



**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

Projeto de Paradigmas de Programação

Licenciatura em Engenharia Informática

2022/2023

Grupo 2

André Faria -8220787

Duarte Sousa – 8220160

Listagens do projeto

- As edições que possuem projetos com submissão em falta em tarefas:

```
/**
 *
 * Lista as edições que possuem projetos com submissões em falta em tarefas.
 *
 * @return o array com as edições
 * @throws EditionsWithoutSubmissionsCompleted caso não existam submissões
 * em falta
 */
public Edition[] getEditionsWithMissingSubmissions() throws EditionsWithoutSubmissionsCompleted {
    Edition[] edtns = new Edition[editions.length];
    int index = 0;
    for (Edition ed : editions) {
        if (ed != null) {
            Project[] projects = ed.getProjects();
            for (Project proj : projects) {
                if (proj != null) {
                    if (proj.isCompleted() == false) {
                        edtns[index] = ed;
                        index++;
                    }
                }
            }
        }
    }
    if (index == 0) {
        throw new EditionsWithoutSubmissionsCompleted("Sem submissões em falta");
    } else {
        return edtns;
    }
}
```

- Os projetos com submissões em falta de uma edição e da edição ativa:

```
/**
 *
 * Lista os projetos com submissões em falta de uma edição e da edição ativa.
 *
 * @return projetos
 * @throws EditionWithoutSubmissionsCompleted
 */
public Project[] getProjectsWithMissingSubmissions() throws EditionWithoutSubmissionsCompleted {
    int index = 0;
    Project[] projs = new Project[projects.length];
    for (Project project : this.projects) {
        if (project != null) {
            if (project.isCompleted() == false) {
                projs[index] = project;
                index++;
            }
        }
    }
    if (index == 0) {
        throw new EditionWithoutSubmissionsCompleted("Nenhum projeto com submissões em falta.");
    } else {
        return projs;
    }
}
```

- O número de projetos de uma dada edição:

```
/**
 *
 * Este método permite obter o número de projetos da edição
 *
 * @return número de projetos da edição
 */
public int getNumProjects() {
    return this.numProjects;
}
```

Listagens do projeto

- O número de edições CBL existentes:

```
/**
 *
 * Obtém o numero de edições
 *
 * @return
 */
public int getNumEditions() {
    return numEditions;
}
```

- Uma representação textual do progresso de um projeto de uma edição (considerando as submissões realizadas até ao momento):

```
/**
 * Este método obtém uma representação textual do progresso de um projeto
 * @return representação textual do progresso de um projeto
 */
public String projectProgressToString(){
    int numCompleted=0;

    for(Task t: this.tasks){
        if(t!=null){
            if(t.getNumberOfSubmissions() !=0){
                numCompleted++;
            }
        }
    }

    String s="";
    s+="\nProgresso de projeto: "+numCompleted+"/"+this.numTasks+" tarefas completas";
    return s;
}
```

- Uma representação textual do progresso de uma edição (considerando as submissões realizadas até ao momento):

```
/**
 * Este método obtém uma representação textual do progresso de uma edição
 *
 * @return representação textual do progresso de uma edição
 */
public String editionProgressToString() {
    int totalTasks = 0;
    int totalTasksCompleted = 0;
    for (Project p : this.projects) {
        if (p != null) {
            totalTasks += p.getNumberOfTasks();
            Task[] tasks = p.getTasks();
            for (Task t : tasks) {
                if (t != null) {
                    if (t.getNumberOfSubmissions() != 0) {
                        totalTasksCompleted++;
                    }
                }
            }
        }
    }

    int projectsCompleted = 0;
    for (Project p : this.projects) {
        if (p != null) {
            if (p.isCompleted() == true) {
                projectsCompleted++;
            }
        }
    }

    String s = "";
    s += "\nProgresso de edição:";
    s += "\n" + projectsCompleted + "/" + this.numProjects + " projetos terminados";
    s += "\n" + totalTasksCompleted + "/" + totalTasks + " tarefas terminadas";

    return s;
}
```

Listagens do projeto

Listagens implementadas/propostas no projeto

- Edições com o maior número de projetos: permite-nos visualizar todas as edições e faz uma listagem das edições com o maior número de projetos.
É importante pois conseguimos controlar as despesas ou os recursos que temos de disponibilizar para as várias edições CBL que existem e que vão existir.

```
//////////////////////////////////listagens//////////////////////////////////
/*
 *
 * Lista as edições com o maior número de projetos
 *
 */
public void listEditionWithMoreProjects() {
    int maxNumProjects = 0;
    int topProject = 0;
    if (this.numEditions == 0) {
        System.out.println("Ainda não existem edições.");
    } else {
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < this.numEditions; j++) {
                if (i == 0) {
                    if (this.editions[j].getNumberOfProjects() > maxNumProjects) {
                        maxNumProjects = this.editions[j].getNumberOfProjects();
                    }
                } else {
                    if (this.editions[j].getNumberOfProjects() > maxNumProjects && this.editions[j].getNumberOfProjects() < topProject) {
                        maxNumProjects = this.editions[j].getNumberOfProjects();
                    }
                }
            }
            for (Edition ed : this.editions) {
                if (ed != null) {
                    if (ed.getNumberOfProjects() == maxNumProjects && maxNumProjects != topProject) {
                        System.out.println(i + 1 + " ");
                        System.out.println("\nNúmero de projetos: " + maxNumProjects);
                        System.out.println(ed.toString());
                    }
                }
            }
            topProject = maxNumProjects;
        }
    }
}
```

•

Listagens do projeto

- Edições com o maior número de participantes: permite-nos visualizar a totalidade de participantes em vários projetos que se encontrem em várias edições. Podemos assim visualizar as edições que foram mais aderidas por parte dos alunos.

```
/**
 * Lista as edições com o maior número de participantes
 */
public void listEditionWithMoreParticipants() {
    int maxNumParticipants = 0;
    int topParticipants = 0;
    int numParticipants = 0;
    if (this.numEditions == 0) {
        System.out.println("Ainda não existem edições.");
    } else {
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < this.numEditions; j++) {
                numParticipants = 0;
                for (Project p : this.editions[j].getProjects()) {
                    if (p != null) {
                        numParticipants += p.getNumberOfParticipants();
                    }
                }
                if (i == 0) {
                    if (numParticipants > maxNumParticipants) {
                        maxNumParticipants = numParticipants;
                    }
                } else {
                    if (numParticipants > maxNumParticipants && numParticipants < topParticipants) {
                        maxNumParticipants = numParticipants;
                    }
                }
            }
            for (Edition ed : this.editions) {
                numParticipants = 0;
                if (ed != null) {
                    for (Project p : ed.getProjects()) {
                        if (p != null) {
                            numParticipants += p.getNumberOfParticipants();
                        }
                    }
                    if (numParticipants == maxNumParticipants && maxNumParticipants != topParticipants) {
                        System.out.println(i + 1 + "=");
                        System.out.println("\tnúmero de participantes: " + maxNumParticipants);
                        System.out.println(ed.toString());
                    }
                }
            }
            topParticipants = maxNumParticipants;
        }
    }
}
```

Listagens do projeto

- Tarefas com submissões mais próxima á data de entrega: permite-nos visualizar a tarefa/tarefas com submissões próximas ao limite do prazo final.
É importante pois permite aos participantes verificar quais são as tarefas que estão mais perto do prazo final permitindo-lhes maior controlo e gestão sobre as próximas tarefas a realizarz

```
/**
 * Lista tarefas com data limite mais próxima
 */
public void listClosestDeadlineTasks() {
    LocalDate closestDate = LocalDate.MAX;
    LocalDate topClosestDate = LocalDate.MAX;

    for (int i = 0; i < 10; i++) {
        closestDate = LocalDate.MAX;
        for (int j = 0; j < this.numTasks; j++) {

            if (i == 0) {
                if (this.tasks[j].getEnd().isBefore(closestDate)) {
                    closestDate = this.tasks[j].getEnd();
                }
            } else {
                if (this.tasks[j].getEnd().isBefore(closestDate) && topClosestDate.isBefore(this.tasks[j].getEnd())) {
                    closestDate = this.tasks[j].getEnd();
                }
            }
        }
        for (Task t : this.tasks) {
            if (t != null) {
                if (t.getEnd().equals(closestDate) && (topClosestDate.equals(closestDate) == false)) {
                    System.out.println("\n\n" + (i + 1) + "° data limite mais próxima");
                    System.out.println("\nData limite: " + closestDate.toString());
                    System.out.println(t.toString());
                }
            }
        }
        topClosestDate = closestDate;
    }
}
```