Aplicações para a Internet

Engenharia Informática	2º Ano	2º Semestre	2020-21	Avaliação Periódica
Projeto	Prazo para divulgação resultados: 30 Junho 2021			
Data: 12 Abril 2021	**Data de Entrega: ~~19~~ 16 Junho 2021**			

Projeto – Loja MagicShirts

OBJETIVO

O objetivo deste projeto é implementar uma aplicação Web baseada no servidor, utilizando a Framework Laravel, para a loja online "MagicShirts" que vende t-shirts estampadas.

CENÁRIO

Os clientes da loja online "MagicShirts" poderão escolher as estampas a partir do catálogo online da loja ou enviar para o servidor Web imagens com as suas próprias estampas. A "MagicShirts" irá então imprimir as estampas nas t-shirts e enviá-las para casa dos clientes.

Cada cliente poderá adicionar t-shirts a um carrinho de compras (o carrinho de compras deverá ser mantido na sessão do servidor web), selecionando para cada t-shirt a estampa (do catálogo da loja ou a partir de uma imagem introduzida pela cliente), a cor e tamanho da t-shirt e a quantidade de t-shirts a adquirir. Associado a cada t-shirt do carrinho de compras ficará também o preço unitário e o subtotal (quantidade * preço unitário). O carrinho de compras inclui também um total que corresponde à soma de todos subtotais das t-shirts do carrinho. O cliente poderá adicionar e remover t-shirts do carrinho, bem como alterar a cor, tamanho e quantidade de cada t-shirt no carrinho (se quantidade for zero, a t-shirt deverá ser removida do carrinho de compras).

Para confirmar o carrinho de compras e criar a respetiva encomenda, o cliente terá que ter uma conta na loja, o que significa que antes de confirmar o carrinho de compras terá que se autenticar na aplicação (caso ainda não o tenha feito), ou se não ainda tiver uma conta, terá que se registar na aplicação da loja. Durante o processo de confirmação do carrinho de compras o cliente terá que preencher informação sobre o próprio (NIF e nome), forma de pagamento e endereço de envio da encomenda. Todos estes dados deverão ser pré-preenchidos com informação associada à conta do cliente, caso essa informação já tenha sido preenchida anteriormente, mas poderão ser alterados em cada encomenda – os valores do NIF, nome, forma de pagamento e endereço de envio da conta do cliente serão os valores por omissão das suas encomendas.

Quando as encomendas são criadas pelo cliente ficam no estado "pendente" (à espera de pagamento) e quando o pagamento é feito, as encomendas passam ao estado "paga". Depois de enviadas as encomendas ficam no estado "fechada". Caso haja algum problema (no pagamento ou outro) durante o tratamento de uma encomenda, esta fica no estado "anulada".

O tratamento das encomendas é feito pelos funcionários, que serão responsáveis pelo recebimento e confirmação do pagamento, pela estampagem das t-shirts e envio das encomendas para os clientes. Estas operações são externas à aplicação, pelo que no contexto da aplicação os funcionários apenas declaram que a encomenda está "paga" (pagamento foi recebido e confirmado) ou "fechada" (estampagem das t-shirts e envio das encomendas foi efetuado). Os administradores da loja, que são também funcionários, para além de declarar a encomenda como "paga" ou "fechada" poderão também declarar qualquer encomenda como "anulada".

No que se refere aos preços das t-shirts, existem apenas 4 preços distintos que poderão ser configurados pelos administradores da loja:

- T-shirt com estampa do catálogo da loja;
- T-shirt com estampa definida pelo cliente;
- T-shirt com estampa do catálogo da loja e com desconto de quantidade;
- T-shirt com estampa definida pelo cliente e com desconto de quantidade.

UTILIZADORES E FUNCIONALIDADES

A aplicação Web a desenvolver deverá suportar 4 tipos de utilizadores: os utilizadores anónimos, os clientes, os funcionários e os administradores da loja.

Os utilizadores anónimos deverão poder consultar informação genérica - e estática – sobre a loja "MagicShirts"; consultar e filtrar o catálogo de estampas da loja; adicionar, remover ou alterar t-shirts do carrinho de compras; registar uma nova conta do cliente.

Os clientes podem fazer tudo o que os utilizadores anónimos fazem (exceto registar uma nova conta) e para além disso, deverão poder ver e alterar a informação da sua conta (perfil de utilizador); alterar a senha de entrada; confirmar o carrinho de compras e criar encomendas; consultar o seu histórico de encomendas (incluindo recibos em PDF); gerir as suas próprias estampas (usadas em exclusivo pelas suas t-shirts). A aplicação deverá enviar para o cliente correspondente, e de forma automática, um e-mail quando a encomenda é criada (estado = "pendente"), caso a encomenda seja anulada (estado = "anulada") e quando a encomenda é enviada (estado = "fechada"). Neste último caso (quando a encomenda é enviada) o e-mail deverá enviar em anexo um ficheiro PDF com o recibo relativo à encomenda.

Os funcionários só podem alterar a senha de entrada na aplicação; consultar a lista de encomendas "pendentes" e "pagas" (as únicas encomendas que necessitam para fazer o seu trabalho) e respetivos detalhes; declarar as encomendas "pendentes" como "pagas" e as encomendas "pagas" como "fechadas".

Nota: os pagamentos serão apenas simulados, ou seja, as operações de pagamento não são realmente executadas no contexto da aplicação – o funcionário ou o administrador apenas declara uma encomenda como "paga" sem que haja uma transação real num serviço de pagamentos (como o PayPal).

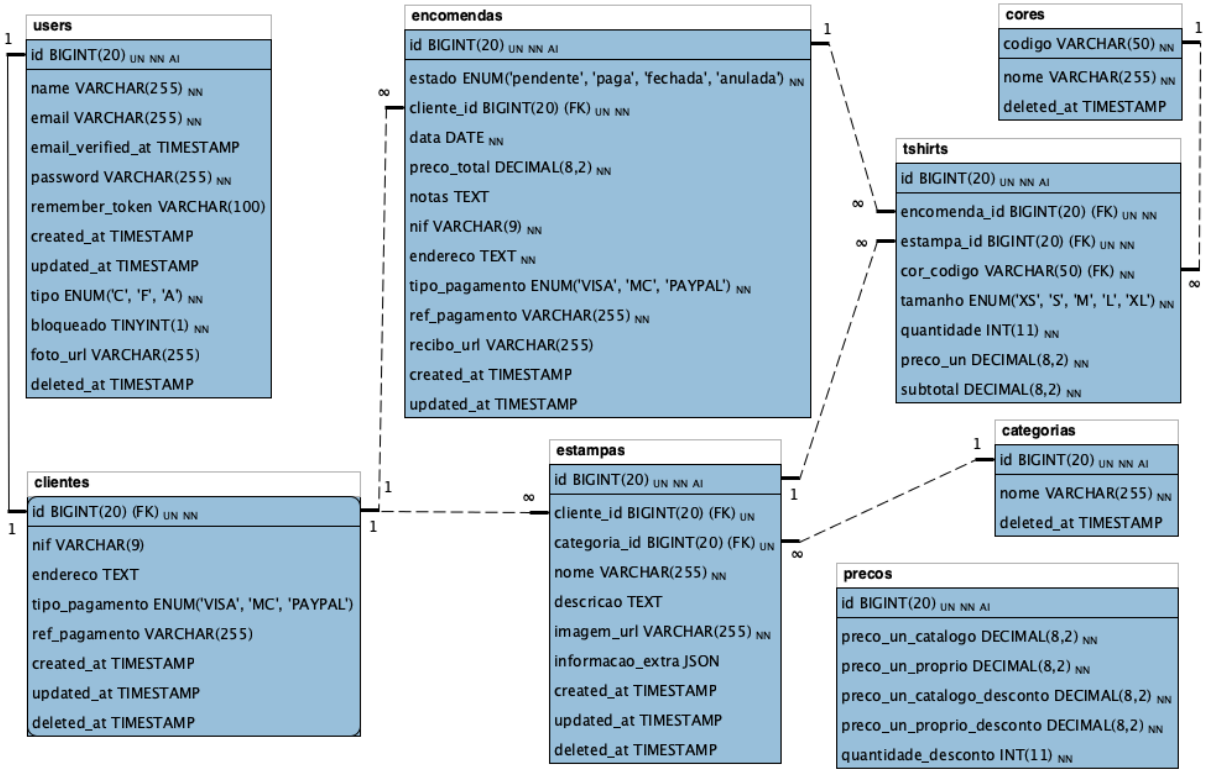
Os administradores da loja, para além de alterarem a sua senha de entrada e os dados da sua conta (perfil de utilizador) poderão também gerir (consultar, filtrar, criar, alterar, bloquear ou remover) as contas dos restantes funcionários e administradores. Poderão também consultar e filtrar a lista de clientes e bloquear e apagar (usando o *soft delete*) as contas dos clientes – apenas o próprio cliente pode criar a sua conta (através do registo) e alterar os dados da sua conta.

Os administradores deverão também gerir as categorias e o catálogo das estampas e configurar os preços e a lista de cores das t-shirts. No que se refere às encomendas deverão conseguir consultar e filtrar qualquer tipo de encomenda (independentemente do estado) e respetivos detalhes (incluindo o PDF com o recibo) e deverão conseguir declarar qualquer encomenda como "anulada", "paga" ou "fechada".

Por último, os administradores deverão conseguir visualizar informação estatística relativa ao negócio da loja online. A decisão sobre a forma e o tipo de informação apresentada é da responsabilidade dos estudantes. Por exemplo, a informação pode ser apresentada em tabelas, gráficos, ou de outras formas, e pode incluir totais, médias, máximos, mínimos de vendas em valor ou quantidade por mês, por ano, organizadas por estampas ou categorias, por cliente, etc.

BASE DE DADOS

A estrutura da base de dados é fornecida (através de migrações) e não pode ser alterada pelos estudantes, exceto se explicitamente indicado pelos docentes. Também serão fornecidos dados de exemplo através de "*database seeders*". O diagrama ER da estrutura da BD é o seguinte:



TABELAS E COLUNAS

De seguida são descritas as tabelas e as colunas da base de dados. A estrutura das tabelas e colunas (por exemplo, as chaves estrangeiras, os tipos de dados das colunas, se as colunas são ou não obrigatórias) e a descrição aqui apresentada, são relevantes para o funcionamento da aplicação e validação dos dados introduzidos pelos utilizadores.

USERS

Tabela com os utilizadores registados da aplicação - clientes, funcionários e administradores.

Esta tabela é uma superclasse da tabela clientes, o que significa que alguns utilizadores são clientes, e nesse caso os dados do cliente estão distribuídos pelas 2 tabelas: users e clientes. A tabela users terá todos os dados comuns a todos os users e a tabela clientes terá todos os dados específicos (exclusivos) dos clientes.

- "**id**" – chave primária – nº inteiro automático.
- "**name**" – nome do utilizador.
- "**email**" – e-mail do utilizador. O e-mail deverá ser único e serve também como credencial de autorização (em conjunto com a password).
- "**password**" – *hash* da password do utilizador.
- "**tipo**" – tipo de utilizador. C (cliente); F (funcionário) e A (administrador).
- "**bloqueado**" – booleano (1 ou 0) que indica se o utilizador está ou não bloqueado. Um utilizador bloqueado não deverá conseguir entrar na aplicação (autorização deverá falhar)
- "**foto_url**" (opcional) – nome ("relativo") do ficheiro com a foto (ou avatar) do utilizador.

Os próximos campos são geridos automaticamente pelo Laravel e normalmente não são apresentados na aplicação:

- ***"email_verified_at"*** *(opcional) – data em que o e-mail do utilizador foi verificado – esta coluna é gerida pelo sistema de autenticação do Laravel.*
- ***"remember_token"*** *(opcional) – para implementação da funcionalidade "remember me" – esta coluna é gerida pelo Laravel.*
- ***"created_at"*** *(opcional) – data e hora em que o registo foi criado – esta coluna é gerida pelo Laravel.*
- ***"updated_at"*** *(opcional) – data e hora em que o registo foi alterado pela última vez – esta coluna é gerida pelo Laravel.*
- ***"deleted_at"*** *(opcional) – data e hora em que o registo foi apagado (com soft delete) – esta coluna é gerida pelo Laravel.*

CLIENTES

Tabela com os clientes registados da loja.

Esta tabela é uma subclasse da tabela users, o que significa que os clientes são também users. Os dados do cliente estão distribuídos pelas 2 tabelas: users e clientes. A tabela users terá os dados que são comuns a todos os users (por exemplo, o nome do cliente está na tabela users) e a tabela clientes terá os dados específicos (exclusivos) dos clientes.

- "**id**" – chave primária – valor inteiro igual ao "id" do user. O "id" da tabela clientes não é preenchido automaticamente, mas sim, terá sempre o mesmo valor que o "id" do user (o cliente é um user). Neste caso, o "id" do cliente é também a chave estrangeira responsável pelo relacionamento da tabela clientes com a tabela users.

  Nota: quando se insere um cliente é sempre necessário inserir primeiro um registo na tabela "users" e de seguida o registo equivalente na tabela "clientes". O contrário deverá acontecer se for necessário apagar um cliente: primeiro apaga-se o registo da tabela "clientes" e de seguida o registo equivalente na tabela "users".
- "**nif**" (opcional) – NIF (Número de Identificação Fiscal) do cliente. O nif (quando preenchido) deverá ter sempre 9 dígitos – por exemplo: 148928093. O nif da tabela clientes será usado para pré-preencher o nif das encomendas do cliente – corresponde ao valor por omissão do nif.
- "**endereco**" (opcional) – Endereço do cliente usado para envio das encomendas. O endereço da tabela clientes será usado para pré-preencher o endereço das encomendas do cliente – corresponde ao valor por omissão do endereço.
- "**tipo_pagamento**" (opcional) – Tipo de pagamento usado por omissão nas encomendas do cliente. Os valores aceites são: "VISA" (cartão de crédito Visa); "MC" (cartão de crédito Master Card) e "PAYPAL" (Paypal).
- "**ref_pagamento**" (opcional) – Referência de pagamento usado por omissão nas encomendas do cliente. Se o tipo de pagamento for "VISA" ou "MC", então o valor da referência de pagamento corresponde ao número do cartão de crédito e deverá ter 16 dígitos. Se o tipo de pagamento for "PAYPAL" então o valor da referência de pagamento corresponde ao e-mail da conta do Paypal. Se o tipo de pagamento não for preenchido (valor = null) então a referência de pagamento também não deverá ser preenchida (valor = null).

Os próximos campos são geridos automaticamente pelo Laravel e normalmente não são apresentados na aplicação:

- ***"created_at"*** *(opcional) – data e hora em que o registo foi criado – esta coluna é gerida pelo Laravel.*
- ***"updated_at"*** *(opcional) – data e hora em que o registo foi alterado pela última vez – esta coluna é gerida pelo Laravel.*
- ***"deleted_at"*** *(opcional) – data e hora em que o registo foi apagado (com soft delete) – esta coluna é gerida pelo Laravel.*

CATEGORIAS

Tabela com as categorias das estampas. Permite às estampas do catálogo serem organizadas por categorias (exemplos de categorias de estampas: "engraçadas"; "memes"; "logotipos"; "abstratas"; etc.).

- "**id**" – chave primária – nº inteiro automático.
- "**nome**" – nome da categoria.

O próximo campo é gerido automaticamente pelo Laravel e normalmente não é apresentado na aplicação:

- ***"deleted_at"*** *(opcional) – data e hora em que o registo foi apagado (com soft delete) – esta coluna é gerida pelo Laravel.*

CORES

Tabela com as cores das t-shirts que a loja vende. As t-shirts do carrinho de compras e das encomendas ficarão associadas a umas das cores nesta tabela – a loja só vende t-shirts nestas cores.

- "**codigo**" – chave primária – corresponde ao código da cor usada nas propriedades css. Alguns exemplos possíveis: "white"; "black"; "steelblue"; "#7fffd4"; "#ab0034"
- "**nome**" – nome da cor que será apresentado ao cliente.

O próximo campo é gerido automaticamente pelo Laravel e normalmente não é apresentado na aplicação:

- ***"deleted_at"*** *(opcional) – data e hora em que o registo foi apagado (com soft delete) – esta coluna é gerida pelo Laravel.*

PRECOS

Tabela com a configuração atual dos preços das t-shirts da loja. Esta tabela deverá ter sempre uma linha, e apenas uma linha, porque a aplicação não suporta múltiplas configurações ao longo do tempo – é apenas relevante a configuração atual.

- "**id**" – chave primária – nº inteiro automático.

  Nota: como a tabela tem sempre uma linha e uma linha apenas, este campo é irrelevante. No entanto, o facto de se definir uma chave primária facilita a integração da tabela com os modelos do Laravel.
- "**preco_un_catalogo**" – preço de uma t-shirt com uma estampa do catálogo da loja.
- "**preco_un_proprio**" – preço de uma t-shirt com uma estampa definida pelo cliente.
- "**preco_un_catalogo_desconto**" – preço de uma t-shirt com uma estampa do catálogo da loja, com desconto de quantidade.
- "**preco_un_proprio_desconto**" – preço de uma t-shirt com uma estampa definida pelo cliente, com desconto de quantidade.

- "**quantidade_desconto**" – quantidade a partir da qual a loja aplica os preços com desconto. Por exemplo, se o valor da "quantidade_desconto" for 10, os preços com desconto de quantidade serão aplicados sempre que numa encomenda o cliente comprar 10 ou mais t-shirts com a mesma estampa, cor e tamanho.

ESTAMPAS

Tabela com as estampas do catálogo da loja e com as estampas dos clientes. Se o campo "cliente_id" estiver a null, será uma estampa do catálogo da loja, e caso contrário, será uma estampa de um determinado cliente.

- "**id**" – chave primária – nº inteiro automático.
- "**cliente_id**" (opcional) – chave estrangeira (relacionada com a tabela clientes) – identifica o cliente que está associado à estampa – que é "o dono" da estampa. Se o valor for null, significa que é uma estampa do catálogo da loja (não tem um cliente associado/dono).
- "**categoria_id**" (opcional) – chave estrangeira (relacionada com a tabela categorias) – identifica a categoria a que pertence a estampa. Quando o valor é null significa que a estampa não pertence a nenhuma categoria específica. No caso de ser uma estampa associado a um cliente (cliente_id é diferente de null), o valor da categoria_id deverá ser sempre null – significa que as estampas dos clientes não serão organizadas por categoria – só é necessário manter essa organização para o catálogo da loja.
- "**nome**" – nome dado à estampa. Apesar das estampas representarem uma imagem que será impressa nas t-shirts, terão também que ter um nome para ajudar à organização, consulta e filtros das estampas.
- "**descricao**" (opcional) – descrição (texto) associada à estampa que irá aparecer no catálogo.
- "**imagem_url**" – nome ("relativo") do ficheiro com a imagem da estampa. Este campo é obrigatório, uma vez que não tem sentido haver um registo de uma estampa sem uma imagem associada à mesma.
- "**informacao_extra**" (opcional) – campo que permite aos estudantes acrescentar qualquer informação sobre a estampa, que considere relevante para a implementação do projeto – a utilização deste campo é opcional e depende da implementação de cada projeto. O campo aceita objetos json com qualquer tipo de estrutura (a definir pelos estudantes).

  Inicialmente, a inclusão deste campo foi feita para permitir que a aplicação desenhe um "*preview*" de como a t-shirt ficaria numa determinada cor e depois de aplicada uma determinada estampa. Ao desenhar a imagem da estampa sobre a imagem de uma t-shirt, poderá ser necessário ajustar o tamanho, a posição ou a transparência da estampa (entre outras propriedades a definir pelos estudantes). Este campo vai permitir guardar os ajustes de cada uma das estampas (que podem ser diferentes conforme as imagens das estampas).

Os próximos campos são geridos automaticamente pelo Laravel e normalmente não são apresentados na aplicação:

- ***"created_at"*** *(opcional) – data e hora em que o registo foi criado – esta coluna é gerida pelo Laravel.*
- ***"updated_at"*** *(opcional) – data e hora em que o registo foi alterado pela última vez – esta coluna é gerida pelo Laravel.*
- ***"deleted_at"*** *(opcional) – data e hora em que o registo foi apagado (com soft delete) – esta coluna é gerida pelo Laravel.*

ENCOMENDAS

Tabela com as encomendas.

Nota: enquanto o carrinho de compras estiver a ser construído, não deverá ser registada nenhuma encomenda. Só depois do cliente confirmar o carrinho de compras é que a aplicação irá registar uma encomenda (com o estado "pendente") – o carrinho de compras deverá ser guardado na sessão do servidor.

- "**id**" – chave primária – nº inteiro automático.
- "**estado**" – estado da encomenda. Valores possíveis: "pendente", "paga", "fechada" e "anulada". Quando uma encomenda é criada o valor do estado deverá ser "pendente".
- "**cliente_id**" – chave estrangeira (relacionada com a tabela clientes) – identifica o cliente da encomenda.
- "**data**" – data da encomenda – é apenas necessário o dia (a hora é ignorada).
- "**preco_total**" – preço total da encomenda (soma de todos os subtotais das t-shirts).
- "**notas**" (opcional) – texto com notas (observações) que o cliente pode acrescentar à encomenda durante o processo de confirmação do carrinho de compras.
- "**nif**"– NIF (Número de Identificação Fiscal) do cliente. O nif (quando preenchido) deverá ter sempre 9 dígitos – por exemplo: 148928093. Se a tabela clientes (do cliente da fatura) tiver um valor para o nif, então esse valor será usado por omissão quando o cliente confirmar a compra. No entanto, o cliente tem a liberdade de escolher outro nif.
- "**endereco**" – Endereço para onde será enviada a encomenda. Se a tabela clientes (do cliente da fatura) tiver um valor para o endereço, então esse valor será usado por omissão quando o cliente confirmar a compra. No entanto, o cliente tem a liberdade de escolher outro endereço.
- "**tipo_pagamento**" – Tipo de pagamento usado nesta encomenda. Os valores aceites são: "VISA" (cartão de crédito Visa); "MC" (cartão de crédito Master Card) e "PAYPAL" (Paypal). Se a tabela clientes (do cliente da fatura) tiver um valor para o tipo_pagamento, então esse valor será usado por omissão quando o cliente confirmar a compra. No entanto, o cliente tem a liberdade de escolher outro tipo_pagamento.

- "**ref_pagamento**" – Referência de pagamento usado nesta encomenda. Se o tipo de pagamento for "VISA" ou "MC", então o valor da referência de pagamento corresponde ao número do cartão de crédito e deverá ter 16 dígitos. Se o tipo de pagamento for "PAYPAL" então o valor da referência de pagamento corresponde ao e-mail da conta do Paypal. Se a tabela clientes (do cliente da fatura) tiver um valor para a ref_pagamento, então esse valor será usado por omissão quando o cliente confirmar a compra. No entanto, o cliente tem a liberdade de escolher outra ref_pagamento.
- "**recibo_url**" (opcional) – nome do ficheiro com o PDF relativo ao recibo da encomenda. Quando a encomenda fica no estado "fechada" a aplicação web deverá gerar automaticamente um PDF com o recibo relativo à encomenda. Este campo irá guardar o nome do ficheiro PDF gerado, de maneira a que possa ser enviado por mail e possa ser consultado a qualquer altura pelo cliente da encomenda ou pelos administradores da loja.

Os próximos campos são geridos automaticamente pelo Laravel e normalmente não são apresentados na aplicação:

- ***"created_at"*** *(opcional) – data e hora em que o registo foi criado – esta coluna é gerida pelo Laravel.*
- ***"updated_at"*** *(opcional) – data e hora em que o registo foi alterado pela última vez – esta coluna é gerida pelo Laravel.*

TSHIRTS

Tabela com as t-shirts vendidas nas encomendas. Uma linha desta tabela irá corresponder a uma linha da encomenda e irá incluir a informação necessária para produzir as t-shirts, nomeadamente a cor, o tamanho, a estampa e a quantidade de t-shirts a produzir. Inclui também a informação relativa ao preço unitário da t-shirt e o subtotal (preço unitário * quantidade).

Nota: Se a encomenda incluir t-shirts com a mesma estampa e cor, mas de 2 tamanhos distintos (ex: M e L), serão necessárias 2 linhas da tabela tshirts (uma linha para cada tamanho).

- "**id**" – chave primária – nº inteiro automático.
- "**encomenda_id**" – chave estrangeira (relacionada com a tabela encomendas) – identifica a encomenda a que pertence a t-shirt.
- "**estampa_id**" – chave estrangeira (relacionada com a tabela estampas) – identifica a estampa da t-shirt.
- "**cor_codigo**" – chave estrangeira (relacionada com a tabela cores) – identifica a cor da t-shirt. O valor deste campo, para além de referenciar uma cor da tabela cores, corresponde ele próprio ao código da cor. Por exemplo: "white"; "black"; "steelblue"; "#7fffd4"; "#ab0034".
- "**tamanho**" – tamanho da t-shirt. Valores possíveis: "XS", "S", "M", "L" e "XL".
- "**quantidade**" – quantidade de t-shirts (com uma determinada estampa, cor e tamanho) vendidas na encomenda.

- "**preco_un**" – preço unitário da t-shirt. Este preço depende da configuração atual (no momento em que a encomenda é criada) dos preços (definida na tabela "precos").
- "**subtotal**" – preço total de todas as t-shirts (com uma determinada estampa, cor e tamanho) nesta encomenda (subtotal = quantidade * preço unitário).

FUNCIONALIDADES

A aplicação Web deverá implementar um conjunto de grupos de funcionalidades, que serão descritos de seguida. A implementação das mesmas deve ter em conta a informação descrita no cenário, na base de dados (na descrição do enunciado e na estrutura da própria base de dados) e na própria descrição dos grupos de funcionalidades, bem como todos os padrões e as boas práticas da Framework Laravel. Para além do modelo de negócio, requisitos e restrições que se podem inferir a partir do enunciado e da estrutura da base de dados, os estudantes têm liberdade para decidir sobre a interface com o utilizador e usabilidade, bem como sobre aspetos do negócio não descritos.

1. AUTENTICAÇÃO, PERFIL E GESTÃO DE UTILIZADORES

A aplicação Web a desenvolver deverá suportar autenticação através de um login com as credenciais e-mail + senha (password) – o processo de login é o mesmo para qualquer tipo de utilizador. Depois de autenticado qualquer utilizador pode alterar a senha (password) e sair da aplicação (logout). Caso o utilizador não se lembre da senha deverá ter a possibilidade de fazer o "*reset*" da senha – a aplicação deverá enviar um mail com um link para efetuar essa operação ("*password reset*").

Os utilizadores anónimos (utilizadores não autenticados), para além do acesso ao login para autenticação podem registar-se como clientes. Depois de concluído o processo de registo, a aplicação irá enviar um e-mail para verificação/confirmação da validade de e-mail.

Os clientes têm acesso ao seu perfil de utilizador, onde podem consultar e alterar os seus dados pessoais, incluindo a sua foto (avatar). Os funcionários não têm acesso ao seu perfil de utilizador, pois apenas os administradores podem consultar e alterar a sua informação pessoal.

Os administradores são os responsáveis pela gestão de utilizadores dos funcionários e dos próprios administradores, o que significa que podem consultar, filtrar, criar, alterar, bloquear ou remover as contas dos funcionários e administradores, bem como aceder e alterar o perfil de utilizador de qualquer funcionário ou administrador. Em contrapartida, mesmo os administradores não podem aceder aos perfis de utilizadores dos clientes – apenas podem consultar e filtrar a lista de clientes e bloquear ou apagar (usando o *soft delete*) as contas dos clientes.

Nota: os e-mails deverão ser enviados através do serviço mailtrap.io.

2. CATÁLOGO

Os clientes e os utilizadores anónimos (qualquer utilizador é anónimo antes de fazer o login) deverão conseguir consultar o catálogo de estampas da loja, que deverá incluir pelo menos as imagens, nomes e descrições das estampas e que poderá ser organizado por categorias. Deverá também ser possível pesquisar ou filtrar estampas pelo nome, categoria e/ou descrição da estampa.

Os administradores deverão fazer a gestão do catálogo, o que inclui gerir (consultar, inserir, atualizar, apagar) as categorias, as estampas e a lista de cores das t-shirts e configurar os preços.

3. CARRINHO DE COMPRAS

Os clientes e os utilizadores anónimos (qualquer utilizador é anónimo antes de fazer o login) deverão ter acesso a um carrinho de compras com um conjunto de t-shirts que pretendem comprar. A opção para adicionar t-shirts ao carrinho de compras deverá estar disponível no catálogo de estampas (por exemplo, através de um botão para adicionar ao carrinho ou um botão complementado com campos para definir a quantidade e escolher a cor e tamanho). Caso implemente as estampas próprias do cliente, terá também que permitir adicionar t-shirts com esse tipo de estampas.

O carrinho de compras deverá ser mantido numa sessão do servidor ("HTTP Session" do Laravel) e deverá apresentar a informação e ter o comportamento descrito no cenário.

4. ENCOMENDAS

Os clientes deverão criar encomendas confirmando a informação presente no carrinho de compras. As novas encomendas ficam no estado "pendente" e as t-shirts incluídas na encomenda replicam a informação sobre as t-shirts do carrinho de compras. A aplicação deverá garantir que apenas os clientes poderão confirmar (ou concluir) o carrinho de compras – a forma como isso é garantido depende da usabilidade da aplicação. Por exemplo, se um utilizador anónimo iniciar o processo de confirmação de encomenda a aplicação poderá redirecionar esse utilizador para a página de login, e só permitir que o processo de confirmação continue depois do utilizador efetuar a autenticação como cliente.

Os clientes deverão também ter acesso ao histórico das suas encomendas, e para cada uma das encomendas no histórico, deverão ter acesso ao detalhe da encomenda – incluindo toda a informação relativa às t-shirts da encomenda, e no caso das encomendas "fechadas", ao ficheiro PDF com o recibo da encomenda (caso tenha sido implementado).

Os funcionários podem consultar a lista de encomendas "pendentes" e "pagas" (as únicas encomendas que necessitam para fazer o seu trabalho). Também deverão ter acesso ao detalhe completo das encomendas "pendentes" e "pagas", bem como declarar as encomendas "pendentes" como "pagas" e as encomendas "pagas" como "fechadas".

Os administradores deverão conseguir consultar e filtrar (filtrar no mínimo por estado, cliente e data) qualquer tipo de encomenda (independentemente do estado) e respetivos detalhes (incluindo o PDF com o recibo, caso tenha sido implementado). Os administradores deverão também conseguir declarar qualquer encomenda como "anulada", "paga" ou "fechada".

5. ESTAMPAS PRÓPRIAS

Os clientes poderão adicionar ao seu carrinho de compras, t-shirts com as suas próprias estampas, ou seja, com imagens que envia para o servidor e que são para uso exclusivo nas suas t-shirts. Estas estampas próprias só serão acessíveis aos donos das mesmas (o dono da estampa é o cliente que enviou a estampa para o servidor) e aos funcionários e administradores da loja – os utilizadores anónimos e os restantes clientes não deverão, em circunstância alguma, aceder à informação e imagem destas estampas.

A aplicação deverá permitir que o cliente faça a gestão (consultar, adicionar, atualizar e remover) das suas próprias estampas, o que implica o envio das imagens (upload de imagens) das estampas para o servidor. Apesar de ser possível (e desejável) reaproveitar o código relativo à gestão das estampas do catálogo, a gestão das estampas do cliente deverá ser feita num espaço exclusivo do cliente. Os administradores não deverão ter acesso à gestão das estampas dos clientes – apenas acedem (veem) às mesmas quando consultam o detalhe de uma encomenda.

Nota 1: as t-shirts com estampas próprias têm um preço diferente das t-shirts com estampas do catálogo.

Nota 2: as estampas próprias não têm uma categoria (categoria_id = null)

6. RECIBOS E E-MAIL

Quando uma encomenda é enviada (quando o estado passa a "fechada"), a aplicação deverá gerar um ficheiro PDF com o recibo relativo à encomenda. Esse ficheiro deverá replicar a informação presente na encomenda, incluindo os detalhes das t-shirts (exceto a imagem da estampa que é opcional no recibo) e deverá ter uma apresentação típica de um recibo, nomeadamente o nome e logotipo da empresa que emite o recibo, informação (NIF e nome) sobre o cliente, data, lista de produtos vendidos, total do recibo, etc.. O ficheiro com o recibo ficará armazenado no servidor e deverá estar disponível para visualização e/ou download sempre que o cliente ou um dos administradores consultar a encomenda associada. Os restantes clientes, os funcionários e os utilizadores anónimos não deverão, em circunstância alguma, aceder ao recibo.

Depois do recibo ser gerado pela aplicação (quando o estado da encomenda passa a "fechada"), esta deverá enviar automaticamente um e-mail para o cliente com um texto a agradecer a compra e com uma cópia do recibo (ficheiro PDF em anexo). A aplicação deverá também enviar um e-mail automático quando a encomenda é criada (estado = "pendente"), com uma mensagem a informar o cliente que a encomenda está a ser processada.

Nota: os e-mails deverão ser enviados através do serviço mailtrap.io.

7. ESTATÍSTICAS

Os administradores deverão conseguir visualizar informação estatística relativa ao negócio da loja online. A decisão sobre a forma e o tipo de informação apresentada é da responsabilidade dos estudantes. Por exemplo, a informação pode ser apresentada em tabelas, gráficos, ou de outras formas, e pode incluir totais, médias, máximos, mínimos de vendas em valor ou quantidade por mês, por ano, organizadas por estampas ou categorias, por cliente, etc.

8. PREVIEW DE T-SHIRTS

A aplicação deverá apresentar uma imagem com um "*preview*" de como uma t-shirt ficaria numa determinada cor e depois de aplicada uma determinada estampa. A implementação desta funcionalidade permitiria que no carrinho de compras e nos detalhes da encomenda, as t-shirts fossem representadas através desta imagem (imagem com a "*preview*"), em vez de apresentar em separado uma imagem com a estampa e algo que represente a cor.

A implementação desta funcionalidade pressupõe que a imagem da estampa se sobreponha (usando CSS e/ou biblioteca de manipulação de imagens) a uma imagem base com a representação de uma t-shirt vazia. Estas 2 imagens poderão ter formatos diferentes (com ou sem transparências) e ter tamanhos completamente díspares. Por essa razão, poderá ser necessário ajustar o tamanho, a posição ou a transparência da estampa, de maneira a que a imagem resultante tenha o aspeto pretendido. Para permitir guardar os ajustes de cada uma das estampas (que podem ser diferentes confirme as imagens das estampas), a base de dados fornecida inclui o campo "informacao_extra" na tabela "estampas" onde é possível guardar um objeto JSON com qualquer tipo de estrutura (a definir pelos estudantes). A utilização ou não deste campo, bem como a sua estrutura, é uma decisão exclusiva dos estudantes, porque a forma como esta funcionalidade é implementada é livre – poderá haver soluções que não necessitem destes dados para ajustar as imagens.

Também a gestão do catálogo das estampas por parte dos administradores, ou das estampas próprias por parte dos clientes, poderá ser ajustada por forma a incorporar algum mecanismo que possibilite o ajuste das estampas por parte dos administradores e clientes – mais uma vez, a incorporação ou não desta funcionalidade depende da forma como a implementação é feita.

ENTREGA

A entrega do projeto deverá incluir 2 ficheiros relativos ao grupo acrescido de 1 ficheiro por cada elemento do grupo. Por exemplo, se o grupo tiver 3 elementos deverão ser entregues 5 ficheiros (2 relativos ao grupo e 3 individuais). Os ficheiros a entregar são os seguintes:

- 1 Ficheiro Excel com o **relatório do grupo** – o relatório do grupo vai incluir a identificação dos elementos do grupo e informação sobre a implementação do projeto. Este ficheiro

deverá ser entregue apenas por um dos elementos do grupo. O modelo para o relatório será fornecido pelo professor;

- 1 Ficheiro Zip com todo o código do **projeto** – este ficheiro zip inclui uma cópia de todos os ficheiros e pastas do projeto, **exceto** as pastas "**vendor**", "**database**" e "**node_modules**". Este ficheiro deverá ser entregue apenas por um dos elementos do grupo
- Vários ficheiros Excel com **relatórios individuais** – o relatório individual irá incluir a autoavaliação e avaliação entre pares. Todos os estudantes terão que entregar um relatório individual. O modelo para o relatório será fornecido pelo professor.

AVALIAÇÃO

Os critérios de avaliação do projeto são definidos pelos grupos de funcionalidades identificados anteriormente. Os pesos dos grupos de funcionalidades são os seguintes:

Nº	Peso	Grupo de funcionalidades
1	20%	Autenticação, perfil e gestão de utilizadores
2	20%	Catálogo
3	15%	Carrinho de compras
4	15%	Encomendas
5	5%	Estampas próprias
6	10%	Recibos e e-mail
7	10%	Estatísticas
8	5%	*Preview* de t-shirts

A avaliação de cada um dos grupos de funcionalidades depende da quantidade e qualidade de funcionalidades implementadas, da integração das funcionalidades na aplicação e da usabilidade e correção (funcionamento correto) das funcionalidades. Serão também tidos em conta o layout e aspeto visual (aspeto consistente e com layout bem estruturado) e alguns requisitos não funcionais e transversais a toda a aplicação, tais como (entre outros):

- A implementação da aplicação deverá seguir o padrão arquitetural MVC (*Model-View-Controller*) e a estrutura, padrões de desenho e as boas práticas definidas na Framework Laravel;
- Os métodos (GET, POST, PUT, PATCH, DELETE) e URL das rotas deverão seguir as boas práticas das aplicações Web;
- A aplicação deverá usar as funcionalidades e componentes do Laravel adequados, para resolver os vários desafios da aplicação. Por exemplo, usar sempre que possível o Eloquent (em vez da classe DB) para aceder à base de dados; usar "Form Request" para implementar validações no servidor, usar o sistema de autenticação fornecido pelo Laravel (em vez de

implementar um sistema próprio), usar o método de hash fornecido pelo Laravel para guardar as passwords, usar Laravel *policies* para autorização, etc;

- A aplicação deverá seguir o princípio DRY (*Dont't Repeat Yourself*), ou seja, deve sempre que possível evitar a repetição de código, sem, no entanto, quebrar as boas práticas do Laravel. Por exemplo, utilizar vistas parciais sempre que houver secções de HTML que se repitam em várias vistas ou acrescentar métodos aos modelos que devolvem determinado conjunto de dados que serão utilizados em vários controladores;
- A aplicação deverá ter o melhor desempenho possível. O cumprimento deste requisito depende em grande parte da forma como é feita a interação com a base de dados. No que concerne ao desempenho do acesso à base de dados, dois grandes princípios deverão nortear a implementação da aplicação: reduzir ao máximo o nº de comandos que são executados sobre a base de dados e garantir que cada consulta à base de dados traz apenas a informação necessária (só as colunas e as linhas estritamente necessárias).

  A aplicação deverá também ter em consideração outras características que melhorem o desempenho geral da aplicação, tais como: utilização de *queues* para operações mais pesadas; utilização adequada de sistemas de cache (pesquisar sobre cache no Laravel); garantir que os recursos devolvidos pelo servidor têm o menor tamanho possível (reduzir o tamanho das respostas HTTP), etc;
- A aplicação deverá garantir o máximo de segurança e privacidade aos seus utilizadores. A aplicação tem que garantir que todos as operações e recursos que disponibiliza, são apenas acessíveis a quem está autorizado a fazê-lo. A informação privada dos utilizadores deverá estar sempre protegida de acessos indevidos.

  A implementação dos mecanismos que garantem a segurança, privacidade e autorização, deverá ser feita de acordo com as boas práticas do Laravel e utilizando as funcionalidades e componentes do Laravel adequados.

  > *Nota: uma das principais características que tornam as aplicações web mais seguras e com melhor proteção da privacidade, tem a ver com a utilização de um protocolo seguro para comunicação entre o cliente e o servidor – utilização do HTTPS em vez do HTTP. Como neste projeto estamos a tratar apenas da camada aplicacional do sistema e não da camada da comunicação, NÃO é necessário a utilização do HTTPS.*