




JANUARY 11, 2021

WS PROJECT REPORT

RICARD PLANA LLAURADÓ
47693712S
UDL – Distributed Computing



Contents

Classes	2
Database.....	3
SQL.....	3
Examgrades	3
Exams	4
Users.....	4
Client	4
GitHub	4
Outputs and executions	5
Menus.....	5
Professor menu	5
Student menu.....	5
Student.....	6
Read all exams (common in both menus).....	6
Read one exam (common in both menus).....	6
Download all exams (common in both menus)	7
Download one exams (common in both menus)	8
Download all grades from a student (common in both menus)	8
Quit option (common in both menus)	8
Professor	9
Insert exam.....	9
Modify description	9
Delete exam	10
Insert Grades	10
Download all grades from exam	11

Classes

In the start of the project only one class was created, being this one the Exam.java this one will hold the next values, including at the same time the getters and setters for them.

```
package myRESTwsExam;

public class Exam {

    private String idexam;
    private String description;
    private String examdate;
    private String examtime;
    private String location;
```

After being developing and testing more services realized that while sending one exam didn't have any problem, while trying to send some at once (or even all of the ones stored in the db), some troubles appeared, so finally a container class called ExamPacks.java was created, the only use of this one was so managed multiple exams at once. So doing it was way easy to manage the exams in the different services provided.

```
package myRESTwsExam;

import java.util.ArrayList;

public class ExamsPack {
    ArrayList<Exam> values = new ArrayList<Exam>();

    public void addExam(Exam _exam)
    {
        this.values.add(_exam);
    }

    public ArrayList<Exam> getExams()
    {
        return this.values;
    }
}
```

At same time while doing the part of store and retrieve grades from exams, the same structure was followed avoiding having the same problems as before.

So we end with the next structure for both of classes. (Always including the needed setters and getters).

```
package myRESTwsStudent;

public class Student {

    private String studentid;
    private String grade;
    private String examid;

package myRESTwsGrades;

import java.util.ArrayList;

public class Grades {

    private ArrayList<Student> listGrades = new ArrayList<Student>();

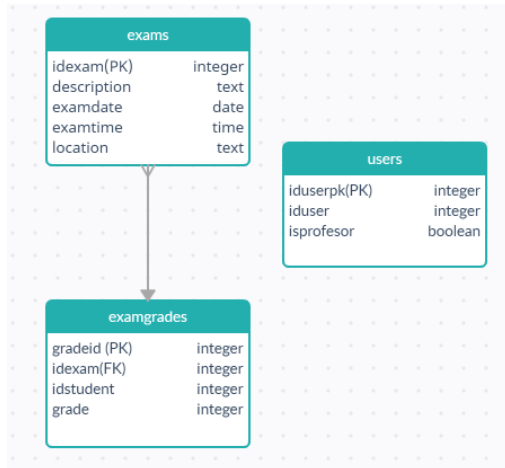
    public void addStudent(Student _student)
    {
        this.listGrades.add(_student);
    }

    public ArrayList<Student> getStudents()
    {
        return this.listGrades;
    }
}
```

Database

All store/read operations made in the project are always related to a postgresql as database and used the driver we were provided.

The tables used follow the next structure:



In the first steps all classes included some uppercase characters, but the prepareStament did not manage so well with them, seemed like the string provided was forced to a non-uppercase one. So, to avoid that problems all uppercace characters were delete from all table names.

SQL

Examgrades

```
CREATE TABLE public.examgrades
(
    gradeid integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    idstudent integer,
    idexam integer,
    grade integer,
    CONSTRAINT examgrades_pkey PRIMARY KEY (gradeid),
    CONSTRAINT "examsFK" FOREIGN KEY (idexam)
        REFERENCES public.exams (idexam) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.examgrades
    OWNER to postgres;
```

Exams

```
CREATE TABLE public.exams
(
    idexam integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    description text COLLATE pg_catalog."default" NOT NULL,
    examdate date NOT NULL,
    examtime time without time zone NOT NULL,
    location text COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "EXAMS_pkey" PRIMARY KEY (idexam)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.exams
    OWNER to postgres;
```

Users

```
CREATE TABLE public.users
(
    iduserpk integer NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 2147483647 CACHE 1 ),
    iduser integer NOT NULL,
    isprofesor boolean NOT NULL,
    CONSTRAINT users_pkey PRIMARY KEY (iduserpk)
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.users
    OWNER to postgres;
```

Client

To be able to test the different services and check how they really work a python small program to manage the different options was made. It allow us to use the different gets / posts options and check if them retrieve the correct info from the database connected to the web service

GitHub

All the documents and code related to this project are available on the next github repository:

- <https://github.com/ricardPlana/WSPProject>

Outputs and executions

Menus

Our client is divided into sections, one for the professor/admin user and one from the students, this user must be created before from our system administrator directly on the database.

If we try to join in using a non-registered we will receive a message saying we must contact the admin to be registered in.

```
Insert ID: 852  
User not found, contact ur administrator
```

Professor menu

```
Insert ID: 123  
  
****OPTIONS****  
*** (a) - Read ALL Exams ***  
*** (b) - Insert Exam ***  
*** (c) - Modify Exam ***  
*** (d) - Delete Exam ***  
*** (e) - Read ONE Exam ***  
*** (f) - Insert grade/s ***  
*** (g) - Download all exams ***  
*** (h) - Download one exam ***  
*** (i) - Download all grades from an exam ***  
*** (j) - Download all grades from a student ***  
*** (q) - CLOSE PROGRAM ***  
Please select one of the above options: |
```

Student menu

```
Insert ID: 456  
  
****OPTIONS****  
*** (a) - Read ALL Exams ***  
*** (b) - Read ONE Exam ***  
*** (c) - Download all exams ***  
*** (d) - Download one exam ***  
*** (e) - Download all grades from a student ***  
*** (q) - CLOSE PROGRAM ***  
Please select one of the above options: |
```

Student

Read all exams (common in both menus)

Our first option is the read all exam, this GET will only provide us with all the exams in the DB via console.

```
Please select one of the above options: 0

***EXAM***
idexam":"2"
"description":"randomDesc"
"examdate":"2021-06-17"
"examtime":"17:00:00"
"location":"lleida"

***EXAM***
"idexam":"3"
"description":"randomDesc"
"examdate":"2021-06-17"
"examtime":"17:00:00"
"location":"lleida"
```

Read one exam (common in both menus)

We added different options while reading only one exam to be exact, by id, by a partial description and by full description.

Id

First of all we will be required to insert an id, and the exam will be write directly on the console.

```
Please select one of the above options: 0
Do u want to search it by id or description partial or full? [id/descp/descf/exit] id
Insert id to find: 9

EXAM VALUES:

***EXAM***
idexam":"9"
"description":"randomDesc"
"examdate":"2021-06-17"
"examtime":"17:00:00"
"location":"lleida"
```

Partial description

After providing a partial description, the service will provide us from all the different exams than can fit in the description provided.

```
Please select one of the above options: 0
Do u want to search it by id or description partial or full? [id/descp/descf/exit] descp
Insert partial desc to find: random
```

```
***EXAM***
"idexam":"23"
"description":"randomDesc"
"examdate":"2021-06-17"
"examtime":"17:00:00"
"location":"lleida"

***EXAM***
"idexam":"25"
"description":"random"
"examdate":"2021-06-17"
"examtime":"17:06:00"
"location":"madrid"
```

Full description

If we choose the full description, we will only receive the different exams where the description it's exactly the one provided.

```
Please select one of the above options: 0
Do u want to search it by id or description partial or full? [id/descp/descf/exit] descf
Insert full desc to find: random

EXAM VALUES:

***EXAM***
idexam:"25"
"description":"random"
"examdate":"2021-06-17"
"examtime":"17:06:00"
"location":"madrid"
```

Download all exams (common in both menus)

A dialog will open to select file name and path. After this the file will be saved directly in the provided path.

```
Please select one of the above options: c
```

Archivo Edición Formato Ver Ayuda

```
"idexam":"8" "description":"randomDesc" "examdate":"2021-06-17" "examtime":"17:00:00" "location":"lleida"}
{"idexam":"9" "description":"randomDesc" "examdate":"2021-06-17" "examtime":"17:00:00" "location":"lleida"}
{"idexam":"25" "description":"random" "examdate":"2021-06-17" "examtime":"17:06:00" "location":"madrid"}
```


Download one exams (common in both menus)

Following the same path as we did in the read one exam, we will be adding the 3 different options and providing in each one the expected result. And we will be asked as before where we want to store the new file.

ID

```
Please select one of the above options: d
Do u want to search it by id or description partial or full? [id/descp/descf/exit] id
Insert id to find: 9
```

```
testexam9: Bloc de notas
Archivo Edición Formato Ver Ayuda
{"idexam":"9" "description":"randomDesc" "examdate":"2021-06-17" "examtime":"17:00:00" }
```

Partial description

```
Please select one of the above options: d
Do u want to search it by id or description partial or full? [id/descp/descf/exit] descp
Insert partial desc to find: random
```

```
all: Bloc de notas
Archivo Edición Formato Ver Ayuda
{"idexam":"8" "description":"randomDesc" "examdate":"2021-06-17" "examtime":"17:00:00" "location":"lleida"}
{"idexam":"9" "description":"randomDesc" "examdate":"2021-06-17" "examtime":"17:00:00" "location":"lleida"}
{"idexam":"25" "description":"random" "examdate":"2021-06-17" "examtime":"17:06:00" "location":"madrid" }
```

Full description

```
fulldesc: Bloc de notas
Archivo Edición Formato Ver Ayuda
{"idexam":"25" "description":"random" "examdate":"2021-06-17" "examtime":"17:06:00" "location":"madrid" }
```

```
Please select one of the above options: d
Do u want to search it by id or description partial or full? [id/descp/descf/exit] descf
Insert full desc to find: random
```

Download all grades from a student (common in both menus)

We added an option to be able to download all the grades related to an student id introduced.

```
Please select one of the above options: e
Insert the student id to retrieve grades all grades: 22
```

```
student22: Bloc de notas
Archivo Edición Formato Ver Ayuda
{"studentid":"22" "grade":"2" "examid":"8"}
{"studentid":"22" "grade":"4" "examid":"8"}
{"studentid":"22" "grade":"4" "examid":"9" }
```

Quit option (common in both menus)

```
Please select one of the above options: q
Quit time
```

Professor

Insert exam

We will be asked for some parameters excluding the id related to this exam, as it will be the primary key and the database will be directly introduced by the database while doing the insert.

```
Insert exam values
Exam description: ExamTestAA
Exam date: 10/01/2021
Exam time: 18:34
Exam Location: El Palau
Exam added
```

We check if the new exam can be already seen in the database (we will be using our own functions made to check that options)

```
Please select one of the above options: e
Do u want to search it by id or description partial or full? [id/descp/descf/exit] id
Insert id to find: 28

EXAM VALUES:

***EXAM***
'idexam':"28"
'description':"ExamTestAA"
'examdate':"2021-01-10"
'examtime':"18:34:00"
'location':"El Palau
```

Modify description

While trying to modify a description we will only need to provide the system with the exam id (we can retrieve it, by using the find by partial/full description for example), and the new one description we would like to have.

```
Please select one of the above options: c
Insert exam id to modify the description: 28
Insert new description: ExamTestModified
Exam updated
```

As before we'll be checking if the values were updated.

```
Insert id to find: 28

EXAM VALUES:

***EXAM***
'idexam':"28"
'description':"ExamTestModified"
'examdate':"2021-01-10"
'examtime':"18:34:00"
'location':"El Palau
```

Delete exam

While trying to delete one exam, we will be asked to provide the id for this one, but if we try to delete one exam with grades, we will receive a message saying we can't do that.

```
*** (q) - CLOSE PROGRAM ***  
Please select one of the above options: d  
Insert exam id to be deleted: 28  
Exam deleted or already deleted
```

```
Insert exam id to be deleted: 8  
Exam have grades, can't be deleted
```

We will check if this exam was deleted


```
Please select one of the above options: e  
Do u want to search it by id or description partial or full? [id/descp/descf/exit] id  
Insert id to find: 28  
  
EXAM VALUES:  
  
exam not found
```

Insert Grades

While doing an insert we will be asked for two parameters before the students info, we will need to introduce the examid, and the number of grades we want to introduce at one in the db.

```
Please select one of the above options: f  
Insert exam id to relate the grade: 29  
Number of grades to be inserted 3  
Insert student id: 123  
Student grade 5  
Insert student id: 234  
Student grade 7  
Insert student id: 345  
Student grade 3  
Grades added / or ignored if already in
```

We check by downloading all grades from an exam that this 3 are the only ones added to the db


 exam29: Bloc de notas

```
Archivo Edición Formato Ver Ayuda  
{ "studentid": "123" "grade": "5" "examid": "29" }  
{ "studentid": "234" "grade": "7" "examid": "29" }  
{ "studentid": "345" "grade": "3" "examid": "29" }
```

Download all grades from exam

We already used this service in the previous one, we will only need to provide an examid and we will be downloading a file with all of them.

```
Please select one of the above options: 1
Insert the exam id to retrieve the grades: 29
```

 exam29: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
{ "studentid": "123" "grade": "5" "examid": "29" }
{ "studentid": "234" "grade": "7" "examid": "29" }
{ "studentid": "345" "grade": "3" "examid": "29" }
```