

Trabalho Prático

O presente trabalho prático tem como objetivo desenvolver as competências dos estudantes na escrita de soluções web modernas. Leia atentamente todo o enunciado e contacte os docentes da disciplina em caso de dúvidas.

- Cada **grupo de trabalho** deve ter **exatamente 2 alunos, inscritos no mesmo horário**.
 - Os alunos que pretendam realizar o trabalho prático durante a avaliação distribuída, deverão comunicar a intenção via Moodle, dentro do prazo, em formulário para o efeito. A não comunicação atempada, implica a realização da componente prática via exame.
 - Para submeter o trabalho, os alunos deverão escolher um grupo atempadamente, **antes de efetuar a submissão do trabalho no Moodle**.
 - Os alunos que pretendam realizar o trabalho sozinhos, deverão comunicar antecipadamente a intenção com o docente da disciplina. Apenas razões excecionais serão consideradas e aprovadas, nomeadamente a existência de estatutos especiais, como trabalhador estudante.
 - **Não existirão exceções** relativamente aos estudantes do mesmo grupo terem de pertencer ao mesmo horário.
 - A entrega de trabalho com um grupo que não respeite as regras não será considerado.
- Cada trabalho prático deverá seguir os requisitos de um dos temas abaixo descritos (A, B, C ou D), assim como os requisitos gerais.
- Distribuição de temas:
 - A1-A5; BC1-BC5; D1-D5 – Tema A
 - A6-A10; BC6-BC10; D6-D10 – Tema B
 - A11-A15; BC11-BC15; D11-D15 – Tema C
 - Restantes grupos – Tema D

Regras

- O trabalho deverá ser desenvolvido utilizando as seguintes tecnologias:

- Framework: .NET Core 7
 - Frontend: Blazor
 - Backend: Web API
 - Base de dados: PostgreSQL
 - Acesso à base de dados: Entity Framework
- Recomenda-se a utilização do Rider como IDE, mas os alunos podem desenvolver o trabalho numa outra IDE ou editor de código. Caso o projeto não seja compilável e executável a partir do Rider, os alunos deverão indicar instruções sobre como compilar e executar o código.
- A implementação de funcionalidades extra não presentes no enunciado será valorizada, desde que estas funcionalidades não modifiquem os requisitos obrigatórios e não reduzam a dificuldade do trabalho. As funcionalidades extra implementadas deverão ser indicadas na apresentação do trabalho.
- Durante as aulas práticas, serão disponibilizados exemplos de bases de código para os alunos poderem trabalhar sobre a mesmas, de forma a facilitar o desenvolvimento.
- A apresentação de implementações **não originais e que constituam plágio** conduzem à imediata atribuição de **nota zero** no trabalho de grupo e a **eventuais processos disciplinares**.

Avaliação e Entrega

- O trabalho prático faz parte da avaliação da Componente Prática da disciplina de **Engenharia de Software II** (correspondendo a 70% da nota final).
- Para aprovação à disciplina, a nota da Componente Prática deverá ter a classificação mínima obrigatória de **7.5 valores**.
- O trabalho prático deverá ser submetido através do Moodle seguindo as instruções lá indicadas. A entrega deverá conter os seguintes elementos:
 - Código fonte em C# com solução constituída por:
 - Projeto com backend em Web API
 - Projeto com frontend em Blazor

- Projeto com testes unitários
 - Scripts para criação da base de dados
 - **A entrega do código do projeto deverá ser feita via GIT (GitHub)**, através de um repositório privado partilhado com o docente da disciplina, através do seu e-mail institucional (luisteofilo@estg.ipvc.pt).
- Vídeo com duração máxima de 5 minutos com instruções de utilização e demonstração de funcionalidades;

Tema A – Sistema de rastreio de tempo em tarefas

Este projeto consiste no desenvolvimento de um sistema de rastreio de tarefas para os *freelancers* poderem anotar o trabalho por eles realizado diariamente, assim como cobrar o valor adequado aos seus clientes.

1. Um utilizador pode criar uma conta. Para aceder ao sistema, o utilizador deverá efetuar login.
2. O utilizador pode editar os dados da sua conta, nomeadamente o seu nome, dados de autenticação e número habitual de horas de trabalho diário.
3. O utilizador poderá criar um projeto para agrupar tarefas. Um projeto tem um nome, nome do cliente e preço por hora. As informações sobre o projeto poderão ser alteradas. Os projetos podem ser removidos, tendo o utilizador a opção de apagar todas as tarefas associadas ou simplesmente desassociar as tarefas.
4. Um utilizador pode iniciar uma tarefa indicando uma curta descrição e data/hora de início. Alternativamente, se não for indicada data e hora, considera-se que a tarefa inicia no momento atual. As tarefas podem ter um projeto associado e um preço hora (por defeito, o preço hora desse projeto).
5. Um utilizador pode terminar uma tarefa em determinada data e hora. Se não for indicada uma data e hora, considera-se o momento atual.
6. Um utilizador pode remover tarefas em curso ou finalizadas.
7. O utilizador pode listar todas as tarefas em curso, obtendo informação do tempo total despendido até ao momento.

8. O utilizador pode listar todas as tarefas finalizadas entre duas datas.
9. Um utilizador pode convidar outro utilizador para participar num projeto. O utilizador que é convidado para um projeto deverá aceitar o convite para poder introduzir tarefas nesse projeto.
10. O utilizador criador do projeto pode remover utilizadores convidados dos seus projetos.
11. Deverá ser possível visualizar um relatório pessoal para determinado mês, que deverá incluir lista de tarefas realizadas em cada dia (com indicação do projeto), número de horas total em cada dia e número de horas totais no mês. Caso alguma das tarefas tenha um preço hora associado, deverá ser indicado o preço total por dia e preço total do mês. As tarefas em curso não deverão ser incluídas no relatório. Os dias em que o utilizador excedeu o número habitual de horas diárias devem ser sinalizados no relatório.
12. Deverá ser possível visualizar um relatório mensal semelhante ao descrito na linha acima, mas ao nível do projeto e ao nível do cliente, indicando adicionalmente todos os utilizadores que executaram as tarefas a cada dia.

Tema B – Gestor de ativos financeiros

Este projeto consiste no desenvolvimento de um sistema de rastreio dos ativos financeiros pessoais, tais como depósitos a prazo, imóveis arrendados, juros de produtos financeiros de risco (ações, fundos, etc.).

1. Um utilizador pode criar uma conta. Para aceder ao sistema, o utilizador deverá efetuar login.
2. O utilizador pode inserir, apagar e atualizar os seus ativos financeiros. Considere 3 tipos de ativo financeiro: depósito a prazo, fundo de investimento e imóvel arrendado. Um ativo financeiro tem uma data de início e duração (em meses). Cada ativo tem uma percentagem de imposto (positiva, a indicar pelo utilizador) que afeta o lucro obtido por esse ativo.

3. Um imóvel arrendado tem uma designação, localização (ou morada), valor do imóvel, valor da renda, valor mensal do condomínio, e valor anual estimado de outras despesas com o imóvel.
4. Um depósito a prazo tem um valor, um banco, número de conta, titulares e taxa de juro anual. Considere que o juro é pago mensalmente, isto é, que o valor do depósito aumenta a cada mês, sendo que os juros do mês seguinte tem em conta o novo valor do depósito.
5. Um fundo de investimento tem um nome, montante investido e uma taxa de juro por defeito. Considere que pode atribuir uma taxa de juro específica a cada mês.
6. Considere que a taxa de juro pode ser negativa e que os impostos só são devidos a cada ano em caso do ativo financeiro der lucro.
7. Permitir ao utilizador pesquisar por ativos financeiros através de nome e tipo. No caso dos fundos de investimento e depósitos, o utilizador poderá pesquisar por montante aplicado.
8. O utilizador pode listar todos os ativos financeiros que estão ativos na data/hora atual, ordenados decrescentemente por valor inicialmente aplicado nesse ativo.
9. O utilizador poderá obter um relatório que contém todos os ativos financeiros que estavam ativos entre 2 datas. O relatório deverá conter o lucro total (antes e depois de impostos) e o lucro mensal médio (antes e depois de impostos) para todos os ativos desse período.
10. O utilizador poderá obter um relatório que lista todos os pagamentos de impostos e respetivo ativo (ao nível mensal) para os ativos financeiros disponíveis em determinado intervalo de tempo.
11. Um administrador poderá obter um relatório com o valor total depositado em cada banco entre duas datas. Deverá indicar igualmente o custo total dos juros pagos nesse período por cada banco.

Tema C – Plataforma de gestão de talentos IT

Este projeto consiste na implementação de uma plataforma de gestão de currículos de profissionais IT, nomeadamente *developers*, *designers*, *product managers* e *project managers*.

1. Um utilizador pode criar uma conta. Para aceder ao sistema, o utilizador deverá efetuar login.
2. Um utilizador pode criar e editar *skills* (*React*, *C++*, *TimeManagement*, etc.), indicando um nome e seleccionando a área profissional (*developer*, *design*, ...). Uma *skill* só pode ser apagada se não estiver atualmente associada a nenhum profissional. As *skills* são partilhadas entre todos os utilizadores do sistema.
3. Um utilizador pode criar um perfil de um talento indicando o nome, país, e-mail, e preço por hora. Os perfis podem ser públicos ou privados, isto é, visíveis ou não para todos os utilizadores do sistema.
4. Um utilizador pode criar um cliente na plataforma ao qual pode apresentar determinado talento.
5. Cada perfil de talento deverá ter várias *skills*, indicando o número de anos de experiência para cada *skill*.
6. A cada perfil poderá ser adicionado detalhe de uma determinada experiência, indicando um título, nome da empresa, ano de começo e ano de término (opcional, caso o talento ainda trabalhe nessa empresa). Considere que não pode haver sobreposição de experiências no mesmo ano.
7. Permitir pesquisar talentos por uma combinação de *skills*. Os resultados desta pesquisa deverão ser ordenados por nome de talento.
8. Um utilizador pode registar propostas de trabalho para os seus clientes. A proposta de trabalho deverá ter um nome, categoria de talento, *skills* necessárias, número mínimo de anos de experiência por *skill* e indicação do número total de horas e descrição do trabalho. As propostas de trabalho podem ser atualizadas ou removidas.
9. É possível listar, para uma proposta de trabalho, todos os talentos existentes no sistema que são elegíveis para essa proposta, ordenados por valor total.
10. Permitir obter um relatório com preço médio mensal (considere um mês como 176 horas) por categoria de talento e por país.
11. Permitir obter um relatório com preço médio mensal (considere um mês como 176 horas) por *skill*.

Tema D – Gestão de Eventos e Participantes

Este projeto consiste no desenvolvimento de um sistema de gestão de eventos e participantes, permitindo que os organizadores criem e gerenciem eventos, enquanto os participantes se registram e gerenciam suas inscrições.

1. Um utilizador pode criar uma conta. Para aceder ao sistema, o utilizador deverá efetuar login.
2. Os utilizadores podem ser organizadores de eventos ou participantes.
3. Um organizador pode criar um evento com informações como nome do evento, data, hora, local, descrição, capacidade máxima de participantes e preço do ingresso. Poderão existir vários tipos de ingresso no mesmo evento, sendo necessário indicar a quantidade de ingressos disponíveis para cada tipo (exemplo: entrada Normal, entrada VIP, etc).
4. Os organizadores podem editar informações do evento e adicionar ou remover atividades dentro do evento, incluindo detalhes como nome da atividade, data, hora e descrição.
5. Os participantes podem pesquisar eventos por nome, data, local e categoria.
6. Os participantes podem se registrar em eventos e escolher atividades específicas dentro do evento, se existirem.
7. Os participantes podem editar suas informações pessoais, como nome, e-mail e número de telefone.
8. Os participantes podem cancelar sua inscrição em eventos e atividades específicas.
9. Os organizadores podem ver a lista de participantes registrados em seus eventos e em atividades específicas.
10. Os organizadores podem enviar mensagens e atualizações aos participantes registrados em seus eventos.

11. Deverá ser possível visualizar um relatório geral de eventos, incluindo informações como número de eventos por categoria, eventos mais populares e número total de participantes.
12. Deverá ser possível visualizar um relatório específico para cada evento, incluindo informações como número de participantes por atividade, receita gerada e feedback dos participantes.

Requisitos gerais (para todos os temas)

1. Criar 3 níveis de autenticação no sistema: *User* (pode aceder aos seus registos ou a registos consigo partilhados); *UserManager* (pode criar novos utilizadores, editar permissões na aplicação e visualizar todos os registos); *Admin* (pode realizar qualquer ação no sistema; criar/editar utilizadores (de todos os tipos), alterar permissões e visualizar todos os registos disponíveis no sistema). Quando a aplicação inicia pela primeira vez (ainda sem utilizadores), deverá automaticamente criar um único utilizador do tipo *Admin* com um nome de utilizador e password conhecidos.
2. Todos os dados da aplicação deverão ser persistidos em base de dados, utilizando a Entity Framework.
3. A aplicação deverá ter uma interface web simples. Não será avaliado o *look & feel* da aplicação, apenas as funcionalidades.
4. Os dados introduzidos na base de dados devem ser validados (exemplo: uma data de fecho de uma tarefa tem de ser posterior à data de início).
5. Estruturar o backend usando padrões de arquitetura, nomeadamente o Repositório.
6. Utilização de Design Patterns.