

Licenciaturas em Engenharia Informática
Programação Web
Ano Lectivo de 2025/2026 – 1º Semestre
TRABALHO PRÁTICO A REALIZAR EM GRUPO DE 2 ALUNOS
TODAS AS ÉPOCAS ORDINÁRIAS
TEMA B

E

1. Introdução

Como trabalho prático a desenvolver em grupo de dois alunos para acesso ao exame de qualquer Época Ordinária, Normal ou Recurso ou Especial, de 2025/26, pretende-se que os alunos criem uma aplicação designada por **MyCOLL** e que será uma aplicação *frontend* de uso “público” que poderá ser executada através de uma interface *Web*, de um device *Android*, *iOS*, *macOS* ou *Tizen*, e que se destina à listagem ou venda consoante o caso, de peças de colecionáveis, tais como moedas, selos, crteiras de fósforo, pacotes de açucar outros colecionáveis, novos ou usados, bem como complementos para os colecciónáveis em venda. O código a desenvolver para esta aplicação “*frontend*” é um código único e desenvolvido utilizando um só projecto *Visual Studio* para qualquer uma das interfaces, aplicações, a desenvolver e atrás referidas.

Esta aplicação *frontend* tem de ser complementada por uma outra aplicação de *Gestão de Loja*, de uso exclusivo por administradores e gestores da empresa proprietária da aplicação e que disponibiliza esses colecionáveis e complementos, e que servirá para que esta empresa faça a gestão dos mesmos, respectivas categorias, modos de disponibilização dos mesmas, etc., bem como gerir os utilizadores quer os que irão adquirir ou comprar, “*Clientes*” quer os proprietários que os irão vender ou alugar, “*Fornecedores*”, e ainda os Gestores e Administradores das aplicações. Deverá permitir ainda a gestão das compras de produtos feitas na aplicação *frontend* pelos clientes.

Como se depreende do que atrás foi referido, a empresa proprietária desta plataforma, a desenvolver, não é necessariamente a proprietária dos colecionáveis para listagem ou compra bem como não é a proprietária dos complementos para venda. Assim, esta empresa através da plataforma a desenvolver é somente uma intermediária através da plataforma entre os *fornecedores* e os *clientes*. Por cada produto vendido, a empresa ganha uma percentagem do *valor base*, e a soma do *preço base* com o valor correspondente a esta *percentagem* constitui o *preço final* de venda. Assim, os produtos terão um *preço base*, definido pelo fornecedor aquando da inserção de um produto e um *preço final*, obtido pela adição ao preço base de uma percentagem definido pelo administrador ou funcionário aquando da validação do registo do produto inserido ou editado na plataforma e em que esse produto também passará do *estado inactivo* para *ativo*.

Tem de ser ainda criada uma outra aplicação API RestFull, que permitirá o acesso, controlado, aos dados de uma base de dados por parte da aplicação *frontend* “pública”. A base de dados desta *API*, será a mesma com a qual a aplicação de gestão de loja interage, isto é, os produtos, categorias e etc são os mesmos e estão por isso numa só Base de Dados. A aplicação “*frontend*”

irá solicitar dados a esta *API* que os fornecerá. A aplicação de gestão da loja, interagirá directamente com esta Base de Dados e não através desta *API* ou de outra.

A aplicação “*frontend*“ a desenvolver terá um único código e esse código de programação da mesma irá ser utilizado quer para gerar a “página” *Web*, quer para gerar as aplicações em código nativo dos devices, *Windows Machine, Android, iOS, macOS, Tizen*.

Qualquer uma das três aplicações referidas, utilizarão a autenticação e a autorização de utilizadores, em particular e obrigatoriamente usando o *.Net 8 Identity Framework* no seu todo. No caso da interacção entre a API e os frontends, em complemento à utilização do framework *Identity Core*, tem de ser utilizado também *JWT*, isto é, não utilizar somente o *framework Identity*. A mesma base de dados da aplicação já referida, também será usada para esta parte de autenticação e autorização. Os utilizadores da aplicação “*frontend*” pública só precisarão de se autenticar aquando da finalização da compra.

As soluções aplicacionais a construir devem representar uma situação real, ainda que dentro do contexto em que são desenvolvidas. As funcionalidades apresentadas neste enunciado são consideradas uma amostra primária de um conjunto de funcionalidades, no âmbito do problema, podendo ser adicionadas outras funcionalidades que sejam consideradas necessárias e úteis à definição da solução.

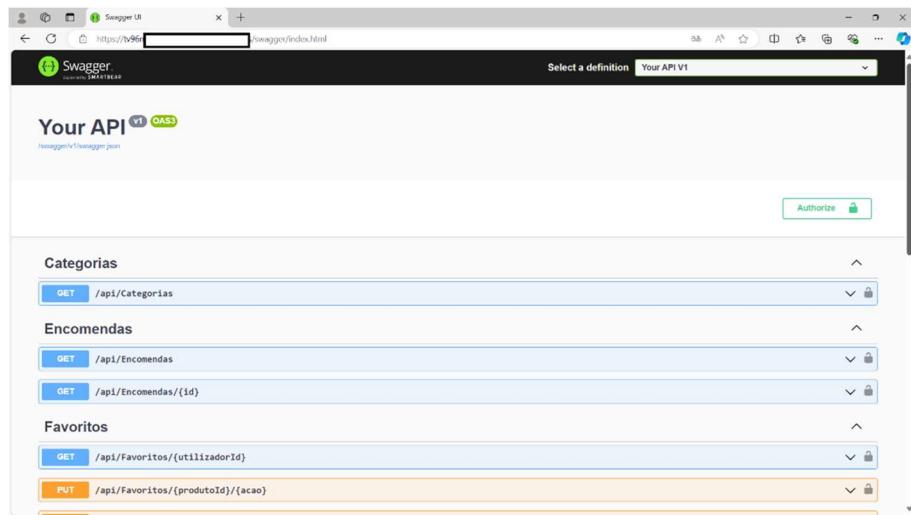
A solução final a apresentar no conjunto das três aplicações a desenvolver, deverão perspetivar uma aplicação de utilização real e prática e constituir no contexto da Engenharia o que neste contexto se entende como “*prova de conceito*”.

2. Ambiente de Desenvolvimento

É obrigatório implementar as aplicações com base no *framework .NET Core 8*, utilizando *Blazor Web, Blazor Hybrid* e *RCL*, a linguagem de programação *C# 12, LINQ, .Net Core 8 Entity Framework* e *.NET Core 8 Identity Framework, SQL Server Local DB* e utilizando o *IDE Visual Studio Community 2022*.

Em cada uma das três partes, aplicações, a desenvolver utilizar-se-ão o ou os *templates .NET 8* adequados para essa parte utilizando como exemplo o trabalho desenvolvido durante as aulas laboratoriais.

Na aplicação *API Restfull*, é obrigatório utilizar o *Swagger* na *API* para efeito de testes à mesma. Na figura seguinte, mostra-se um exemplo de um ecrã do *Swagger* da *API RestFull* a desenvolver.



Para a comunicação, na fase de desenvolvimento, entre a *API* e a aplicação *frontend*, tem de ser utilizado um *Dev Tunnel*.

3. Estruturas de Dados e Base de Dados das Aplicações

- A estrutura de dados utilizada tem de ter por base um *Modelo-Entidade-Relacionamento*, *MER*, também de entrega obrigatória, o qual deve reflectir todas as necessidades em termos de Entidades e respectivas Relações nas aplicações, concretizado nas aplicações através de classes de domínio codificadas em *C#* e que corresponderão a tabelas da base de dados criadas com recurso a instruções *migration*.

4. Regras de Negócio consideradas no desenvolvimento das aplicações

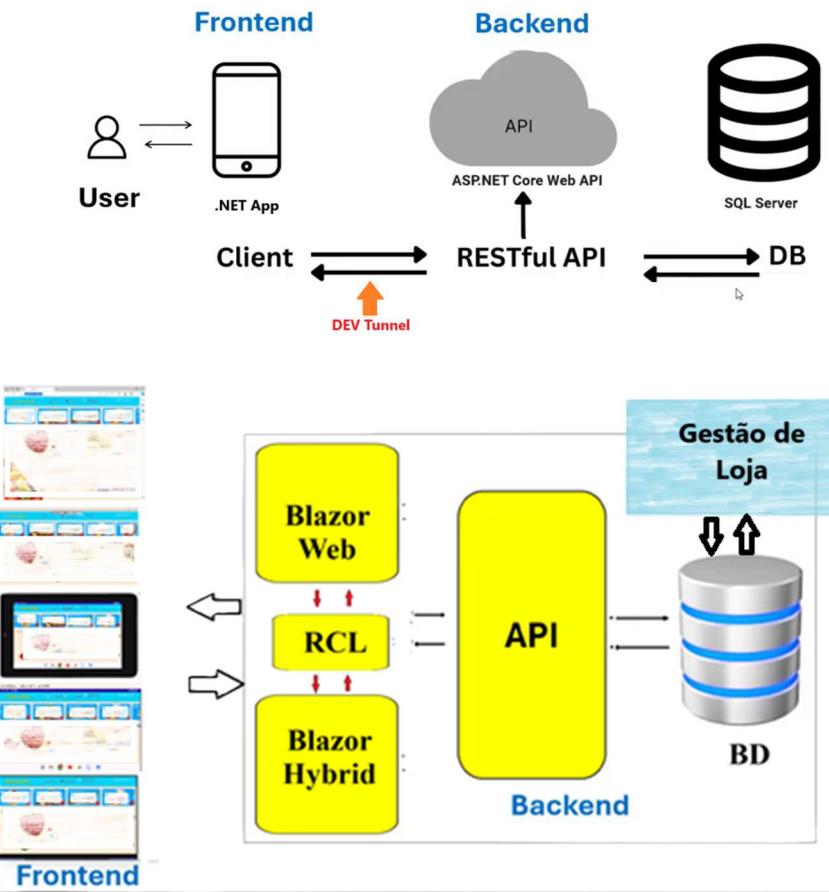
É obrigatória a especificação de um conjunto de especificações funcionais que traduzam as especificações definidas pela empresa que irá utilizar a aplicação e que constituirão as designadas “*Regras de Negócio*” a serem consideradas no desenvolvimento e em particular no funcionamento das aplicações a desenvolver.

No caso específico deste Trabalho Prático, os alunos simularão a empresa que utilizará esta aplicação e serão eles que desse modo substituirão a empresa na especificação das regras de negócio a utilizar no desenvolvimento e funcionamento das aplicações.

Estas regras de negócio são de especificação e entrega obrigatória sendo essa entrega concretizada através da sua inclusão no relatório numa secção própria.

5. Arquitectura de Desenvolvimento e Interacção entre as Partes

Como orientação para o referido, nas figuras seguintes é mostrada uma arquitectura exemplo. Na primeira figura foi excluída a aplicação de gestão de loja dadas as especificidades desta aplicação de gestão já atrás referidas.



Trabalhos práticos que não utilizem as tecnologias referidas em 2. e a arquitectura aqui indicada, serão liminarmente rejeitados e os alunos que os tenham desenvolvido sem o recurso a essas tecnologias e arquitectura estarão reprovados na componente Trabalho Prático e consequentemente na unidade curricular.

6. Interface Gráfica de Utilização Obrigatória

Será ainda obrigatório que o Trabalho Prático seja desenvolvido de acordo com o modelo e *template* exemplo indicados de seguida nas figuras, devidamente adaptados à aplicação desenvolvida, mas sem “fugir” ao esquema geral indicado.



Exemplo “Web”

Exemplo “Smartphone/Tablet”

Exemplo “Windows Machine”

Cores, imagens e outros pormenores gráficos poderão ser alterados de acordo com a sensibilidade gráfica e de *UI* do grupo que realizou o trabalho, mas sem “fugir” ao esquema geral indicado.

Poderá, caso seja necessário, ser acrescentado um outro “*slider*” para subcategorias.

Esta *UI* tem de ser a mesma em qualquer dos *devices* e “página” *Web* a usar para acesso à aplicação *frontend*, e suportada num único conjunto geral de instruções *CSS* e a serem utilizadas por todas as interfaces gráficas das aplicações de “*frontend*”, admitindo-se, no entanto, diferenças/falhas mínimas no que é apresentado em cada device ou *Web* atendendo-se a que quem realizou o trabalho prático não é um especialista em *design* gráfico. No entanto a *UI* e a *UX* deve ser bastante agradável e rica dentro das eventuais diferenças/mínimas já referidas.

Não se aceitarão trabalhos práticos desenvolvidos com base noutros modelos ou *templates* que não os exemplificados e indicados atrás e nesse caso esses eventuais trabalhos práticos serão liminarmente excluídos.

7. Descrição Genérica dos Produtos à Venda e suas classificações

Os “produtos” podem ser listados, os que não estejam à venda, ou vendidos no caso dos coleccionáveis ou dos complemento e podem ser categorizados de vários modos, por exemplo tipo, país, para alugar ou para vender, por tema, ou etc., ou doutros modos que entendam possível e adequado de ser categorizados.

Uma encomenda pode ser constituída por um ou mais produtos à venda na aplicação e também pode ser adquirido mais do que um item de cada produto à venda.

Os produtos podem ser categorizados e, eventualmente, sub-categorizados de mais do que uma maneira diferente, por exemplo: Tipo (Moedas, Selos, ...) → País → Tema → etc. Algumas destas categorias e subcategorias devem ser apresentadas em diferentes “separadores”, ou seja, em diferentes frisos “deslizantes” e a partir de uma mostrar os produtos respectivos. Por exemplo, escolhendo no friso do Tipo um dos tipos existentes, devemos ver num 2º friso os produtos desse tipo por País e neste friso escolhendo um país devemos ver num 3º friso os Temas dessa colecção. Isto é aplicável aos suportes para listagem ou para venda bem como os complementos, com as devidas alterações e adaptações. Não é necessário fazer mais do que 3 “frisos”.

Seleccionada um suporte ou complemento, têm de ser mostradas as respectivas características e ser possível de as adquirir ou listar consoante o caso.

8. Perfis de Utilizadores e respectivas funções

Existem quatro perfis/roles de utilizador:

- **Utilizador Anónimo**, só no caso da aplicação “*frontend*”, listar os coleccionáveis quer os que não estejam à venda ou quer os que estejam à venda;
- **Cliente**, listar coleccionáveis quer os que não estejam à venda ou quer os que estejam à venda ou comprar coleccionáveis ou de complementos só no caso da aplicação “*frontend*”;

- **Fornecedor**, os fornecedores de colecionáveis ou de complementos, só no caso da aplicação “*frontend*”;
- **Funcionário**, só no caso da aplicação de Gestão de Loja;
- **Administrador**, só no caso da aplicação de Gestão de Loja;

A. As principais funcionalidades/características da aplicação *frontend*, para cada tipo de utilizador, são:

- Utilizador com acesso livre **Utilizador Anónimo**
 - Visualizar e comprar (sem efectivar a compra) os produtos à venda ou para listar;
 - Tem de ser possível visualizar os produtos por categoria, eventualmente por subcategorias, o seu preço e disponibilidade;
 - Tem de ser possível mostrar um produto em “destaque” escolhido aleatoriamente;
 - Seleccionar produtos e respectivas quantidades para compra;
 - Permitir o registo do utilizador como «Cliente» e estado “Pendente”.
- Utilizador com o role/perfil **Cliente**
 - As mesmas funcionalidades descritas atrás para o utilizador anónimo, após registo;
 - Logar-se como “**Cliente**”;
 - Efectivar a listagem e compra dos produtos, isto é encomendar e “pagar”, selecionados para compra;
 - Consultar o histórico das compras que já efetuou;
 - Os dados solicitados para o registo de um Cliente, devem ser específicos deste tipo de “cliente”
- Utilizador com o role/perfil **Fornecedor**
 - Logar-se como “**Fornecedor**”, após registo;
 - Inserir, Consultar e Editar os “seus” produtos para listagem ou para venda podendo alterar os seus preços base, retirá-los das listagens ou de venda ou suspender a sua listagem ou venda e outras funcionalidades que entenda como necessárias (recordar-se que os produtos para listagem ou para venda, suportes media ou complementos, não são propriedade da empresa detentora da aplicação, mas sim de clientes seus do tipo Fornecedor que contratam essa empresa para colocar os seus produtos para venda ou para listagem);
 - Os fornecedores podem ser empresas ou pessoas individuais e podem ter para venda e ou listagem mais do que um produto, suportes media ou complementos;
 - Os produtos inseridos ou editados pelos fornecedores, têm de ficar após estas operações no *estado pendente*, isto é não visíveis pelos clientes só ficando visíveis após activação por parte de um funcionário ou administrador, *estado activo*;
 - Consultar o histórico das vendas de produtos seus que já foram efetuadas;
 - Os dados solicitados para o registo de um Fornecedor, devem ser específicos deste tipo de “cliente”
- Utilizadores com o role/perfil **Funcionário ou Administrador**
 - Na aplicação “*frontend*” não têm nenhuma funcionalidade específica associada a estes dois tipos de utilizador.

B. As principais funcionalidades/características da aplicação de *Gestão de Loja*, para cada tipo de utilizador, são:

- Utilizadores com o role/perfil **Utilizador Anónimo ou Cliente ou Fornecedor**

- Na aplicação *Gestão de Loja* não têm nenhuma funcionalidade específica associada a estes dois tipos de utilizador nem mesmo acesso à mesma.
- Utilizadores com o role/perfil **Funcionário** ou **Administrador**
 - Gestão dos produtos à venda e respectivos preços e stocks:
 - Listar os registos de produtos - com filtros (categorias, disponibilização, etc) e com ordenação e paginação;
 - Adicionar o registo de produtos quer para listagem quer para venda;
 - Editar o registo de produtos;
 - Apagar “fisicamente” os registos de produtos (apenas se não existirem vendas desse produto);
 - Ativar ou inativar o registo de produtos de acordo com a sua disponibilidade;
 - Gestão dos preços e stocks dos produtos;
 - Proceder à alteração de estado dos produtos, activação e desactivação, colocando-os ou retirando como visíveis para os clientes;
 - Ativar ou inativar o registo de clientes tanto do tipo **Cliente** como do tipo **Fornecedor**;
 - Alterar o preço base para preço final, aplicando uma percentagem sobre o preço base definido pelo fornecedor desse produto de modo a obter-se o preço final, preço de venda, que é mostrados aos utilizadores do tipo Cliente, isto é, aos compradores.
 - Gestão das categorias:
 - O mesmo, adaptando a este caso, o indicado atrás para os produtos
 - Gerir os modos de disponibilização:
 - O mesmo, adaptando a estes casos, o indicado atrás para os produtos
 - Gerir as Listagens de produtos que não estejam à venda, mas somente para “mostrar” as colecções individuais:
 - Listar os suportes media e os complementos - com filtros adequados;
 - Gerir as vendas:
 - Listar as vendas - com filtros adequados;
 - Confirmar uma venda;
 - Rejeitar uma venda;
 - Expedir os produtos ao cliente (simular expedição)
 - Actualizar os stocks
 - Gerir os pagamentos (simular pagamento)
 - Não pode apagar nem desativar o seu próprio registo de utilizador.
 - Só os utilizadores com o role/perfil de **Administrador** podem gerir o tipo de utilizador **Funcionário**, nomeadamente atribuir ou retirar esse tipo de perfil;
 - Só os utilizadores com o role/perfil de **Administrador** ou **Funcionário**, podem gerir o tipo de utilizador **Cliente** ou **Fornecedor**, nomeadamente atribuir ou retirar esse tipo de perfil;
 - Só os utilizadores com o role/perfil de **Administrador** ou **Funcionário**, podem alterar o estado de “Pendente” no tipo de utilizador **Cliente** ou **Fornecedor**, de “Pendente” para “Activo” e vice-versa;
- C. As principais funcionalidades/características da aplicação *API RestFull*, para cada tipo de utilizador, são:

- Nenhum tipo de utilizador interagirá directamente com esta aplicação. As interacções com a mesma serão somente feitas indirectamente pela aplicação “*frontend*” através de chamadas a esta *API* e recorrendo-se para tal à utilização de *JWT* e do framework *Identity Core*.

9. Operações Genéricas a serem realizadas por cada uma das aplicações a desenvolver e diferentes plataformas

- Em qualquer das aplicações *frontend* desenvolvidas, quer para a Web, Android, iOS, etc, as operações/funcionalidades devem ser exactamente as mesmas.
- Para o utilizador **Utilizador Anónimo** na aplicação *frontend*, deve ser possível a visualização, selecção e aquisição, após login, de qualquer produto à venda e disponível para tal na aplicação.
- Para o utilizador **Cliente** deve ser também possível que na aplicação *frontend*, além do atrás referido, seja possível a efectivação da aquisição de produtos, ou seja, encomendar e pagar (pagamento simulado), de qualquer produto à venda e disponível para tal na aplicação.
- Para o utilizador **Fornecedor** deve ser possível que na aplicação *frontend*, seja possível a Inserção, Edição, Apagamento e Consulta dos seus produtos, não podendo fazer estas operações para produtos que não sejam “seus”. Os produtos *inseridos* ou *editados* ficam no *estado pendente* até um administrado alterar o estado para *visível*;
- Na aplicação de Gestão de Loja, tem de ser possível aos utilizadores com o role/perfil de **Administrador/Funcionário** executar todas as funções referidas no ponto 8 para estes dois tipos de utilizadores.
- Na aplicação **API RestFull**, nenhum tipo de utilizador terá acesso directo à aplicação

10. Elementos de Entrega Obrigatória

a. Base de Dados

- Para as várias aplicações desenvolvidas e interfaces de UI, *Web + Android + iOS + macOS + Tizen*, a base de dados deve ser uma só e concretizada através do *SQLServer (localdb)*. A base de dados deve estar bem estruturada, completa e consistente, de forma a simplificar o desenvolvimento (atual e futuro) das aplicações ora desenvolvidas.
- Como já referido, a estrutura de dados utilizada tem de ter por base um Modelo-Entidade-Relacionamento, *MER*, também de entrega obrigatória, o qual deve reflectir todas as necessidades em termos de Entidades e respectivas Relações nas aplicações, concretizado através de classes de domínio codificadas em *C#* e correspondentes a tabelas da base de dados criadas com recurso a instruções *migration*.

- A edição/manipulação dos dados, relativamente à base de dados, tem de ser efetuada através da *Entity Framework Core* e *LINQ*.
- É obrigatória a entrega de uma base de dados submetida conjuntamente com o restante trabalho, a qual tem de estar preenchida/populada com uma quantidade suficiente de dados que permita a demonstração e a avaliação de todas as funcionalidades implementadas. *O não cumprimento deste quesito obrigatório resultará numa rejeição liminar do trabalho submetido.*

b. Relatório Obrigatório

- *O relatório é de realização e entrega obrigatória* e deverá conter não somente a indicação de credenciais de acesso mas outros aspectos, como o objectivo do trabalho, se esse objectivo foi ou não alcançado, a completude do que foi realizado, as principais dificuldades sentidas na sua realização e o modo como foram superadas bem como uma análise autocritica do que foi implementado.
- No relatório, têm de ser ainda incluídas, em secção própria, as “regras de negócio” consideradas para a realização do trabalho.

c. Código Desenvolvido

- Todo o código desenvolvido e necessário ao funcionamento da aplicação tem de ser obrigatoriamente entregue. A sua não entrega é motivo para exclusão liminar do trabalho prático submetido.

11. Avaliação

Critérios gerais para a avaliação do trabalho prático:

- A adequação e a abrangência da solução desenvolvida relativamente aos objetivos enunciados. A quantidade do trabalho realizado deve ser relevante e o âmbito do problema não deve estar reduzido de forma excessiva, valoração desta componente 5%.
- A completude e a consistência dos dados de demonstração das funcionalidades da aplicação. Os dados de demonstração têm de permitir, de um modo conveniente, o correcto funcionamento de todas as funcionalidades da aplicação dentro do domínio do problema apresentado, valoração desta componente 5%.
- A completude e a consistência do relatório entregue, valoração desta componente 15%.
- O desempenho do aluno na defesa, nomeadamente a sua capacidade para explicar e apresentar o trabalho submetido, a sua capacidade para realizar pequenas alterações ao código entregue bem como a sua capacidade, se lhe for pedido, para apresentar, explicar e justificar trechos de código específico que tenha utilizado e entregue com a submissão do trabalho e a justificação das opções tomadas e implementadas serão objecto de avaliação e valoração específica, valoração desta componente 35%.

- A qualidade do código desenvolvido e a consistência entre as diversas “partes” das aplicações será bastante valorada e com parâmetros e avaliação e valoração específica. Note-se que a estrutura de classes e a estrutura de dados devem representar convenientemente o domínio do problema e definir uma solução consistente e com qualidade, valoração desta componente 40%.

12.Prazo de Entrega

O tempo previsto para a realização deste trabalho é de um mês, pelo que a data-limite para a **entrega do trabalho prático é o dia 14 de Dezembro de 2025**. Não serão aceites trabalhos entregues após essa data. Os alunos que não façam esta entrega até esta data limite serão considerados como não tendo realizado o trabalho prático. Após a entrega não é possível alterar o trabalho que já foi entregue. Os trabalhos serão entregues via plataforma *moodle*, ano de 2025/26. Quando da entrega, o aluno que realizou o trabalho, escolherá na plataforma *moodle* o dia e hora da defesa do trabalho dentro das opções disponíveis constantes nessa plataforma.

13.Modos de Entrega

A entrega do trabalho prático é realizada em formato digital no *Moodle* da UC (um **ÚNICO** ficheiro *.ZIP* com a seguinte designação **obrigatória**: nomealuno_naluno_n_TP20252026.zip).

Para além do projeto, que tem de incluir todo o código fonte, a base de dados populada, as imagens, o relatório e outros elementos que tenham sido utilizados e também atrás indicados, o ficheiro *.ZIP* deve incluir um ficheiro em formato *PDF* com a seguinte informação:

1. Nome completo + Número de aluno + Ramo + Curso (D, PL, CE), exemplo: **JoaoSilva2022500145DSPL**
2. Dados de acesso à aplicação *Web*, tais como o *username*, a *password* e o perfil, associados aos diversos “utilizadores exemplo”.