


| | | | |
|--|--|-------------------------|------|
|  <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small> | Tipo de Prova Trabalho prático – Avaliação em período letivo | Ano letivo 2022/2023 | Data |
| | Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores | Hora | |
| | Unidade Curricular Paradigmas de Programação | Duração | |

Objetivos

Com a realização do trabalho prático, pretende-se que os alunos ponham em prática todos os conhecimentos adquiridos na utilização do Paradigma de Programação Orientado a Objetos (POO) e a sua implementação na linguagem de programação Java, demonstrando as suas competências em:

- Conhecer e compreender os conceitos fundamentais associada à POO;
- Conceber e implementar, para problemas concretos, soluções que tenham por base a POO.
- Reconhecer e compreender a semântica e a sintaxe da linguagem Java.
- Reutilizar, alterar e desenvolver código recorrendo à linguagem Java tendo em vista um determinado problema com regras semânticas específicas.

Considere ainda que:

- Não é permitida a utilização de API's/conceitos Java que não tenham sido alvo de lecionação no ano letivo corrente na unidade curricular de Paradigmas de Programação (LEI e LSIRC). Os alunos que pretendam utilizar API's adicionais devem atempadamente pedir autorização a um dos docentes da unidade curricular.
- Não é permitida a utilização de coleções Java predefinidas (*Java Collections Framework*).
- Os recursos de suporte ao trabalho referenciados no enunciado, são de utilização **obrigatória**.

Introdução

O *Challenge Based Learning* (CBL) é uma abordagem de ensino que tem como objetivo fornecer aos alunos a oportunidade de resolver problemas do mundo real de uma maneira prática e colaborativa. Através do CBL, os alunos são incentivados a trabalhar em equipa, identificar problemas e procurar soluções criativas.

Uma das principais características do CBL é o papel do facilitador. O facilitador é o responsável por guiar e apoiar os alunos durante todo o processo de aprendizagem, desde a identificação do problema até a implementação da solução. O facilitador é um mentor que incentiva os alunos a desenvolver habilidades como pensamento crítico, resolução de problemas e comunicação.

Além do facilitador, o CBL também envolve a colaboração com parceiros externos. Os parceiros podem ser empresas, organizações sem fins lucrativos, instituições de ensino superior, entre outros. Os parceiros fornecem aos alunos informações e recursos valiosos para ajudá-los a entender e resolver o problema em estudo.


O CBL também enfatiza a importância do trabalho em equipa. Os alunos são organizados em equipas para trabalhar em projetos específicos. Cada equipa é composta por alunos com diferentes habilidades e conhecimentos, o que ajuda a promover a diversidade e a criatividade.

A cada edição do CBL, vários projetos são lançados. Cada projeto é uma sequência de tarefas que devem ser submetidas pelos alunos. As tarefas são projetadas para ajudar os alunos a desenvolver habilidades específicas, como a pesquisa, a análise de dados e a prototipagem. Ao final do projeto, os alunos devem apresentar sua solução para o problema que identificaram.

Descrição técnica

A API a desenvolver deverá ser capaz de suportar a gestão de edições de CBL (*Edition*), nomeadamente os seus projetos (*Project*), as tarefas associadas (*Task*) e a submissão de trabalhos (*Submission*) nas tarefas. Deve ainda suportar a gestão de participantes (*Participant*), sejam facilitadores (*Facilitator*), parceiros (*Partner*) ou estudantes (*Student*).

Como suporte ao desenvolvimento da API, são disponibilizados um conjunto de recursos (*MA02_Resources*), de utilização obrigatória e que definem os contratos e definem o ponto de partida para o desenvolvimento da API. Os conteúdos fornecidos são um complemento ao presente enunciado, contendo informação adicional e específica sobre as particularidades de implementação.

| | | | |
|--|---|-------------------------|------|
|  <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small> | Tipo de Prova Trabalho prático – Avaliação em período letivo | Ano letivo 2022/2023 | Data |
| | Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores | Hora | |
| | Unidade Curricular Paradigmas de Programação | Duração | |

Deverá realizar a implementação do código necessário para suportar cada uma das operações definidas nos contratos. A existência dos contratos não deve ser impeditiva para a implementação de novos contratos, funcionalidades e/ou novos métodos ou classes. A utilização dos contratos constitui um ponto de partida, cujos ficheiros não podem ser alterados. **Caso não utilize os recursos disponibilizados, todo o trabalho é invalidado.**

A criação dos projetos é realizada através de *templates* nas edições. O template representa um ficheiro JSON com a informação necessária para criar os projetos e as suas tarefas. O seguinte excerto apresenta um exemplo da forma como a informação se encontra:

```
{
  "number_of_facilitators": 2,
  "number_of_students": 4,
  "number_of_partners": 16,
  "tasks": [
    {
      "title": "Long list of Stakeholder ",
      "description": "The team's initial goal is to identify up to 15 stakeholder/user groups relevant to the project topic.",
      "start_at": 0,
      "duration": 14
    },
    {
      "title": "Short list of target groups",
      "description": "Your team should discuss and select four target groups from the long list of previously identified stakeholder/user groups.",
      "start_at": 0,
      "duration": 14
    },
    (...)
  ]
}
```


O ficheiro contém:

- Número de facilitadores (*number_of_facilitators*)
- Número de estudantes (*number_of_students*)
- Número de parceiros (*number_of_partners*)

Tarefas (*tasks*), composta por título (*title*), descrição (*description*), dia de início a partir da data de início da Edição (*start_at*) e a duração em dias (*duration*)

Para além dos contratos disponibilizados, **deve ainda implementar** o(s) contrato(s) para suportar os seguintes comportamentos:

- Deverá ser possível gerir (adicionar, remover e obter) edições de CBL, assim como definir qual a edição ativa. Apenas deverá existir uma edição atualmente ativa.
- Deverá ser possível adicionar projetos à edição ativa ou a edições inativas.
- Devolver:
 - As edições que possuem projetos com submissões em falta em tarefas.
 - Os projetos com submissões em falta de uma edição e da edição ativa.
 - O número de projetos de uma dada edição.
 - O número de edições de CBL existentes;
 - Uma representação textual do progresso de um projeto de uma edição (considerando as submissões realizadas até ao momento);
 - Uma representação textual do progresso de uma edição (considerando as submissões realizadas até ao momento).
- Adicionar submissões a projetos de uma edição ativa apenas por estudantes pertencentes aos projetos respetivos.
- Cada grupo deve ainda propor e implementar, no mínimo, 3 listagens/relatórios distintas. Estas listagens/relatórios devem ser relevantes para o domínio. O principal objetivo é avaliar a compreensão do

| | | | |
|--|--|-------------------------|------|
|  <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small> | Tipo de Prova Trabalho prático – Avaliação em período letivo | Ano letivo 2022/2023 | Data |
| | Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores | Hora | |
| | Unidade Curricular Paradigmas de Programação | Duração | |

problema bem como a capacidade do grupo de analisar os dados armazenados. Estas listagens devem ser claramente identificadas no menu do programa desenvolvido.

- Deverá criar uma funcionalidade para o agendamento de eventos (identificando a data de início e de fim e o local) para edição de CBL ativa. Esta funcionalidade é obrigatória para a avaliação do trabalho. Considere que:
 - Os eventos podem ser de 3 tipos: *Kick-off meetings*, *Pitches* e *Bootcamps*.
 - Nos *Kick-off meetings* devem ser associados a um projeto e só podem participar os parceiros e os facilitadores;
 - Nos *Pitches* podem participar todos os tipos de participantes, devendo ainda ser criado um novo tipo de participante que representa o júri.
 - Nos *Bootcamps* só poderão participar os facilitadores.
 - Só podem ser adicionados eventos cuja data de início após o dia atual e compreendidos entre a data de início e de fim da CBL.
 - Não deve ser permitida a edição/remoção de eventos que já tenham iniciado;
 - Deverá ser possível listar eventos, apresentando toda a informação dos eventos, incluindo as datas, locais e participantes.
- A API deverá suportar a persistência de dados de forma que nas várias interações com o programa seja armazenada a informação modificada ou adicionada na interação anterior. A abordagem para a persistência de dados deverá ser selecionada por cada grupo. Esta funcionalidade é obrigatória para a avaliação do trabalho.
- Deverá criar os contratos/classes necessárias para a implementação de um sistema de ranking dos projetos de uma edição. Esta funcionalidade permitirá a atribuição manual de uma classificação para os vários projetos. Esta funcionalidade é obrigatória para a avaliação do trabalho.

Deve ainda criar um **programa** utilizando a API desenvolvida, considerando que a criação do projeto (e informação associada) deverá ser suportada pela utilização de um *template* (ficheiro .JSON fornecido).

Teste o mais exaustivamente possível o código que desenvolveu como resposta aos requisitos apresentados. Recorra a comentários JavaDoc e não só de modo a documentar, o mais exaustivamente possível, o código que desenvolveu.

Elaboração do trabalho

Este trabalho é realizado em grupo que deverá ser composto por, no máximo, 2 alunos da unidade curricular.

Datas e considerações


Os alunos devem comunicar o seu grupo de trabalho através da página da unidade curricular de Paradigmas de Programação na plataforma Moodle¹, até à data constante nessa plataforma.

O trabalho deve ser entregue através da página da unidade curricular de Paradigmas de Programação na plataforma Moodle até à data constante nessa plataforma.

A data de defesa do trabalho será publicada na plataforma Moodle na página da unidade curricular de Paradigmas de Programação. A defesa será realizada por turnos e a data exata para cada aluno será comunicada na plataforma moodle após a entrega do trabalho.

Considera-se por defesa satisfatória, quando o aluno demonstra que realizou o trabalho submetido e que domina todos os conceitos de programação orientada a objetos aplicados na resolução do trabalho. Tentativas de fraude, resultarão na avaliação do trabalho como: Fraude Académica.

¹ <http://moodle.estg.ipp.pt>

| | | | |
|--|---|-------------------------|------|
|  <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small> | Tipo de Prova Trabalho prático – Avaliação em período letivo | Ano letivo 2022/2023 | Data |
| | Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores | Hora | |
| | Unidade Curricular Paradigmas de Programação | Duração | |

Formato da entrega

Os trabalhos entregues deverão evitar (se possível) utilizar caminhos absolutos ou endereços específicos, de modo que possam ser facilmente utilizados em qualquer máquina. Para além disso, e no sentido de facilitar a receção dos vários trabalhos recebidos, estes deverão observar as seguintes regras:

- Todos os elementos do grupo deverão submeter o trabalho no link respetivo (Entrega do Trabalho);
- O trabalho desenvolvido deverá ser entregue através do moodle, através da submissão de um ficheiro com o nome PP_EN_<nr_do_aluno>_<nr_do_aluno>.zip, contendo:
 - Os ficheiros criados incluindo o(s) projeto(s) do IDE Netbeans e uma pasta com a distribuição (jar) da solução proposta.
- Comentários JavaDoc, e não só, de modo a documentar, o mais exaustivamente possível, o código desenvolvido.
- Um comentário, no início de cada ficheiro de código entregue por cada grupo, com pelo menos a seguinte informação (com as adaptações óbvias para cada aluno/grupo):

```
/*
 * Nome: <Nome completo do aluno>
 * Número: <Número mecanográfico do aluno>
 * Turma: <Turma do aluno>
 *
 * Nome: <Nome completo do colega de grupo>
 * Número: <Número mecanográfico do colega de grupo>
 * Turma: <Turma do colega de grupo>
 */
```

Os alunos que não realizem a entrega do trabalho até à data/hora definida serão sujeitos a penalização ou a invalidação do trabalho.