

ENUNCIADO DO PROJETO DE ENGENHARIA DE SOFTWARE

TÍTULO	Aplicação de suporte à gestão de horários
DESCRIÇÃO	<p>O projeto de ES a ser desenvolvido em grupo, deverá permitir a leitura de dados sobre os horários do Iscte (dados sobre as aulas e sobre as salas), representados em ficheiros CSV (<i>comma-separated values</i>). Serão disponibilizados dois ficheiros CSV de exemplo (<i>HorarioDeExemplo.csv</i> e <i>CaracterizaçãoDasSalas.csv</i>), que correspondem ao horário do 1º semestre 2022/2023 (<i>HorarioDeExemplo.csv</i>) e ao inventário de salas do Iscte (<i>CaracterizaçãoDasSalas.csv</i>).</p> <p>O grupo de trabalho deverá desenvolver uma aplicação para suporte à gestão de horários no Iscte, que permita:</p> <ol style="list-style-type: none">1. O carregamento de um horário a partir de um ficheiro CSV (<i>horario-exemplo.csv</i>) com a seguinte estrutura: <i>Curso, Unidade Curricular, Turno, Turma, Inscritos no turno, Dia da semana, Hora início da aula, Hora fim da aula, Data da aula, Características da sala pedida para a aula, Sala atribuída à aula</i>. Note que o utilizador deve poder indicar a localização dos ficheiros para carregar (ou gravar) e os ficheiros poderão existir localmente no sistema de ficheiros do computador, ou remotamente, por exemplo, na plataforma GitHub;2. Mostrar o horário e permitir navegar no horário, sob a forma de tabela, incluindo todas as colunas indicadas no horário de exemplo. Deverá também ser permitido ao utilizador esconder colunas, ordenar o horário por qualquer um dos campos da tabela, filtrar os dados por qualquer um dos campos (por exemplo, curso, UC, turno, por sala em que as aulas ocorrem, aulas entre data/hora início e data/hora fim, aulas com número mínimo ou máximo de estudantes inscritos, etc.) e por combinação (operadores E e OU) de filtros. Devem ser adicionadas à tabela as colunas <i>Semana do ano</i> (1 a 52) e <i>Semana do semestre</i> (1 a 15) correspondentes às semanas em que as aulas ocorrem;3. Mostrar e permitir navegar no cadastro de salas do Iscte, incluindo todas as colunas indicadas no cadastro de salas de exemplo, bem como poder filtrar salas pelas características das salas (por exemplo, tipo, capacidade, localização, etc.), por critérios temporais de ocupação ou disponibilidade da sala (por exemplo, disponibilidade de salas num determinado período, indicando data/hora início e data/hora fim), por combinação de características das salas e de critérios temporais (operadores E e OU);4. A aplicação deverá permitir gravar (em CSV e JSON) o horário que foi alterado pelas ações de edição direta do utilizador nas tabelas disponíveis na interface gráfica da aplicação, ou pelas alocações automáticas da aplicação, solicitadas pelo utilizador (indicadas nas alíneas 5 e seguintes);5. Sugerir <i>slots</i> para alocação de uma aula de substituição, indicando o utilizador a aula a mudar. O software deverá perguntar as regras a seguir para a marcação da aula de substituição, por exemplo, períodos que devem ser excluídos para alocação da aula (por exemplo, manhãs, tardes, noite, segundas-feiras 10h-12h, etc.), períodos em que é possível alocar a aula (por exemplo, no mesmo dia, na mesma semana, entre duas datas/horas), a preferência dos tipos de sala (ou salas específicas) para a aula de substituição (o utilizador deverá poder manifestar interesse por mais que um tipo de sala, ou sala, por ordem de preferência), a indicação dos tipos de salas (ou salas específicas) que não são aceitáveis para a aula de substituição (por exemplo BYOD ou OS08), entre outros critérios que o grupo entenda que o software deve ter;6. Sugerir <i>slots</i> para alocação das aulas de uma UC, indicando o utilizador a UC a alocar no horário. O software deverá perguntar as regras a seguir para a marcação das aulas da UC, por exemplo, numero de aulas da UC que devem ser marcados no horário, períodos que devem ser excluídos para alocação da aula (por exemplo, manhãs, tardes, noite, segundas-feiras 10h-12h, etc.), períodos em que é possível alocar a aula (por exemplo, na mesma semana, uma aula em cada semana, num período entre data/hora início e data/hora fim), a preferência dos tipos de sala (ou salas específicas) para a aula de substituição (o utilizador deverá poder manifestar interesse por mais

	<p>que um tipo de sala, ou sala, por ordem de preferência), a indicação dos tipos de salas (ou salas específicas) que não são aceitáveis para a aula de substituição (por exemplo 0S08), entre outros critérios que o grupo entenda relevantes para o software a desenvolver;</p> <p>7. A sugestão dos <i>slots</i> produzidos pela aplicação, descritos nas alíneas 5 e 6, deve ser visualizada de forma similar à indicada na alínea 2, devendo ser adicionadas à tabela as colunas, informação e formatação necessárias, para que sejam facilmente distinguidas as sugestões produzidas, em relação à informação previamente existente dos horários. O utilizador deverá poder escolher na tabela as sugestões que entender serem as mais adequadas, eliminar as sugestões que entenda desadequadas e acrescentar manualmente as suas próprias alternativas;</p> <p>8. A aplicação deverá permitir visualizar de forma gráfica as relações de conflitualidade entre aulas (aulas que não podem ocorrer em simultâneo), recorrendo, por exemplo, a gráficos do tipo <i>network graph diagram</i> (as aulas são representadas por nós do grafo e as arestas devem ligar aulas/nós que não podem ocorrer em simultâneo), ou alternativamente, recorrendo a <i>chord diagrams</i>. Deverá ser permitido ao utilizador definir filtros (curso, UC, período entre datas, etc.), para as aulas em que pretende analisar as relações de conflitualidade;</p> <p>9. A aplicação deverá permitir visualizar de forma gráfica o mapa de ocupação das salas, recorrendo, por exemplo, a gráficos do tipo <i>heatmap</i>, tendo em consideração as horas letivas do Iscte (8h00-22h30) no eixo vertical, os dias da semana ou os dias do mês (à escolha do utilizador) no eixo horizontal, e a indicação da quantidade de salas disponíveis (ou ocupadas) nas células do <i>heatmap</i>. Deverá ser permitido ao utilizador definir filtros relacionados com as características das salas (por exemplo, Laboratório de Informática, capacidade superior a 25 lugares, etc.), bem como o período em que pretende analisar a disponibilidade (ou ocupação) das salas (por exemplo, 1/Março/2024 a 30/Maio/2024).</p>	
ATIVIDADES ENVOLVIDAS E SUAS TECNOLOGIAS	ATIVIDADE	TECNOLOGIAS
	Ambiente de programação / IDE	À escolha de cada grupo
	Linguagens de programação	À escolha de cada grupo
	Gestão de configurações / controlo de versões no repositório local	SCM Git
	Gestão de configurações / controlo de versões no repositório remoto para suporte ao trabalho colaborativo e entrega do projeto	GitHub.com
	Gestão das dependências de bibliotecas que forem usadas	À escolha de cada grupo
	Avaliação da qualidade do software produzido, a partir das métricas extraídas do código	À escolha de cada grupo
	Testes e avaliação de cobertura de testes	JUnit ou ferramentas similares à escolha do grupo
	Documentação do software produzido	JavaDoc ou ferramenta similar à escolha do grupo
	Gestão do projeto segundo abordagem Scrum, com rastreio entre os cartões das <i>user stories</i> e os eventos respetivos no GitHub (<i>commits</i> , <i>pull requests</i>)	Trello com Power-Up do GitHub
ELEMENTOS A ENTREGAR (<i>Deliverables</i>)	<p>Os elementos a disponibilizar na entrega do projeto são:</p> <ul style="list-style-type: none"> • Repositório do projeto na plataforma GitHub, com identificador <i>ES-2023-24-2Sem-Identificador_do_Grupo_No_Moodle</i> (por exemplo, o grupo identificado no Moodle como Terça-Feira-LEIPL-Grupo-A, deve criar um repositório com a identificação <i>ES-2023-24-2Sem-Terça-Feira-LEIPL-Grupo-A</i> na plataforma GitHub); • A pasta raiz deve conter um ficheiro de texto "README.md" com a identificação do grupo, identificação dos membros do grupo (nome, número de estudante e <i>username</i> no GitHub), identificação e descrição de erros e funcionalidades não implementadas ou incompletas; • Deve ser associado o rótulo <i>GestaodeHorarios-1.0</i> ao último <i>commit</i> feito no <i>branch main</i> do repositório no GitHub; • Relatório de avaliação da qualidade do software do projeto gerado pela ferramenta escolhida pelo grupo, disponibilizado no repositório GitHub; • Documentação do software produzido (JavaDoc ou ferramenta similar à escolha do grupo), disponibilizado no repositório GitHub; • Bateria de testes, sem falhas, com uma cobertura de funcionalidades do software produzido superior a 50% em 2 métricas de cobertura de testes (uma delas sendo obrigatoriamente a complexidade ciclomática); 	

	<ul style="list-style-type: none"> Evidências da gestão ágil do projeto segundo a abordagem Scrum, nomeadamente: <ul style="list-style-type: none"> representação das <i>user stories</i>, <i>sprints</i> e cerimónias; elementos relativos ao planeamento de cada <i>sprint</i> (o quê, quem e quando); evolução do <i>sprint</i> (<i>burndown chart</i>) e rastreio entre os cartões relativos às <i>user stories</i> com os eventos na plataforma GitHub (<i>commits</i>, <i>pull requests</i>).
MÉTODO DE AVALIAÇÃO	<p>O projeto prático será avaliado de acordo com os seguintes parâmetros de avaliação.</p> <p>Apresentação de Progresso do Projeto (7.5% de peso na nota final da UC, 18-22/Março/2023)</p> <ul style="list-style-type: none"> 20%, correta utilização do Git e organização do repositório do grupo na plataforma GitHub; 35%, correta adoção da abordagem Scrum na gestão e condução do projeto, com os correspondentes registos na plataforma Trello, ligados ao GitHub; 20%, implementação dos requisitos indicados no ponto 1 e 2 da descrição do projeto (pelo menos esses); 15%, testes, sem falhas, com uma cobertura de testes do software produzido superior a 50% em 2 métricas de cobertura de testes (uma delas sendo obrigatoriamente a complexidade ciclomática); 10%, qualidade técnica do trabalho (documentação, qualidade do código, adoção de boas práticas de programação). <p>Apresentação de Progresso do Projeto (7.5% de peso na nota final da UC, 22-26/Abril/2024)</p> <ul style="list-style-type: none"> 20%, correta utilização do Git e organização do repositório do grupo na plataforma GitHub; 35%, correta adoção da abordagem Scrum na gestão e condução do projeto, com os correspondentes registos na plataforma Trello, ligados ao GitHub; 20%, implementação dos requisitos indicados no ponto 3 e 4 da descrição do projeto (pelo menos esses); 15%, testes, sem falhas, com uma cobertura de testes do software produzido superior a 50% em 2 métricas de cobertura de testes (uma delas sendo obrigatoriamente a complexidade ciclomática); 10%, qualidade técnica do trabalho (documentação, qualidade do código, adoção de boas práticas de programação). <p>Apresentação Final do Projeto (35% de peso na nota final da UC, 13-17/Maio/2024)</p> <ul style="list-style-type: none"> 15%, correta utilização do Git e organização do repositório do grupo na plataforma GitHub; 30%, correta adoção da abordagem Scrum na gestão e condução do projeto, com os correspondentes registos na plataforma Trello, ligados ao GitHub; 30%, implementação das funcionalidades da aplicação de acordo com os requisitos do enunciado; 15%, testes, sem falhas, com uma cobertura de testes do software produzido superior a 50% em 2 métricas de cobertura de testes (uma delas sendo obrigatoriamente a complexidade ciclomática); 10%, qualidade técnica do trabalho (documentação, qualidade do código, adoção de boas práticas de programação). <p>A avaliação do trabalho será realizada sobre o projeto como um todo, mas as classificações serão individualizadas, com base no desempenho das apresentações (de progresso e final do projeto) e nas atividades planeadas (atribuídas) a cada membro da equipa e as efetivamente realizadas, tal como registado nas ferramentas Trello e GitHub, as quais permitem a verificação cronológica e auditoria do progresso de cada projeto.</p> <p>Os projetos serão objeto de comparação para efeitos de deteção de plágio. A deteção de práticas fraudulentas levará ao desencadear das sanções previstas nos regulamentos do Iscte.</p>