



## Data Model1::ERDiagram1

Entidades a persistir @Entity para usuario y alumno.  
 Usuario: porque representa cada usuario con su identificación única, además contiene el nombre usuario y la contraseña para inicio de sesión en la autenticación.  
 Alumno, porque representa cada una de las filas a persistir, como máximo en este sistema, 3 alumnos por archivo.  
 No decidimos persistir las clases que solo poseen comportamiento y aquellas cuyo datos se guardan en memoria pero no es necesario persistirlo en la base de datos.  
 Todas las validaciones forman parte de un mismo método y se van implementando con el código pero no es necesario persistirlas en la base de datos.  
 Tampoco nos interesa persistir el path de cada archivo.  
 Asimismo, IntentoInicioSesión no posee ningún dato a persistir es decir son temporales mientras el sistema esta en funcionamiento y se almacenan en memoria.  
 Una vez que se finaliza, el sistema se reinicia.

La base de datos esta desnormalizada debido a que hay redundancia de datos, como por ejemplo el nombre de la materia que se repite.

Para mejorar la base de datos podría normalizarla con respecto a alumnos pero no esta dentro de los requerimientos y sería parte de otra iteración. Lo haría creando una tabla Materia y una tabla nota, debido a que un alumno podría tener muchas notas de una misma materia. La tabla nota tendría la FK id\_alumno e id\_materia. A su vez pondría constraint para que nombre de materia y alumno con apellido no se repita. Al importar el archivo por cada fila busca si existe el alumno usando apellido y nombre, si no existe lo crea, lo mismo con materia, agregando una nueva entrada en nota con id\_alumno e id\_materia.

Las PK son de tipo de dato integer y autoincremental para que la base de datos asigne la identificación única y delegar la responsabilidad

utilizo varchar [50] para nombre de usuario, nombre, apellido, materia con el objetivo de limitar los espacios de memoria y varchar 64 debido a que la contraseña como máximo puede tener 64 caracteres

ultimoCambioContraseña: me guarda la fecha y hora en que se realizo el último cambio de contraseña. Esta información es utilizada para exigir la rotación de contraseña

Usuario esta asociado a uno o muchos alumnos pero cada id\_Alumno estara asociado a un único id\_usuario. FK permite la relación de Alumno con Usuario, de uno a uno o muchos.

Impedance Mismatch:  
 Clases por tablas  
 Comportamiento vs Constraints  
 Utilizamos PK para la identificación única de las tablas y las hacemos autoincremental para facilitar la asignación, delegando la responsabilidad de identidad de la base de datos.  
 Los strings sin limitaciones se convierten en VARCHAR con longitud definida.

