MODELOS:

User:

- username
- pass
- profile
- associateProfile
- active

Company:

- name
- mentorName
- mentorEmail
- mentorPhone
- active

Mentor:

- fullName
- active

Student:

- fullName
- course
- evaluation
- courseYear
- mentor
- company

Date:

- date

- courseYear

- evaluation

PracticeRecord:

- associateStudent

- associateDate

- hours

- description

SERVICE:

Una vez hecho el login, el front guarda en usuario entero en una variable.

public User login(username¿,pass);

- Si hay algún problema con la consulta lanzará UserException

- Hay que comprobar que tenga un alumno asociado, el atributo profile tendrá que ser estudiante, y habrá que comprobar 1º que tenga un alumno asociado, 2º, que ese alumno exista (repository.findById()). Tendrá que lanzar una excepción llamada WrongUserException¿

- Cifrar la contraseña dada y comparar si el username y pass son iguales. Sino lanza IncorrectDataException

- Comprobar si está el alumno está activo

- Consultar el usuario entero y devolverlo

public User changePass(newPass, user);

- Si hay algún problema con la consulta lanzará UserException

- Comprobar si tiene más de 8 caracteres (se puede comprobar desde el modelo con @Size)

- cifrar la contraseña y ponérsela al usuario

- devuelve el usuario con la contraseña cambiada

public User showUser(user);

- Si hay algún problema con la consulta lanzará UserException

- Habrá que crear un repository para las prácticas asociadas

public List<PracticeRecord> consultAllRecords(user, date1, date2, String stateDate);

- Si hay algún problema con la consulta lanzará UserException

- se hace la consulta entre las dos fechas dadas

- Usa consulta para ver en qué estado está el registro (si no está completa se supone que el único registro que tiene es la fecha, el alumno tiene que hacer un registro cada día, si un día no lo hace, se crea uno de ese día sin datos, un registro incompleto)

- Si los filtros están vacíos significa que no hay ningún filtro, se trae todo

public void deleteRecord(idRecord);

- Si hay algún problema con la consulta lanzará UserException

- borrar un registro por el id

public void createRecord(PracticeRecord)

- Si hay algún problema con la consulta lanzará UserException

- Crear un registro a partir del que se pasa