ENTORNOS DE DESAROLLO

Aleksander Mosiadz

Diagramas

Para empezar un a base de datos y organizar el código es fundamental tener de base unos diagramas listos para poder desarrollar el contenido y ver si tiene sentido lo que queremos realizar asique empezares con el diagrama de la base de datos

Diagrama de base de datos

Este diagrama se puede contemplar las tablas y las uniones que tiene con las otras tablas para poder hacer una correlación de los datos a través de sus FK (Foring Key) y PK(Primary key), también se puede observar los diferentes datos que deben tener agregadas las diferentes tablas , como nombre , descripción , id , etc..

Gracias a ello se puede tener un orden en el almacenamiento de la información el cual nos será más fácil de acceder o encontrar la información que necesitemos cuanto más desarrollado este

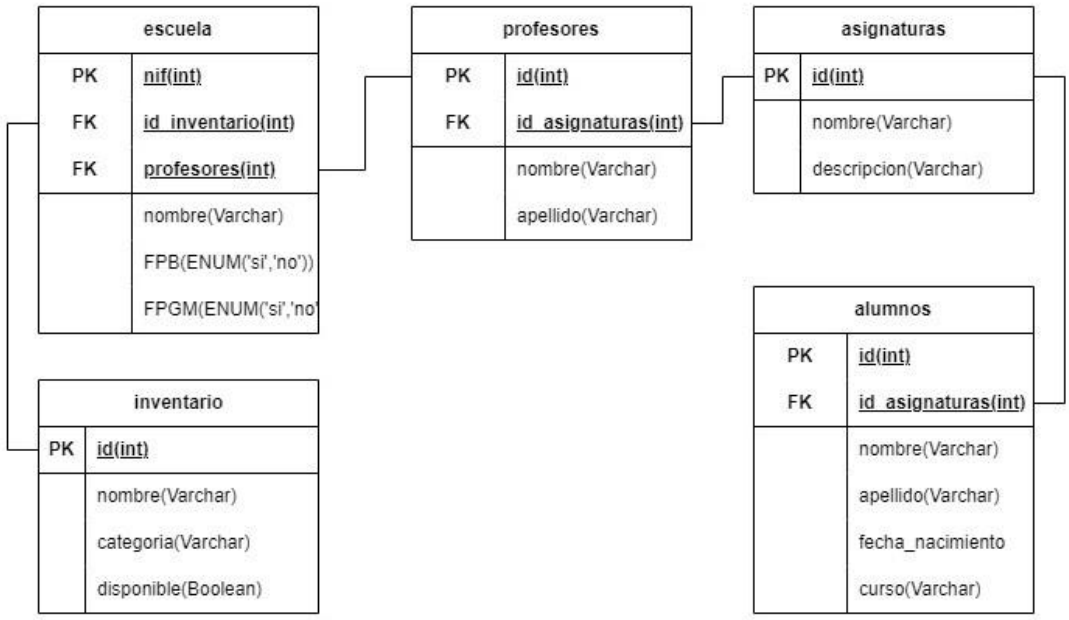


Diagrama de Clase

Este diagrama es mucho más extenso ya que se representan las clases corresponden a las tablas de la base de datos junto con sus atributos, además de que se indican modificadores de acceso a los atributos como los prívate y los tipos de datos de los atributos que son ligeramente alterados o extendidos de forma mucho mas amplia.

Diagrama de comportamiento

Los diagramas de comportamiento se utilizan para poder ver una manera mas visual el funcionamiento de la base de datos o como interactuaría un programa con dicha base incluyendo actores junto sus casos de uso.

En este diagrama de comportamiento podemos observar como profesores tiene una unión a asignatura y imparte asignatura el alumno toma la lección y a la vez el alumno puede solicitar material al inventario pero solo lo puede solicitar si la escuela renueva o repone dicho material , también se puede observar de alumno debe devolver el material solicitado para poder reutilizarlo en el centro

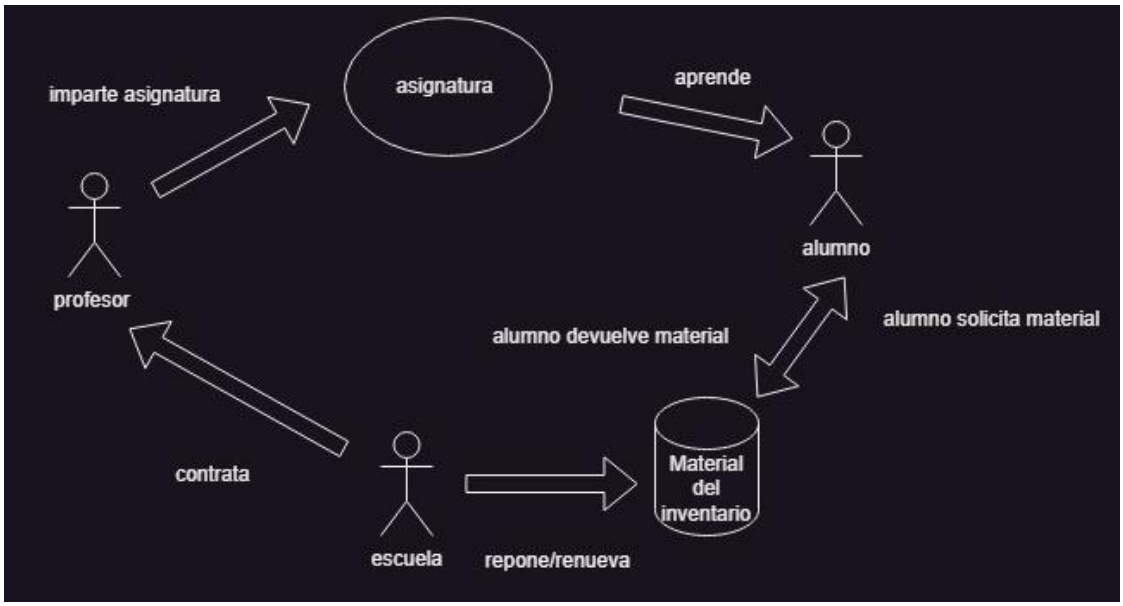
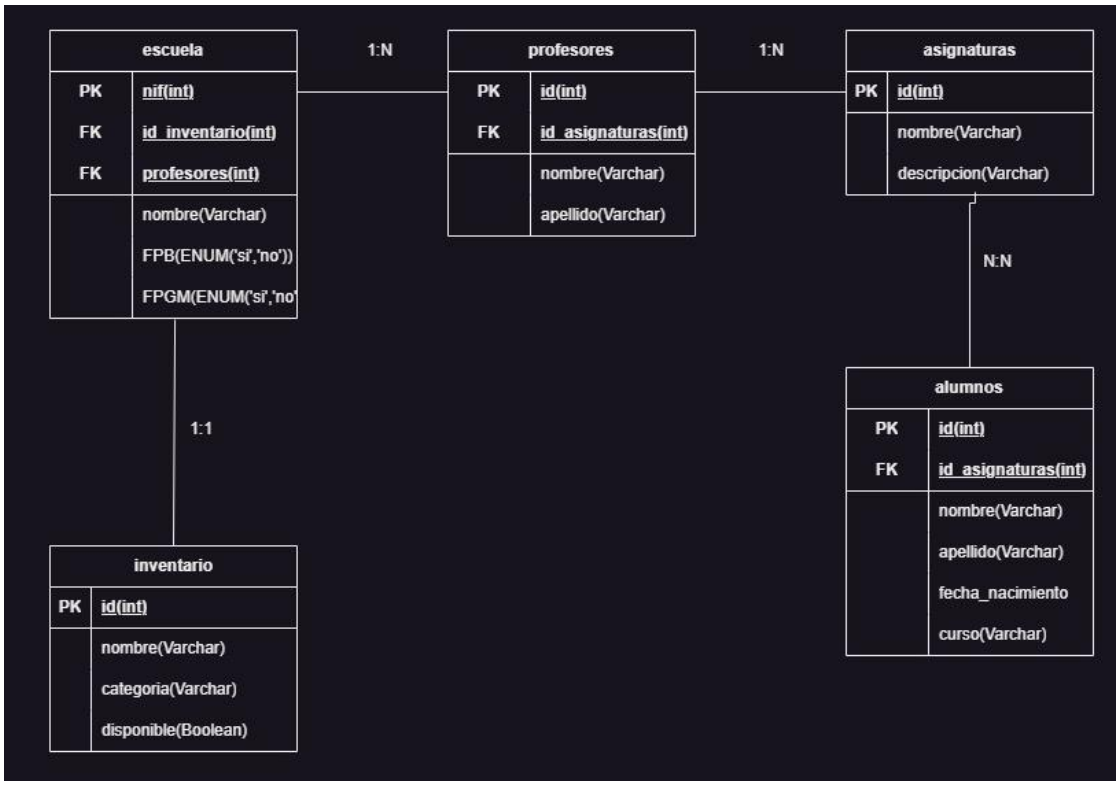


Diagrama de Relación



Este tipo de diagramas se pueden ver ciertos números entre las líneas en los cuales podemos determinar si un dato de esa tabla es equivalente a un uno a uno lo cual seria de que cada escuela solo tiene un inventario u otro ejemplo escuela tiene varios profesores pero los profesores no tiene varias escuelas ya que si no acabarías quemados por el estrés o peor harían más dinero que del que deberían por lo que hacienda metería mano y tal vez pierdan el 46% por ciento de sus ganancias por los dichosos impuestos , pero bueno retomando el tema también se puede observar una tabla que es de varios a varios (n:n) entonces hay habría que hacer una tabla intermedia para poder asociar a un alumno varias asignaturas y a asignaturas varios alumnos pero por los recortes de la educción que fueron redirigidos en el ministerio de igualdad no nos da el presupuesto para crear una tabla nueva lamentamos las molestias .

CASO DE USO

El caso de uso es una técnica utilizada en la ingeniera de software para capturar y describir las interacciones entre los usuarios y un sistema, se representa la funcionalidad, especificaciones Dels sistema, descripción de lo que haría en disco contexto

Esta segmentado en 7 partes

- Nombre del caso
- ID para poder identificarlos
- Una Descripción breve del propósito
- Precondiciones para saber que se necesita
- El curso o flujo de los eventos que deben suceder también nombrado curso normal
- Postcondiciones las condiciones que se cumplen después de que el caso se haya ejecutado
- Por ultimo las alternativas que pueden suceder si no se cumple alguna condición

Caso de Uso: Gestionar Asignaturas

ID: UC001

Descripción: Este caso de uso permite al usuario gestionar las asignaturas en el sistema.

Actores:

- Usuario: Persona que interactúa con el sistema.

Precondiciones:

- El usuario ha iniciado sesión en el sistema.

- Existen asignaturas registradas en el sistema.

Curso Normal del Caso de Uso:

1. El usuario accede a la funcionalidad de gestión de asignaturas en el sistema.

2. El sistema muestra la lista de asignaturas disponibles.

3. El usuario selecciona una opción del menú:

   a. Crear Asignatura:

      i. El sistema solicita los datos de la nueva asignatura al usuario (nombre, descripción, etc.).

      ii. El usuario proporciona los datos requeridos.

      iii. El sistema valida y registra la nueva asignatura en la base de datos.

      iv. El sistema muestra un mensaje de éxito y vuelve al menú principal.

   b. Actualizar Asignatura:

      i. El usuario selecciona una asignatura de la lista para actualizar.

      ii. El sistema muestra el formulario con los datos actuales de la asignatura.

iii. El usuario modifica los datos necesarios.

iv. El sistema valida y actualiza los datos de la asignatura en la base de datos.

v. El sistema muestra un mensaje de éxito y vuelve al menú principal.

c. Eliminar Asignatura:

i. El usuario selecciona una asignatura de la lista para eliminar.

ii. El sistema muestra una confirmación para confirmar la eliminación.

iii. El usuario confirma la eliminación.

iv. El sistema elimina la asignatura de la base de datos.

v. El sistema muestra un mensaje de éxito y vuelve al menú principal.

4. El usuario selecciona la opción de salir del caso de uso.

5. El sistema finaliza el caso de uso y regresa al menú principal.

Postcondiciones:

- Los cambios realizados (creación, actualización o eliminación de asignaturas) se reflejan en la base de datos.

- El sistema muestra un mensaje de éxito al completar las operaciones.

Alternativas:

- En el paso 3a, si el usuario no proporciona los datos requeridos correctamente, el sistema muestra un mensaje de error y vuelve al formulario para corregir los datos.

- En el paso 3b, si el usuario selecciona una asignatura que no existe, el sistema muestra un mensaje de error y vuelve al menú principal.

- En el paso 3c, si el usuario cancela la eliminación, el sistema vuelve al menú principal sin realizar cambios en la base de datos.

- En el paso 4, si el usuario decide cancelar el caso de uso, el sistema regresa al menú principal sin realizar cambios en la base de datos.

Prueba de JUNIT

Este sería el código de prueba cuándo tengamos montado todo para saber de que funcionan la relaciones :

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

```java
public class ClasesTest {

    @Test
    public void testCrearProfesor() {
        Profesor profesor = new Profesor(1, "Juan", "Pérez");
        assertEquals(1, profesor.getId());
        assertEquals("Juan", profesor.getNombre());
        assertEquals("Pérez", profesor.getApellido());
    }

    @Test
    public void testCrearInventario() {
        Inventario inventario = new Inventario(1, "Laptop",
"Electrónica", true);
        assertEquals(1, inventario.getId());
        assertEquals("Laptop", inventario.getNombre());
        assertEquals("Electrónica", inventario.getCategoria());
        assertTrue(inventario.isDisponible());
    }

    @Test
    public void testCrearEscuela() {
        Inventario inventario = new Inventario(1, "Laptop",
"Electrónica", true);
        Profesor profesor = new Profesor(1, "Juan", "Pérez");
        Escuela escuela = new Escuela(123456789, "Colegio XYZ", true,
false, inventario, profesor);
        assertEquals(123456789, escuela.getNif());
        assertEquals("Colegio XYZ", escuela.getNombre());
        assertTrue(escuela.isFPB());
        assertFalse(escuela.isFPGM());
        assertEquals(inventario, escuela.getInventario());
        assertEquals(profesor, escuela.getProfesor());
    }
}
```

Que es GIT?

Git es un sistema de control de versiones utilizado para el seguimiento de cambión es archivos y directorios a lo largo del tiempo, es un lugar que se dedica solo a guardar las diferencias que encuentra con lo que subes y el archivo original es decir si yo pongo

2+2=4

5+5=10

El tomará eso como referencia entonces cuando yo suba otro archivo el mirará el archivo y si ve que se a añadido algo como 3+2 = 5 únicamente añadirá eso por lo que no recibirá tanta carga de datos .