

DOUMENTACION DOMENICO_HAMZA

- 1. Diagrama de BBDD**
- 2. Diagram de clase**
- 3. Diagram de caso de uso**
- 4. Prueba JUNIT**

Diagrama de BBDD

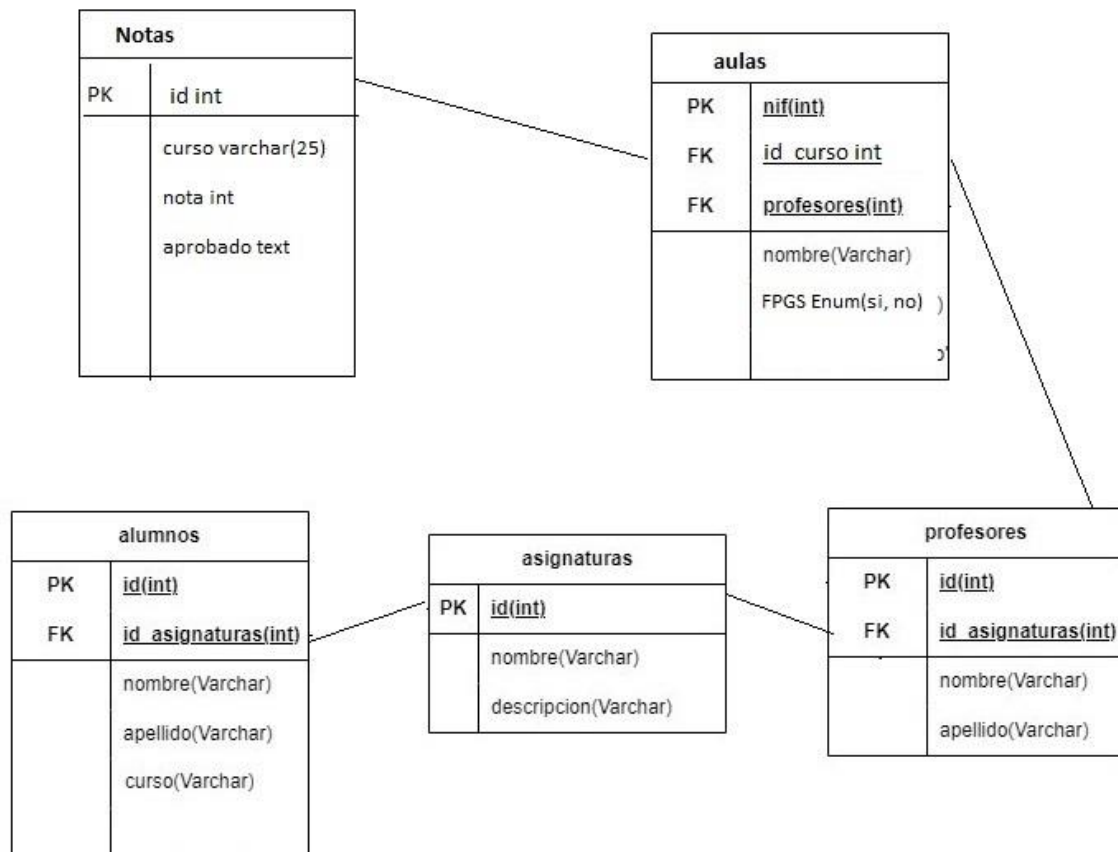


Diagrama de clase:

alumnos
<pre>getid() int setid(id int) void getnombre() string setnombre(nombre string) void getapellido() string setapellido(apellido string) void getcurso() string setcurso(curso string) void</pre>

asignaturas
<pre>getid() int setid(id int) void getnombre() string setnombre(nombre : string) void getdescription() string setdescription(descrip string) void</pre>

profesores
<pre>getid() int setid(id int) void getnombre() string setnombre(nombre: string) void getapellido() string setapellido(apellido: string) void</pre>

notas
<pre>getid() int setid(id int) void getcurso() varchar setcurso(curso varchar) void getnota() int setnota(nota int) void getaprobado() string setaprobado(aprobado string) void</pre>

aulas
<pre>getid() int setid(id int) void getnombre() string setnombre(nombre:string) getfpgs() boolean setfpgs(fpgs: boolean) void</pre>

Casos de prueba:

Flujo principal:

El usuario desea gestionar la base de datos "domenico_hamza".

El sistema muestra las opciones disponibles: Alumnos, Asignaturas, Profesores, Aulas, Notas.

El administrador selecciona la opción "Alumnos".

El sistema muestra las funciones disponibles para la tabla "Alumnos":

getId(): int

setId(id: int): void

getNombre(): String

setNombre(nombre: String): void

getApellido(): String

setApellido(apellido: String): void

getCurso(): String

setCurso(curso: String): void

El administrador elige una función para realizar una operación en la tabla "Alumnos":

- a) Si selecciona "getId()", el sistema devuelve el ID del alumno.
- b) Si selecciona "setId(id: int)", el sistema permite establecer el ID del alumno.
- c) Si selecciona "getNombre()", el sistema devuelve el nombre del alumno.
- d) Si selecciona "setNombre(nombre: String)", el sistema permite establecer el nombre del alumno.
- e) Si selecciona "getApellido()", el sistema devuelve el apellido del alumno.
- f) Si selecciona "setApellido(apellido: String)", el sistema permite establecer el apellido del alumno.
- g) Si selecciona "getCurso()", el sistema devuelve el curso del alumno.
- h) Si selecciona "setCurso(curso: String)", el sistema permite establecer el curso del alumno.

El administrador realiza la operación deseada.

El sistema muestra el resultado de la operación realizada.

El administrador puede volver al paso 4 para seleccionar otra función o seleccionar una opción diferente para gestionar otra tabla.

Flujo alternativo:

En cualquier punto del flujo principal, el administrador puede seleccionar una opción diferente para gestionar otra tabla de la base de datos "domenico_hamza".

Este caso de uso describe cómo un administrador puede utilizar las funciones proporcionadas para interactuar con la tabla "Alumnos" de la base de datos. Los pasos son similares para las otras tablas, simplemente seleccionando la tabla correspondiente y utilizando las funciones específicas de esa tabla.

Recuerda que este caso de uso es una representación general y se puede adaptar o ampliar según los requisitos específicos de tu aplicación o sistema.

Pruba JUNIT:

```
import org.junit.Before;
```

```
import org.junit.Test;
```

```
import static org.junit.Assert.*;
```

```
public class pruebahamza {
```

```
    private DatabaseManager dbManager;
```

```
    @Before
```

```
    public void setUp() {
```

```
        dbManager = new DatabaseManager();
```

```
        dbManager.connect("domenico_hamza");
```

```
    }
```

```
    @Test
```

```
    public void testAlumnoAsignaturaRelation() {
```

```
        Alumno alumno = new Alumno();
```

```
        alumno.setId(1);
```

```
        alumno.setNombre("John");
```

```
        alumno.setApellido("Doe");
```

```
        alumno.setCurso("Cálculo");
```

```
        Asignatura asignatura = new Asignatura();
```

```
        asignatura.setId(1);
```

```
        asignatura.setNombre("Matemáticas");
```

```
        asignatura.setDescripcion("Estudio de los números y las operaciones matemáticas");
```

```
        alumno.setAsignatura(asignatura);
```

```
        dbManager.save(alumno);
```

```
        dbManager.save(asignatura);
```

```
        Alumno alumnoConsultado = dbManager.getAlumnoById(1);
```

```
        Asignatura asignaturaConsultada = dbManager.getAsignaturaById(1);
```

```
        assertEquals(asignatura, alumnoConsultado.getAsignatura());
```

```
        assertEquals(alumno, asignaturaConsultada.getAlumno());
```

```
}
```

```
@Test
```

```
public void testProfesorAulaRelation() {
```

```
    Profesor profesor = new Profesor();
```

```
    profesor.setId(1);
```

```
    profesor.setNombre("Jane");
```

```
    profesor.setApellido("Smith");
```

```
    Aula aula = new Aula();
```

```
    aula.setId(1);
```

```
    aula.setNombre("Aula 101");
```

```
    profesor.setAula(aula);
```

```
    dbManager.save(profesor);
```

```
    dbManager.save(aula);
```

```
    Profesor profesorConsultado = dbManager.getProfesorById(1);
```

```
    Aula aulaConsultada = dbManager.getAulaById(1);
```

```
    assertEquals(aula, profesorConsultado.getAula());
```

```
    assertEquals(profesor, aulaConsultada.getProfesor());
```

```
}
```

```
}
```