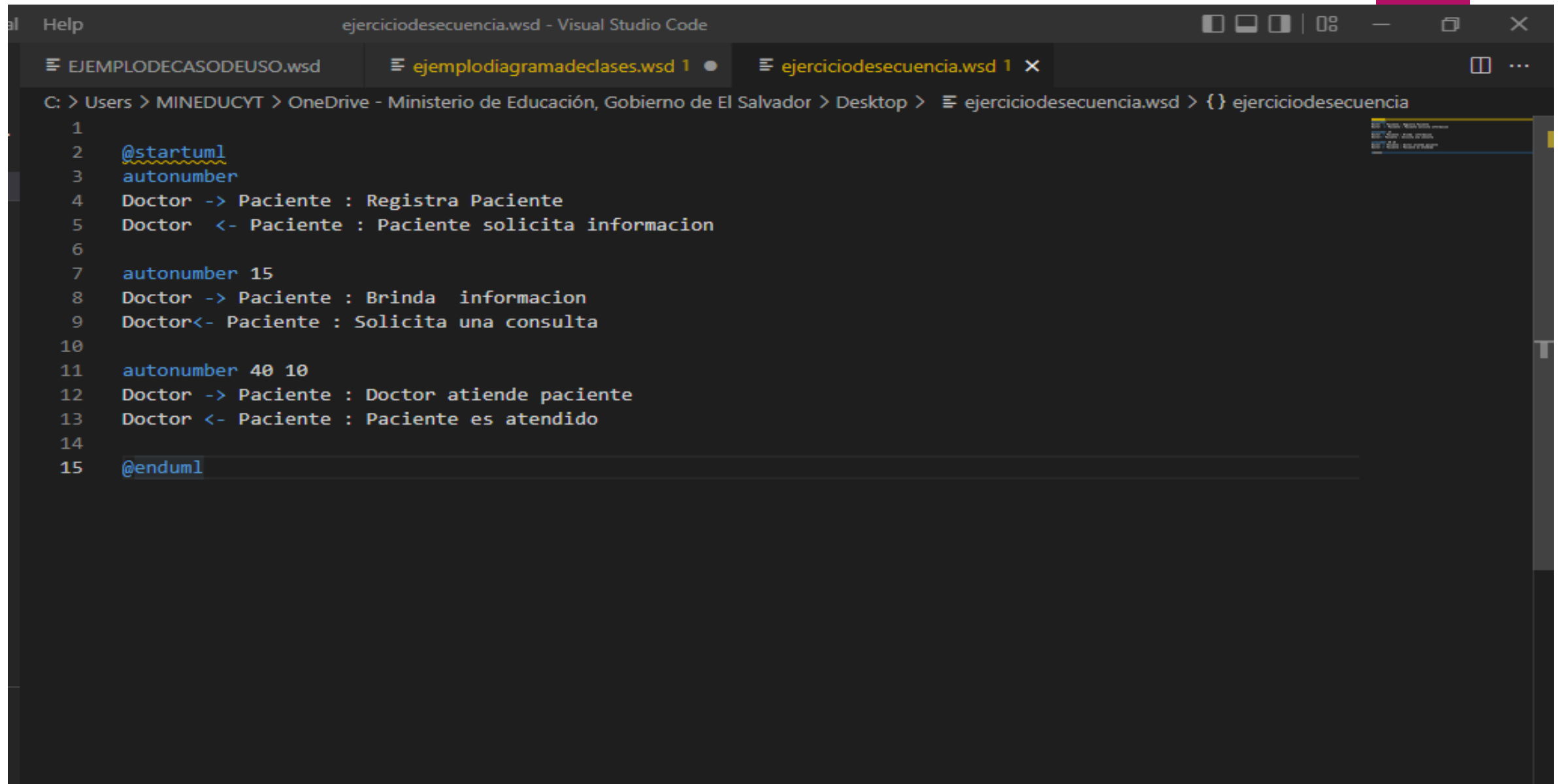
Establece una secuencia de pasos y sigue el parámetro de actores en escena.

Cada paso se encuentra agrupado por medio de números en secuencia.



Ejemplo de diagrama de secuencia con código.

Fuente: maquinasde.com (Caleb López)



The image shows a Visual Studio Code editor window with three tabs open: 'EJEMPLODECASODEUSO.wsd', 'ejemplodiagramadeclasses.wsd 1', and 'ejerciciodesecuencia.wsd 1'. The active tab is 'ejerciciodesecuencia.wsd 1', which contains a sequence diagram written in plantuml format. The code is as follows:

```
1
2  @startuml
3  autonumber
4  Doctor -> Paciente : Registra Paciente
5  Doctor <- Paciente : Paciente solicita informacion
6
7  autonumber 15
8  Doctor -> Paciente : Brinda informacion
9  Doctor<- Paciente : Solicita una consulta
10
11  autonumber 40 10
12  Doctor -> Paciente : Doctor atiende paciente
13  Doctor <- Paciente : Paciente es atendido
14
15  @enduml
```

Código de en plantuml de un diagrama de secuencias.

2. MODELO DE DISEÑO DE LA LOGICA DEL CODIGO

- ▶ En esta parte en la mayoría de los lenguajes se utiliza el modelo de 3 niveles, el cual establece tres capas:

1. Capa de presentación:

Interfaces del software, o fases genéricas del Sistema Informático.

2. Capa de lógica empresarial:

Código o implementación de la solución del problema con el código de programación.

3. Capa de acceso a datos:

Base de datos, estructura lógica, conceptual y relacional, a base de un diseño.

Herramientas tecnológicas:



DOCENTE: CALEB LOPEZ

TECNOLOGIA Y DESARROLLO DE SOFTWARE

Son todas aquellas que nos facilitan el desarrollo de software, desde el uso de editores de código, IDE, gestores de bases de dato, hasta aquellas que nos brindan la planificación del proyecto tecnológico.

Ejemplos de los que hemos utilizado en clase:

- Visual code
- Workbench
- Plantuml
- Trello

Depuración de código:

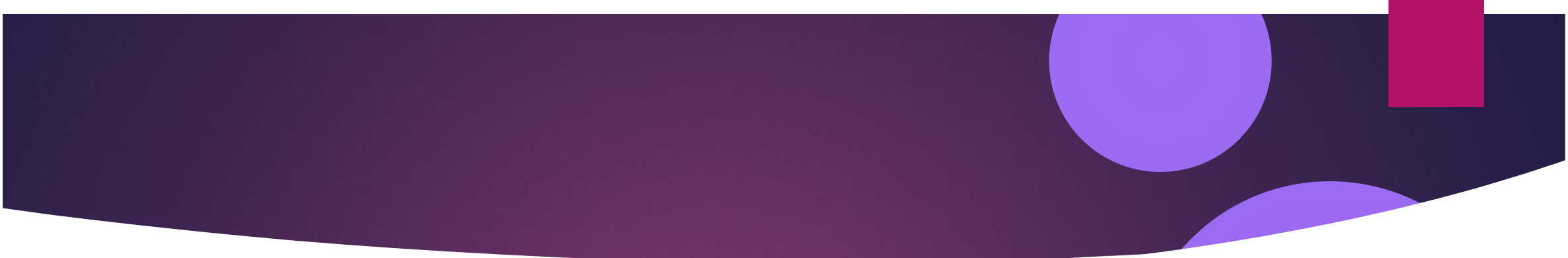


DOCENTE: CALEB LOPEZ

TECNOLOGIA Y DESARROLLO DE SOFTWARE

Es sacar los errores(mistakes) o los mal llamados bugs, cada lenguaje y IDE tiene su propia forma de depurar, y de realizar pruebas de testeo, funcionales y no funcionales.

Bugzilla – es un depurador de Código que cumple una funcion.



CONCLUSION

Hemos concluido aprendiendo diversos temas de arquitectura que nos ayudaran a profundizar en las temáticas futuras en nuestra área de desarrollo de software.

En este tema aprendimos a diseñar y modelar una estructura mas optima para un sistema informático.