 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Avaliação durante período letivo	Ano letivo 2024/2025	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*)** nem qualquer API não lecionada ou não autorizada previamente por um dos docentes.
- Os recursos de suporte ao trabalho referenciados no enunciado, são de utilização **obrigatória**.

Introdução

A **PPStudios**, uma empresa de desenvolvimento de software na zona de Felgueiras, pretende expandir seus horizontes para o mercado de **videojogos de simulação**. Reconhecendo o potencial crescente deste nicho, a empresa decidiu desenvolver um protótipo de **simulador de gestão desportiva** para avaliar a aceitação no mercado de jogos de estratégia e gestão.

Para fomentar a criatividade e descobrir novos talentos na região, a PPStudios lançou um desafio aos jovens programadores: desenvolver um **simulador de gestão de futebol em formato textual**, aproveitando as potencialidades da biblioteca de desenvolvimento da empresa. O objetivo passa por desenvolver um programa que utilize os **recursos disponibilizados pela PPStudios** que estabelecem os contratos necessários para o **desenvolvimento da API** que será utilizada no **programa a desenvolver**.




O jogador assume o papel de um treinador de futebol que deverá assumir a orientação de uma equipa da liga portuguesa durante uma temporada, sendo responsável por escolher a estratégia da equipa e os jogadores titulares em cada jornada do campeonato. O objetivo passa por ser campeão nacional da referida temporada!

Descrição técnica da API

O simulador deverá ser desenvolvido em Java, utilizando os recursos disponibilizados pela PPStudios (disponíveis na plataforma Moodle). O sistema deverá simular uma temporada regular com jogos semanais. Uma temporada é constituída por várias equipas (**Team**) que são constituídas pelos seus jogadores (**Player**) que possuem diversos atributos de perfil e técnicos como nome, posição, remate, passe, velocidade, etc. No início do jogo é gerado o calendário (**Schedule**) para a liga (**League**), que define os jogos a realizar por cada equipa assegurando uma alternância entre jogos realizados em casa e jogos realizados fora de casa.

Antes de cada jogo, o jogador poderá selecionar a formação adotada (**Formation**) assim como os jogadores que vão jogar, associados a uma posição específica (**Squad**). Ao avançar para o jogo é realizada a simulação do jogo de acordo com uma estratégia (**MatchSimulatorStrategy**). Cada jogo possui vários eventos de jogo (por exemplo: **FOUL**, **SHOT**, **PASS**, **GOALKICK**) que podem ocorrer durante a simulação do jogo. Para cada minuto do jogo, o simulador deverá **calcular** se deve gerar ou não um evento. Cada evento possui uma **probabilidade** de ocorrer e possui uma lógica associada que deverá resultar ou não na ocorrência com **sucesso do evento**. Por exemplo, um remate à baliza poderá ser gerado e no momento da sua simulação poderão ser utilizados os **atributos de remate do jogador** e os **reflexos do guarda-redes adversário** para determinar se o remate é bem sucedido e resulta, por exemplo, num golo. No final é gerado o log de eventos assim como o resultado final do jogo da equipa do jogador e dos restantes jogos da jornada.

Devem ainda ser calculadas as **estatísticas** por jogador e por equipa, assim como a classificação do campeonato.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Avaliação durante período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores	Hora	
	Unidade Curricular Paradigmas de Programação	Duração	

A definição dos eventos do jogo, as estratégias para a sua concretização, assim como as probabilidades de ocorrência são da responsabilidade do grupo de trabalho. A PPStudios **valorizará** especialmente projetos que demonstrem criatividade nas mecânicas de jogo e fidelidade às dinâmicas do futebol real, **salvaguardando** sempre as **boas práticas de programação orientada a objetos**.

Como suporte ao desenvolvimento da API, são disponibilizados um conjunto de recursos (*PP_Resources*), de **utilização obrigatória** 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. 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.**

Desenvolvimento do programa

Um programa em linha de comandos (modo textual) que deverá utilizar a API desenvolvida, será responsável com a interação com o jogador.

Toda a lógica do jogo deverá estar na API, com o programa a servir apenas de interface para os comportamentos implementados na API.

Será valorizada uma interface intuitiva que permita através de um conjunto de menus interativos visualizar os jogadores da equipa, o calendário, as estatísticas e permita também avançar/gerar a próxima jornada do campeonato. O programa deverá apresentar de forma intuitiva a simulação de cada jogo, incluindo os eventos gerados.

Em anexo ao enunciado são disponibilizados alguns exemplos de interação. O detalhe ao nível do jogador e dos eventos é meramente ilustrativo. Através da API, cada grupo deverá definir os seus eventos e regras associadas à sua geração.

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.

Conteúdo fornecido

Como suporte ao trabalho prático, são disponibilizados documentos JSON com os dados das equipas e dos jogadores. Estes dados devem ser importados (a API escolhida fica ao critério de cada grupo) para servir de base de dados inicial para o jogo. Podem melhorar e adicionar mais informação. Deve primeiro codificar toda a lógica baseando-se nos ficheiros JSON e só depois testar com a API.

Juntamente com os recursos (*PP_Resources*) encontram-se classes (package **htmlgenerators**) que realizam a geração de páginas HTML com base nos resultados obtidos através da implementação dos contratos fornecidos. Permitindo uma visualização mais agradável e simples dos dados. O grupo de trabalho deve assegurar que estas classes recebem a informação necessária para gerar os documentos HTML.

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 terá **3 pontos de controlo obrigatórios** que implicam a submissão do trabalho prático. Em cada ponto de controlo devem submeter o trabalho desenvolvido até então, identificando, em documento próprio disponibilizado do

¹ <http://moodle.estg.ipp.pt>

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho prático – Avaliação durante período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores	Hora	
	Unidade Curricular Paradigmas de Programação	Duração	

Moodle, a contribuição de cada elemento do grupo. As datas limite de submissão encontram-se no moodle. **A não submissão de um ponto de controlo resulta na invalidação do trabalho.**

O trabalho final e pontos de controlo devem ser entregues através da página da unidade curricular de Paradigmas de Programação na plataforma Moodle até à data constante nessa plataforma. **Não são aceites trabalhos entregues depois da hora definida.**

A data de defesa do trabalho será publicada na plataforma Moodle na página da unidade curricular de Paradigmas de Programação. A defesa será realizada individualmente por cada elemento do grupo. 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 (incluindo a utilização de Inteligência Artificial na realização do trabalho), resultarão na avaliação do trabalho como: **Fraude Académica.**

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_<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.

Anexos

As imagens são ilustrativas e não refletem na totalidade as funcionalidades pedidas no trabalho.

 ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova Trabalho prático – Avaliação durante período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores	Hora	
	Unidade Curricular Paradigmas de Programação	Duração	

```
=====
PPFootball Manager v1.0 - Temporada 2024/25
=====
```

```
*** MENU PRINCIPAL ***
1. Gerir Plantel
2. Ver Calendário e Classificação
3. Preparar Próximo Jogo
4. Simular Jornada
5. Estatísticas
6. Salvar e Sair
```

Opção: 3

```
=== ESTRATÉGIA ===
Escolha a tática para este jogo:
1. Defensiva (5-3-2) - Foco em contenção
2. Equilibrada (4-4-2) - Padrão
3. Ofensiva (4-3-3) - Pressão alta
4. Personalizada...
```

Opção: 2

```
=== ESCALAR 11 INICIAL ===
(GK) 1. Hugo Oliveira (Energia: 92%)
(DEF) 2. João Pedro (Energia: 85%) | 3. Rúben Lima (78%) | 4. Neto (89%) | 5. Kiko (82%)
(MED) 6. André Leão (91%) | 7. Fábio Pacheco (88%) | 8. Pedro Ribeiro (76%)
(ATQ) 9. Steven Vitória (95%) | 10. Diogo Campos (83%)

Deseja fazer alterações? (S/N): S
Substituir jogador (número): 8
Jogadores disponíveis:
14. Miguel Costa (MED, Energia 100%, Passes: 84)
16. Bruno Freitas (MED, Energia 97%, Recuperação: 79)
Escolha o substituto: 14
```

```
- Início de jogo! A equipa começa com 4-4-2.

⌚ 23' - EVENTO: Falta marcada por Miguel Costa (SLF)
- Cartão amarelo para Miguel Costa (agressividade alta)

⌚ 37' - EVENTO: Remate de Diogo Campos (SLF)!
- Defesa do guarda-redes! (Remate: 65% vs Defesa: 72%)


⌚ 45+1' - EVENTO: Canto para SL Felgueiras
- Cruzamento de André Leão... Cabeceamento de Steven Vitória!
⚽ GOLLO! SL Felgueiras 1-0 (Precisão cabeceamento: 88% vs Posicionamento GR: 63%)

[Intervalo] SLF 1-0 FCP | Estatísticas:
Remates: 5(3) - 2(0) | Posse: 54% - 46% | Faltas: 7-4

⌚ 68' - SUBSTITUIÇÃO: Entra Bruno Freitas (MED) por Miguel Costa (cansaço: 42%)

⌚ 89' - EVENTO: Contra-ataque do FC Paços!
- Passe longo... Falha defensiva de Kiko!
⚽ GOLLO! SL Felgueiras 1-1 (Erro defesa: 34% vs Finalização: 81%)

[FIM DE JOGO] SL Felgueiras 1-1 FC Paços de Ferreira
```

 ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova Trabalho prático – Avaliação durante período letivo	Ano letivo 2024/2025	Data
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de computadores	Hora	
	Unidade Curricular Paradigmas de Programação	Duração	

```

*** RESUMO DA JORNADA 5 ***
- SL Felgueiras 1-1 FC Paços de Ferreira
- FC Porto 2-0 SC Braga
- Benfica 3-1 Sporting CP

=== CLASSIFICAÇÃO ATUALIZADA ===
1. FC Porto - 15 pts
2. Benfica - 13 pts
...
7. SL Felgueiras - 8 pts

=== ESTATÍSTICAS DESTAQUE ===
Melhor jogador: Steven Vitória (7.8 rating)
Mais passes corretos: André Leão (89%)
Lesionados: Rúben Lima (2 semanas - entorse)

Pressione ENTER para continuar...

```