



Licenciatura Engenharia Informática e Multimédia

Instituto Superior de Engenharia de Lisboa

Ano letivo 2024/2025

Sistemas Multimédia para a Internet

Relatório: Projeto Final

Turma: 61D

Grupo: G44

Nome: Miguel Cordeiro

Número: 49765

08 de junho de 2025

Índice:

Introdução.....	5
Estrutura do Projeto.....	7
Modelo de Dados “bodybliss”.....	9
Tabelas e Campos.....	9
Users.....	9
Repository.....	10
Products.....	10
Order_items.....	10
Orders.....	10
Email_accounts.....	11
Cart.....	11
Relacionamentos entre Tabelas.....	11
users → cart:.....	11
products → cart:.....	11
users → repository:.....	12
users → orders:.....	12
orders → order_items:.....	12
Enquadramento com Plataformas Semelhantes.....	13
Mockups.....	15
Página Inicial.....	15
Página About.....	15
Página Catalog.....	16
Página Contact.....	16
Página Login.....	17
Página Register.....	17
Página Account.....	17
Página Cart (com mouse no hover cart).....	18
Página Checkout.....	18
Funcionalidades Implementadas.....	19
Mecanismo de Redirecionamento (PID “Page ID”).....	19
Gestão de Utilizadores.....	19
WebServices Desenvolvidos.....	21
Uso de API's.....	22
Envio email de verificação.....	23
Hierarquia de diferentes utilizadores.....	24
Atualização e Gestão da imagem de perfil.....	25
Utilização XML.....	26
Sistema de Gestão de Conteúdos (SGC) - Repositório de Imagens.....	26
Sistema de pesquisa e filtros.....	27
Carrinho de compras.....	28
Melhoria da Interface e Experiência do Utilizador.....	29
Sistema de Aviso e Erros (Modals).....	29
Demonstração da Plataforma.....	33

Conclusão.....	35
Bibliografia.....	37

Índice de Figuras:

Figura 1.....	5
Figura 2.....	8
Figura 3.....	12
Figura 4.....	15
Figura 5.....	15
Figura 6.....	16
Figura 7.....	16
Figura 8.....	17
Figura 9.....	17
Figura 10.....	17
Figura 11.....	18
Figura 12.....	18
Figura 13.....	21
Figura 14.....	22
Figura 15.....	23
Figura 16.....	26
Figura 17.....	30
Figura 18.....	30
Figura 19.....	30
Figura 20.....	30
Figura 21.....	30
Figura 22.....	31
Figura 23.....	31
Figura 24.....	31
Figura 25.....	31

(Página intencionalmente deixada em branco)

Introdução

O objetivo deste trabalho prático é desenvolver uma aplicação web dinâmica utilizando PHP, HTML e CSS, MySQL com foco na gestão de utilizadores, navegação entre páginas e funcionalidades interativas. Este projeto tem como finalidade aplicar os conhecimentos adquiridos na unidade curricular de Sistemas Multimédia para a Internet, incluindo sessões, formulários, validação de dados e organização de código.

O desenvolvimento foi estruturado em três fases distintas, que permitiram uma abordagem progressiva e sistemática à construção da solução:

1. Análise do Problema

Nesta fase inicial, procedeu-se à identificação dos principais módulos do sistema, das funcionalidades a implementar e das estruturas de dados a utilizar, garantindo uma base sólida para as fases seguintes.

2. Desenvolvimento da Aplicação Web

Com base na análise feita, passou-se à implementação da aplicação, recorrendo a técnicas de programação web que garantissem uma experiência de utilizador fluida, segura e personalizada.

3. Testes e Validação

Por fim, foram realizados testes com o objetivo de comprovar o correto funcionamento da aplicação, assegurando que todos os requisitos funcionais foram cumpridos e que o sistema responde de forma fiável às diferentes interações dos utilizadores.



Figura 1 – Logótipo da plataforma BodyBliss.

(Página intencionalmente deixada em branco)

Estrutura do Projeto

O projeto está organizado segundo uma estrutura modular, com diretórios dedicados a diferentes tipos de conteúdos, todos mostrados na Figura 2:

- A diretoria “api” destaca-se pelo armazenamento dos WebServices desenvolvidos.
- A diretoria “backup_files” é responsável por armazenar certos ficheiros de backup como linhas SQL para utilizar caso a base de dados necessite dar backup.
- A diretoria “imgs” guarda imagens utilizadas na página, como ícones e elementos decorativos.
- A diretoria “lib” é dedicada a bibliotecas e APIs externas utilizadas no sistema. Atualmente, esta diretoria armazena exclusivamente os ficheiros da API PHPMailer e API Dompdf.
- A diretoria “style” contém os ficheiros CSS responsáveis pela apresentação gráfica da aplicação.
- A diretoria “uploads_pdf” armazena temporariamente os PDF automaticamente gerados.
- A diretoria “uploads_profile” armazena as fotos de perfil dos utilizadores após o upload.
- A diretoria “uploads_repository” armazena as fotos atualizadas pelos VIP’s e ADMIN’s na página dedicada a um repositório de partilha de fotos dos eventos realizados.
- A diretoria “xml” contém documentos XML relacionados para o projeto.
- O ficheiro “index.php” funciona como ponto de entrada da aplicação, este é o “controller” da aplicação web.
- Foram ainda criadas várias páginas com prefixo `page_` que representam secções distintas da aplicação, como `page_home.php`, `page_about.php`, `page_catalog.php`, entre outras.
- Outros ficheiros tratam dos processos para o sucesso do funcionamento da página.
- O ficheiro `basic_functions.php` contém funções auxiliares reutilizáveis ao longo do projeto, como verificação de sessões e manipulação de dados.

```


▼ └ bodybliss C:\Users\Miguel\Desktop\IselBackup2025April\ISEL\3oAno\SV
  ▼ └ api
    php cart_products.php
    php checkout.php
    php products.php
  ▼ └ backup_files
    bodybliss.sql
    email_account_sql_query.txt
  > └ imgs
  ▼ └ lib
    > └ dompdf
    > └ PHPMailer
      php lib-mail-v2.php
  ▼ └ style
    main.css
    uploads_pdf
  > └ uploads_profile
  > └ uploads_repository
  ▼ └ xml
    config.xml
  php add_to_cart.php
  php admin_delete_profile_pic.php
  php basic_functions.php
  php consume_products.php
  php db_init.php
  php delete_file.php
  php delete_profile_pic.php
  php delete_unverified.php
  docker-compose.yml
  index.php
  login.php
  logout.php
  page_about.php
  page_account.php
  page_adminzone.php
  page_catalog.php
  page_checkout.php
  page_checkout_success.php
  page_contact.php
  page_header.php
  page_home.php
  page_jesmonite.php
  page_login.php
  page_massagens.php
  page_register.php
  page_repository.php
  page_velas.php
  page_view_cart.php
  register.php
  remove_from_cart.php
  test_db.php
  upload_profile.php
  upload_repository.php
  verify_email.php
  version_checker.php


```

Figura 2 – Módulo Estrutural.

Modelo de Dados “bodybliss”

A base de dados desenvolvida para este projeto constitui uma parte fundamental da aplicação BodyBliss, sendo responsável pela persistência de toda a informação como utilizadores, produtos, encomendas e conteúdos multimédia.

Optou-se pela utilização de MySQL por se tratar de um sistema de gestão de bases de dados leve, fiável e fácil de integrar com aplicações web em PHP. Através deste SGBD, foi possível estruturar os dados de forma relacional, garantindo integridade, consistência e escalabilidade da informação.

A modelação da base de dados seguiu uma lógica normalizada, com recurso a chaves primárias e estrangeiras, o que permitiu estabelecer ligações entre entidades e evitar redundância de dados, Figura 3. Este modelo facilita ainda a execução de consultas eficientes e seguras, sendo ideal para suportar funcionalidades como login, gestão de carrinho de compras, repositório de imagens e histórico de encomendas.

Para além disso, esta abordagem relacional contribui para a manutenção e evolução futura da aplicação, uma vez que permite adicionar novas tabelas ou funcionalidades com mínimo impacto nas existentes.

Tabelas e Campos

Users

Esta tabela guarda todos os utilizadores registados na aplicação. Contém os dados essenciais de identificação, autenticação, verificação de email e tipo de utilizador.

- “id”: Identificador único do utilizador (chave primária).
- “name”: Nome do utilizador.
- “email”: Endereço de email, que deve ser único.
- “password”: Palavra-passe encriptada com “password_hash()”.
- “gender”: Género do utilizador (texto livre).
- “role”: Papel do utilizador no sistema (“client”, “vip” ou “admin”).
- “country”: País correspondente ao utilizador, posteriormente editado.
- “phone”: Número de telemóvel associado ao utilizador, posteriormente editado.
- “profile_pic”: Caminho para a imagem de perfil (por padrão “nullprofile.jpg”).
- “is_verified”: Indica se o utilizador já verificou o email (0 ou 1).
- “verify_token”: Token único enviado por email para verificação.

- “created_at”: Data de criação da conta.

Repository

Guarda imagens enviadas por utilizadores com permissões especiais (como “vip” ou “admin”). Estas imagens podem representar ideias, produtos personalizados, entre outros.

- “id”: Identificador único da imagem (chave primária).
- “user_id”: ID do utilizador que fez o envio (foreign key para “users.id”).
- “file_name”: Nome do ficheiro da imagem armazenada.
- “upload_date”: Data e hora do envio da imagem.

Products

Armazena os produtos disponíveis para compra. Cada produto pertence a uma categoria e tem uma imagem associada.

- “id”: Identificador único do produto (chave primária).
- “name”: Nome do produto.
- “description”: Descrição do produto.
- “price”: Preço do produto.
- “category”: Categoria do produto.
- “image”: Nome do ficheiro da imagem correspondente ao produto.

Order_items

Armazena todos os produtos já comprados no sistema, caso algum “product” seja alterado ou removido, assim, continuamos a conter a informação de compras passadas.

- “id”: Identificador único do produto (chave primária).
- “order_id”: Número da compra.
- “product_name”: Nome do produto comprado.
- “quantity”: Quantidade do produto comprado.
- “price”: Preço da unidade do produto comprado.
- “subtotal”: Preço total de todas as unidades do produto comprado.

Orders

Funciona com o mesmo propósito do “order_items”.

- “id”: Identificador único do produto (chave primária).
- “user_id”: ID do utilizador que efetuou a compra.
- “total”: Total absoluto da compra.
- “order_date”: Data de realização da compra.

Email_accounts

Contém as definições para o envio de emails via SMTP. É usada em conjunto com a biblioteca PHPMailer.

- “id”: Identificador único da conta de email (chave primária).
- “accountName”: Nome descritivo da conta.
- “stpmServer”: Endereço do servidor SMTP.
- “port”: Porta utilizada para o envio.
- “useSSL”: Define se o envio utiliza SSL (1 = sim, 0 = não).
- “timeout”: Tempo máximo de espera pelo envio.
- “loginName”: Nome de utilizador da conta de email.
- “password”: Palavra-passe usada no envio.
- “email”: Endereço de email usado no envio.
- “displayName”: Nome a mostrar como remetente dos emails.

Cart

Representa os itens que cada utilizador adiciona ao carrinho de compras.

Serve de ligação entre os produtos e os utilizadores.

- “id”: Identificador único do registo (chave primária).
- “user_id”: ID do utilizador que adicionou o produto (foreign key para “users.id”).
- “product_id”: ID do produto adicionado (foreign key para “products.id”).
- “quantity”: Quantidade do produto no carrinho.

Relacionamentos entre Tabelas

users → cart:

Um utilizador pode adicionar vários produtos ao seu carrinho. Cada entrada na tabela “cart” associa um “user_id” a um “product_id”, permitindo gerir múltiplos itens por utilizador.

products → cart:

Um produto pode ser adicionado por vários utilizadores diferentes aos seus respetivos carrinhos. Esta relação é representada através do campo “product_id” na tabela “cart”.

users → repository:

Cada utilizador com permissões especiais pode enviar múltiplas imagens para o repositório. A relação é estabelecida através do campo “user_id” na tabela “repository”.

users → orders:

Um utilizador pode realizar várias encomendas ao longo do tempo. Esta ligação é feita através do campo “user_id” na tabela “orders”, associando cada encomenda ao utilizador que a efetuou.

orders → order_items:

Cada encomenda (orders) pode conter múltiplos itens (order_items). O campo “order_id” na tabela “order_items” identifica a que encomenda pertencem os produtos registados.

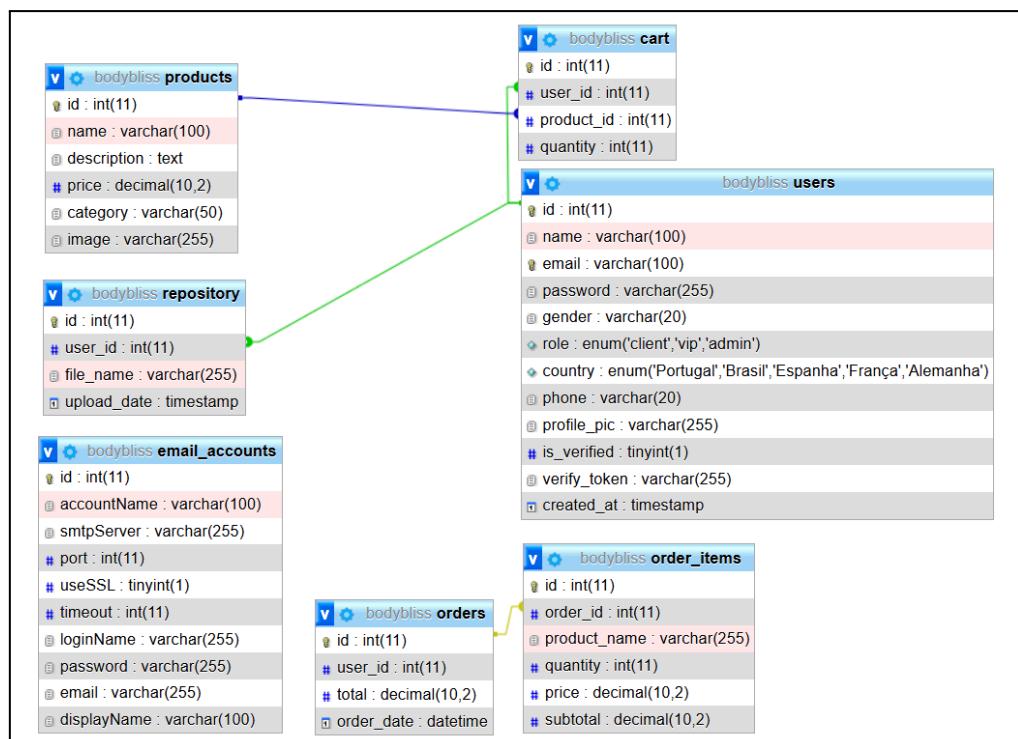


Figura 3 – Ligações entre Tabelas.

Enquadramento com Plataformas Semelhantes

A plataforma BodyBliss apresenta funcionalidades comuns a muitas soluções de e-commerce modernas, como o registo de utilizadores com perfil personalizado, carrinho de compras persistente, checkout com geração de fatura e upload de conteúdos multimédia. Estas funcionalidades são frequentemente encontradas em plataformas como a Etsy, Shopify ou Wix, que também privilegiam a experiência do utilizador e a flexibilidade na gestão de conteúdos. O projeto procura assim seguir boas práticas já testadas em sistemas reais, adaptando-as a um contexto académico e formativo.

(Página intencionalmente deixada em branco)

Mockups

Página Inicial

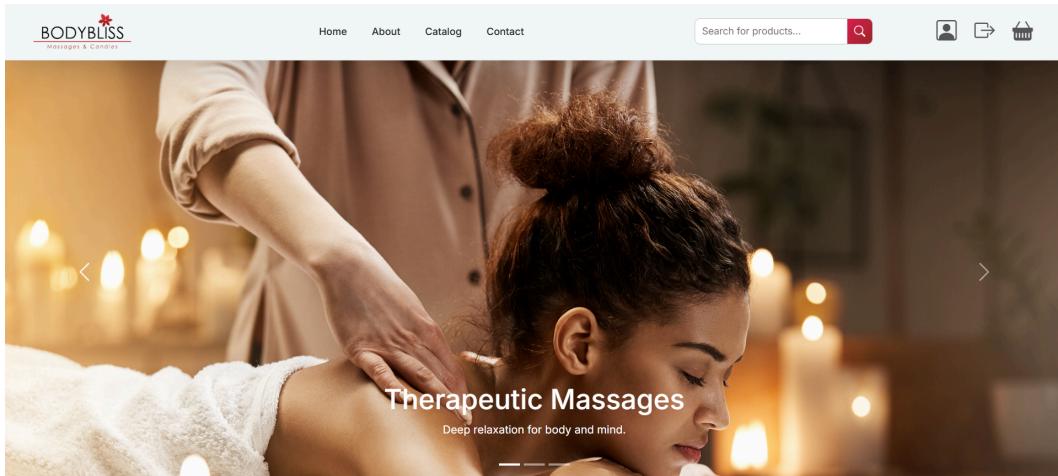


Figura 4 – “page_home.php”.

Página About

“Promoting well-being is not a luxury — it’s a necessity.”

BODYBLISS was born from my passion for Therapeutic Massages and the creation of unique handmade pieces, like aromatic candles and Jesmonite décor. The name combines “BODY” (body, self) with “BLISS” (ecstasy, happiness), reflecting the purpose of offering well-being, relaxation, and harmony. Each candle is crafted with care and tailored to the preferences of the person receiving it — more than just decoration, they are gifts with soul. Whether through a massage with natural oils or by lighting a custom-made BODYBLISS candle, the goal is always the same: to help you slow down and reconnect with your inner self. Allow yourself this experience. **BODYBLISS YOURSELF** ▾

What we offer:

- PERSONALIZED AROMATIC CANDLES
- THERAPEUTIC MASSAGES WITH NATURAL OILS
- UNIQUE AND HANDMADE GIFTS

“I created BODYBLISS to share what brings me peace — I hope it brings you peace too.”

[Explore Catalog](#)

Figura 5 – “page_about.php”.

Página Catalog

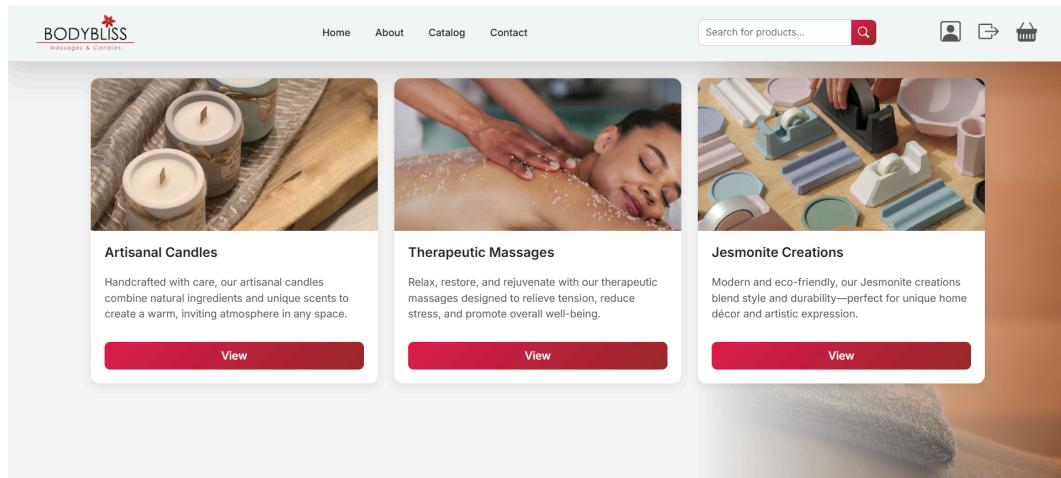


Figura 6 – “page_catalog.php”.

Página Contact

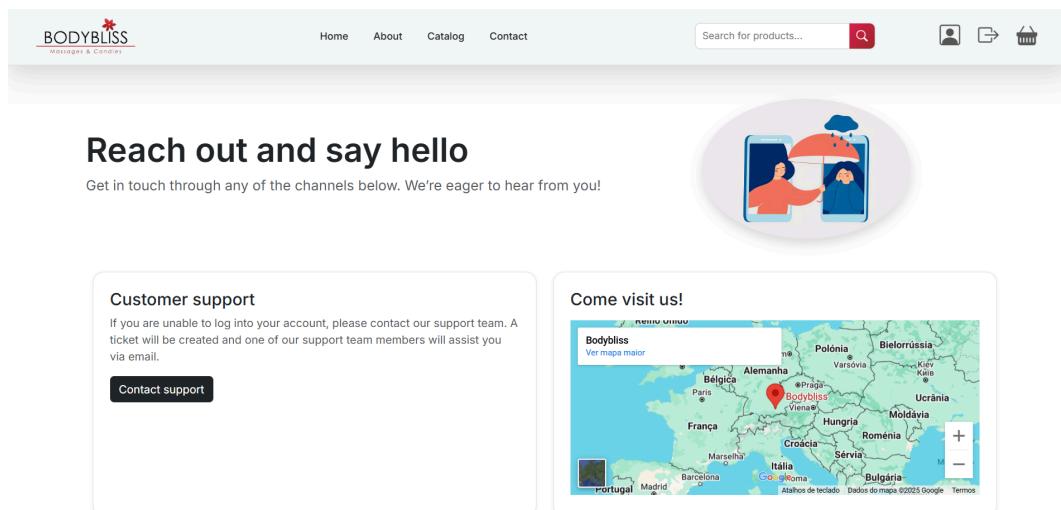


Figura 7 – “page_contact.php”.

Página Login

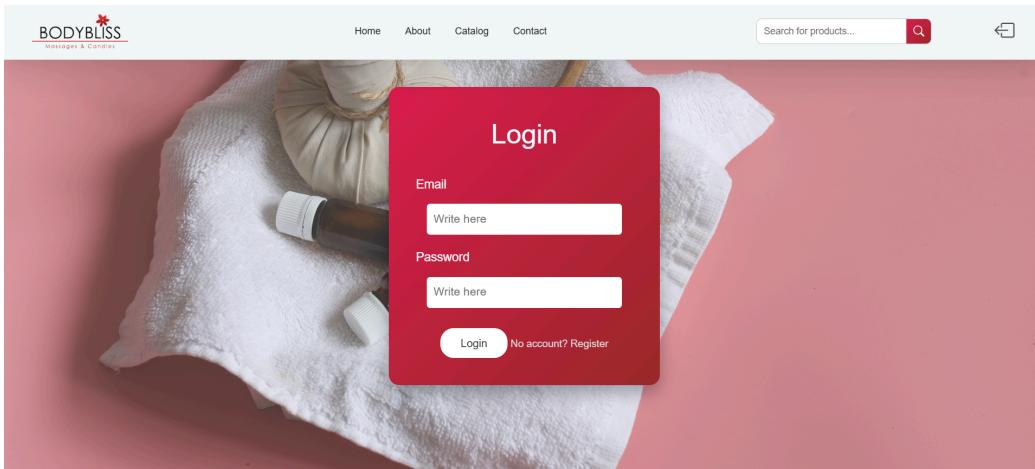


Figura 8 – “page_login.php”.

Página Register

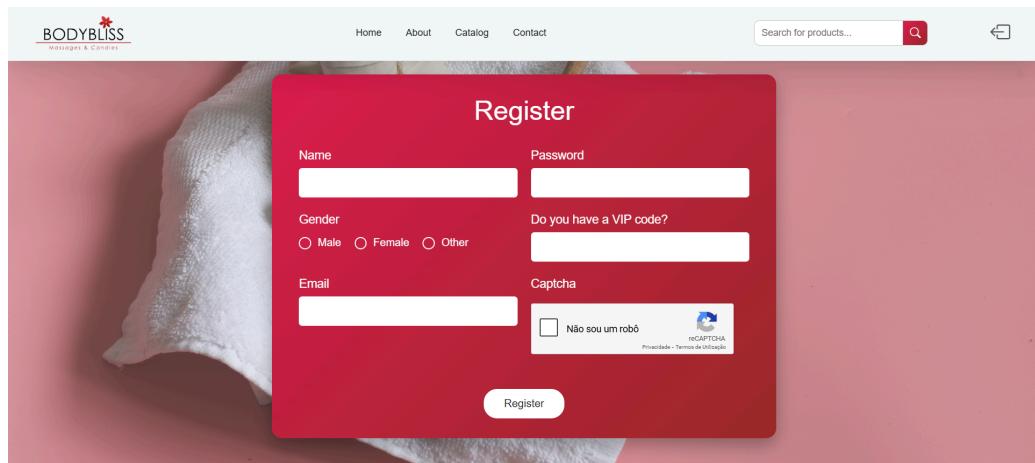


Figura 9 – “page_register.php”.

Página Account

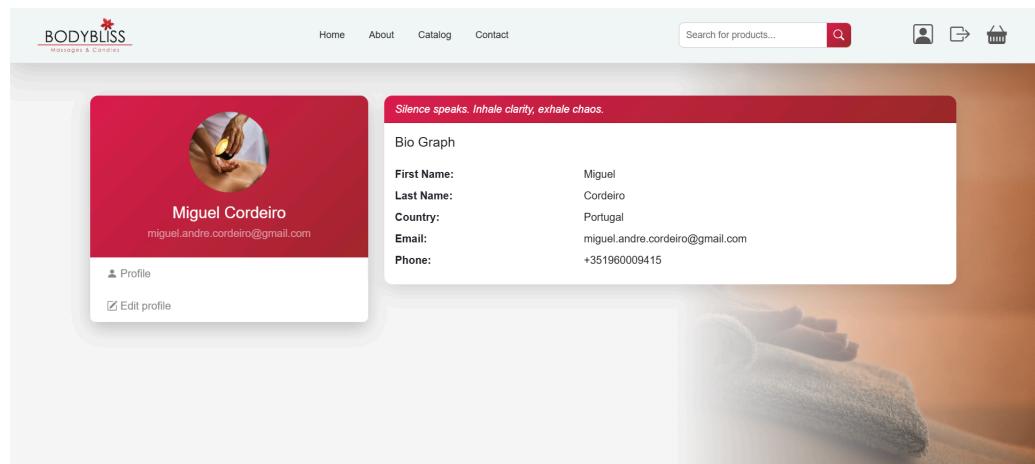


Figura 10 – “page_account.php”.

Página Cart (com mouse no hover cart)

The screenshot shows the BODYBLISS website's shopping cart page. At the top, there is a navigation bar with links to Home, About, Catalog, and Contact. A search bar and user icons are also present. The main content area is titled "Your Shopping Cart". It displays four items in the cart:

- Aromatherapy Candle Massage**: Price: 35.00 €, Quantity: 3, Subtotal: 105.00 €.
- Decoration**: Price: 17.75 €, Quantity: 1, Subtotal: 17.75 €.
- Bubble Cube Candle**: Price: 12.99 €, Quantity: 7, Subtotal: 90.93 €.
- Christmas Tree Candle**: Price: 9.50 €, Quantity: 1, Subtotal: 9.50 €.

The total amount is **Total: 223.18 €**. On the right side of the cart summary, there is a modal window with a light gray background containing the same cart items and their details. Below the modal are two buttons: "View Cart" (gray) and "Checkout" (red).

Figura 11 – “page_cart.php”.

Página Checkout

The screenshot shows the BODYBLISS website's checkout page. The top navigation bar and search bar are identical to the previous screenshot. The main content area starts with a section titled "Cart Resume:" which lists the same four items from the cart page.

Below the resume, the total amount is **Total: 223.18€**. A large red button labeled "Complete Purchase" is centered below the total.

Underneath the purchase button is a "Credit Card" section with the subtext "Secure and encrypted". This section includes fields for "Card Name", "Card Number", "MM/AA", and "CVC/CVV". To the right of these fields are logos for VISA, MasterCard, American Express, and Discover.

At the bottom of the credit card form is a red "Pay" button.

Figura 12 – “page_checkout.php”.

Funcionalidades Implementadas

Mecanismo de Redirecionamento (PID “Page ID”)

Para o mecanismo de controle de páginas é usada a variável (REQUEST) \$pid, inserida no url. Esta variável é usada para o "controller" script (index.php) saber que página deverá mostrar, através dos "include" do php.

Dentro do Código o pid tambem pode ser mudado, como por exemplo se houver erro, fazendo o override da página.

Optou-se por usar este mecanismo, em vez do header(), que obriga o browser a fazer um redirect para uma nova página por uma razão de eficiência. No entanto, também pode ser usado o header() em casos que queremos modificar o url no browser do cliente.

Gestão de Utilizadores

Para implementar o sistema de autenticação, foram criados quatro ficheiros: "login.php", "register.php", "verify_email.php", "delete_unverified.php". Os dois primeiros interagem com a base de dados MySQL através de queries preparadas, de forma a garantir maior segurança e proteção contra ataques como "SQL Injection". Já os ficheiros "verify_email.php" e "delete_unverified.php" têm como objetivo reforçar a segurança do sistema, permitindo a verificação de emails para evitar registos automatizados (bots) e a limpeza periódica da base de dados através da remoção de contas não verificadas.

Após o login com sucesso, é iniciada uma sessão PHP onde são guardadas informações do utilizador como o seu ID, nome, email, role e foto de perfil. Estes dados ficam disponíveis enquanto o utilizador estiver com a sessão ativa, permitindo que a plataforma adapte a navegação, funcionalidades e permissões de forma personalizada. Esta abordagem evita que o utilizador tenha de fazer login a cada mudança de página, garantindo uma experiência mais fluida e segura.

Durante o registo (register.php):

- O utilizador preenche um formulário com os seus dados pessoais.
- É verificada a unicidade do email, impedindo regtos duplicados.
- Existe um campo opcional para introdução de um código de acesso especial. Se esse código for válido, o utilizador recebe o papel (*role*) de "vip", caso contrário, será atribuído o papel padrão de "client".

- Os dados são inseridos na base de dados, incluindo o campo (role) na tabela de utilizadores.
- As passwords são armazenadas de forma encriptada usando `password_hash()`, seguindo boas práticas de segurança.
- É armazenado na database a data de criação da conta, para posteriormente ser utilizada, como por exemplo, no `delete_unverified.php`

Durante o login (`login.php`):

- O utilizador submete o email e password.
- A password submetida é verificada com `password_verify()` para garantir que coincide com o valor encriptado guardado na base de dados.
- Se os dados estiverem corretos, é iniciada uma sessão PHP, onde são guardados o id, nome, email e role do utilizador.
- A partir dessa sessão, o sistema reconhece o utilizador e adapta a interface e permissões de acordo com o seu papel (role).

Durante a verificação de email (`verify_email.php`):

- Esta página é acionada quando o utilizador clica no link de verificação recebido por email após o registo.
- O link contém um *token* único gerado aleatoriamente no momento do registo.
- Ao aceder à página, o sistema compara o *token* recebido com o existente na base de dados.
- Se a correspondência for válida, o campo “`is_verified`” do utilizador é atualizado para 1, e o *token* é removido da base de dados, permitindo assim que o utilizador possa efetuar login na plataforma.

Remoção de contas não verificadas (`delete_unverified.php`):

- Este ficheiro elimina automaticamente utilizadores que não confirmaram o email no prazo de 7 dias.
- Usa uma query que verifica se “`is_verified = 0`” e se a data de criação “`created_at`” tem mais de 7 dias.
- Ajuda a manter a base de dados limpa e livre de registos inválidos.
- Foi criada uma query adicional (comentada) para fins de debug, permitindo testar o funcionamento da limpeza automática num intervalo mais curto (1 minuto), em vez dos 7 dias definidos na versão final.

WebServices Desenvolvidos

Com o objetivo de promover a interoperabilidade da aplicação e preparar o sistema para futuras integrações, foram desenvolvidos vários WebServices RESTful que expõem dados e funcionalidades essenciais da plataforma BodyBliss em formato JSON.

Entre todos estes endpoints, destaca-se o checkout.php, pelo seu nível de complexidade, este atua como ponto central para o processo de finalização de compras na plataforma. Este endpoint recebe os dados da encomenda via JSON, valida os campos, regista a encomenda e os respetivos produtos na base de dados, e envia por email (Figura 14) um comprovativo em PDF gerado automaticamente com a biblioteca Dompdf (Figura 15).

Para facilitar os testes e demonstração do funcionamento do sistema, foi implementado um mecanismo de validação de cartão “hardcoded”. Assim, apenas dados específicos (por exemplo, número de cartão 4242424242424242) são aceites, simulando a resposta de um sistema de pagamento real. Isto garante segurança e previsibilidade durante o desenvolvimento e apresentação do projeto.

Além disso, após a encomenda ser registada, o carrinho do utilizador é automaticamente limpo e o ficheiro PDF da fatura é temporariamente guardado, anexado ao email e, posteriormente, eliminado do servidor.

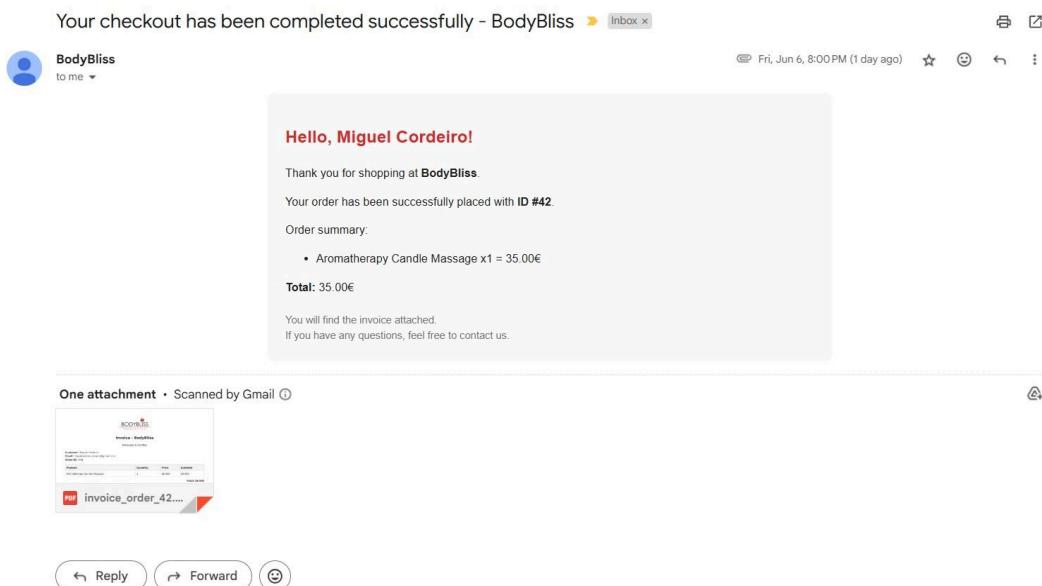


Figura 13 – Email da compra.



Figura 14 – Pdf anexado no email da compra.

Adicionalmente, foi criado o products.php, responsável por fornecer a lista de produtos disponíveis na base de dados. Este serviço estabelece ligação à base de dados, consulta a tabela de produtos e disponibiliza os dados de forma estruturada, facilitando o consumo por aplicações externas.

E por último, implementou-se o endpoint cart_products que devolve a lista de produtos de um determinado user.

Estes WebServices representam uma evolução na arquitetura da aplicação, garantindo maior flexibilidade e escalabilidade, estes retornam a informação em JSON.

Uso de API's

Neste projeto foram integradas diversas APIs externas que complementam e reforçam a funcionalidade e segurança da plataforma BodyBliss. Estas APIs permitem estender as capacidades da aplicação sem necessidade de desenvolver funcionalidades complexas de raiz, garantindo maior fiabilidade, segurança e uma melhor experiência de utilização.

Google reCAPTCHA v2:

A API reCAPTCHA da Google foi integrada na página de registo com o objetivo de evitar submissões automáticas por bots. Esta verificação é feita através de um token gerado pelo widget do reCAPTCHA, que é validado no backend junto dos servidores da Google. Esta proteção assegura que apenas utilizadores humanos conseguem criar contas na plataforma.

PHPMailer (via SMTP):

Para o envio de emails (como a verificação de conta), foi utilizada a biblioteca PHPMailer, uma API amplamente reconhecida em PHP para envio de emails através de servidores SMTP autenticados. Esta integração permite o envio seguro e fiável de emails com mensagens personalizadas e suporte a HTML, garantindo que o utilizador confirme o seu endereço antes de aceder à conta.

Dompdf:

Para a geração de documentos em formato PDF (como as faturas), foi integrada a biblioteca Dompdf. Esta API em PHP permite converter código HTML e CSS em ficheiros PDF, possibilitando a criação dinâmica de documentos com o mesmo estilo visual do site. A integração facilita a exportação e distribuição de informação estruturada ao utilizador de forma prática e compatível com diversos dispositivos.

Google Maps Embed API

Através da API de incorporação do Google Maps, foi integrado um mapa interativo na secção de contacto da plataforma. Esta funcionalidade permite aos utilizadores visualizar rapidamente a localização da marca BodyBliss (Figura 16), sem necessidade de desenvolvimento de funcionalidades geográficas próprias. A integração é feita com recurso a um <iframe> personalizado gerado pela própria Google.

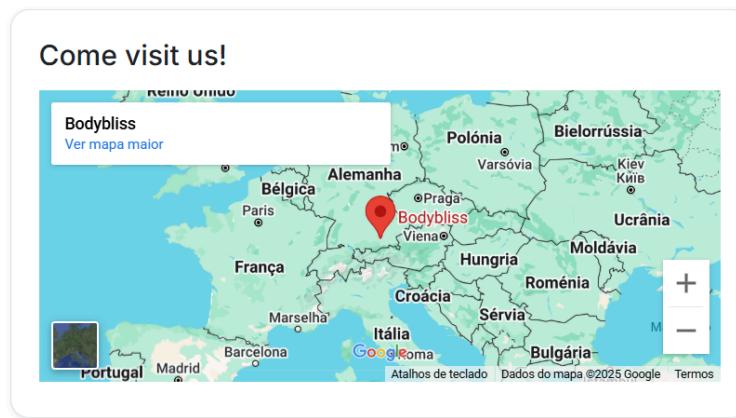


Figura 15 – Google Maps Localização.

Envio email de verificação

Após o registo do utilizador, o sistema procede automaticamente ao envio de um email de verificação. Esta funcionalidade foi implementada com o objetivo de garantir que o endereço de email fornecido é legítimo, evitando assim contas falsas e promovendo a segurança da plataforma.

O processo funciona da seguinte forma:

- Após a submissão do formulário de registo, é gerado um token aleatório e único através da função “openssl_random_pseudo_bytes()”, que é guardado na base de dados na coluna “verify_token” da tabela de utilizadores.
- É posteriormente construído um link de verificação que inclui esse token como parâmetro “verify_email.php?token=...”.
- Utilizando a biblioteca PHPMailer, é enviado um email com uma mensagem HTML personalizada que convida o utilizador a clicar no botão “Verificar Email”.
- O utilizador dispõe de 7 dias para completar a verificação. Caso contrário, a conta será removida automaticamente por uma rotina de limpeza “delete_unverified.php” como já referido no relatório.

Hierarquia de diferentes utilizadores

O projeto admite 4 perfis de utilização diferente:

- Visitante
- Cliente (role=client)
- Contribuidor (role=vip)
- Administrador (role=admin)

Na tabela da DataBase foi adicionado uma coluna “role” com cada tipo de utilizador diferente exceto o perfil Visitante, este basta aceder à plataforma sem fazer login. Ao fazer “register” existe um campo para preencher um código que dá a possibilidade ao utilizador pertencer a um “role=vip” caso esse código introduzido não seja o correto o programa define “role=client”, automaticamente. Só desenvolvedores de código conseguem criar contas com “role=admin”.

Permissões de cada role:

role
admin
vip
client
vip

client

- efetuar compras
- mudar a foto de perfil

vip (contribuidor)=

- efetuar compras
- mudar a foto de perfil
- acesso à página de repositório (download e upload de fotos dos eventos)

admin=

- efetuar compras
- mudar a foto de perfil
- acesso à página de repositório (download e upload de fotos dos eventos)

-efetuar alterações nas imagens de todos os utilizadores
(Admin Zone)

Atualização e Gestão da imagem de perfil

Implementou-se a possibilidade de qualquer utilizador editar a sua informação (número de telemóvel e país de residência), assim como, adicionar, atualizar ou remover a imagem de perfil, que será posteriormente guardada numa diretoria do projeto e o nome do ficheiro guardado na linha do respetivo ID. Para simplificação optou-se por gravar cada nome do ficheiro com “p” concatenado com o ID e concatenado também com o file type, assim são sempre únicos e não será necessário apagar o anterior em caso de update. Esta funcionalidade está acessível através da página de conta do utilizador (page_account.php), sendo suportada pelos ficheiros upload_profile.php e delete_profile_pic.php.

Upload / Atualização da Imagem:

- O utilizador tem à disposição um formulário onde pode selecionar uma imagem do seu dispositivo.
- Ao submeter, o ficheiro é enviado para o servidor e processado pelo script upload_profile_pic.php.
- O sistema realiza uma validação do formato do ficheiro, aceitando apenas formatos como .jpg, .jpeg ou png, de forma a garantir compatibilidade e segurança.
- É definido um nome único para o ficheiro com base no ID do utilizador, seguindo o padrão p<ID>.<filetype>. Desta forma, evita-se qualquer conflito entre ficheiros e assegura-se que cada utilizador tem uma imagem exclusiva. Desta forma cada utilizador tem sempre o mesmo nome para a imagem de perfil e facilita a remoção da antiga em caso de atualização de imagem.
- O ficheiro é fisicamente guardado na diretoria uploads_profile.
- O nome do ficheiro é atualizado na base de dados, associado ao utilizador correspondente, para que seja possível apresentar corretamente a imagem nas páginas do site.

Utilização XML

Desenvolveu-se um ficheiro xml (config.xml) onde estão as credenciais para utilização na ligação a database (Figura 17).

```
<?xml version="1.0" encoding="utf-8" ?>
<config>
    <database>
        <servername>localhost</servername>
        <port>3306</port>
        <username>root</username>
        <pass></pass>
        <dbname>bodybliss</dbname>
    </database>
</config>
```

Figura 16 – “config.xml”.

Sistema de Gestão de Conteúdos (SGC) - Repositório de Imagens

Foram desenvolvidas duas funcionalidades para gestão de conteúdos multimédia, sendo a primeira já abordada em cima sobre a alteração de imagem de perfil.

Com o objetivo de criar uma plataforma colaborativa em conteúdos multimédia, foi desenvolvida uma funcionalidade de repositório multimédia acessível apenas a utilizadores com o papel de Contribuidor (role=vip) ou Administrador (role=admin).

Esta funcionalidade foi integrada na página page_repository.php e permite aos utilizadores autorizados realizar o upload e download de imagens relacionadas com eventos ou outras atividades da plataforma.

Estrutura e Funcionamento

- A interface apresenta uma galeria com todas as imagens previamente carregadas, ordenadas por data de envio.
- Através de um formulário simples, os utilizadores podem selecionar uma imagem do seu dispositivo e submetê-la para o servidor.

- O script upload_repository.php trata do envio da imagem:
 - Gera um nome único baseado no id do utilizador e usa um uniqid() para o utilizador poder dar upload de várias imagens de modo a que os paths não fiquem iguais.
 - Move o ficheiro para a diretoria dedicada uploads_repository.
- Os ficheiros ficam imediatamente disponíveis para download público, permitindo a partilha de conteúdos entre membros do mesmo grupo (vip/admins).

Sistema de pesquisa e filtros

Foi implementado um sistema de pesquisa na aplicação, disponível no cabeçalho da página (page_header.php), com o objetivo de permitir aos utilizadores localizar facilmente produtos de interesse com base em múltiplos critérios.

A barra de pesquisa permite a introdução de termos de pesquisa livres, ao mesmo tempo que oferece filtros adicionais através de um menu suspenso interativo. Este menu apresenta as seguintes opções de filtragem:

- Categoria: Permite restringir os resultados às categorias "Velas Artesanais", "Massagens Terapêuticas" ou "Criações em Jesmonite";
- Preço Mínimo e Máximo: Possibilita definir um intervalo de preços, oferecendo maior controlo na pesquisa de produtos ajustados ao orçamento do utilizador.

Para garantir usabilidade e evitar submissões desnecessárias, o botão de pesquisa permanece desativado até que pelo menos um dos campos (termo de pesquisa ou filtros) seja preenchido. Este comportamento é controlado por JavaScript, o que contribui para uma experiência de navegação mais fluida e responsiva.

Além disso, foi definido que todas as pesquisas redirecionam automaticamente para a página catalog, onde são apresentados os resultados com base nos critérios definidos. Esta abordagem modular permite manter a interface limpa, reutilizar código e facilitar futuras expansões, como ordenação por popularidade ou integração com web services.

Carrinho de compras

A funcionalidade de carrinho de compras foi implementada de forma a permitir que os utilizadores registados possam adicionar produtos ao seu carrinho de forma dinâmica e persistente.

Para estruturar esta funcionalidade, foram criadas relações entre três tabelas da base de dados:

- users: representa os utilizadores.
- products: contém a informação dos produtos disponíveis na loja.
- cart: relaciona utilizadores com produtos, armazenando os produtos que cada utilizador adicionou ao seu carrinho, bem como a quantidade de cada produto.

O funcionamento baseia-se nas seguintes operações:

- Quando o utilizador clica em “Adicionar ao carrinho”, o sistema verifica se o produto já existe no carrinho do utilizador. Se sim, incrementa a quantidade. Se não, insere um novo registo com “quantity = 1”.
- O carrinho está associado ao utilizador autenticado através do “user_id”, garantindo que cada cliente tem o seu próprio carrinho independente.
- No cabeçalho da aplicação é apresentado um dropdown interativo com o resumo do carrinho (produtos e total), usando PHP e consultas “JOIN” entre “cart” e “products”.
- O sistema permite também visualizar uma página dedicada ao carrinho, com opção futura de checkout.

Este modelo com três tabelas torna o sistema mais organizado e fácil de manter. Como cada utilizador tem o seu próprio carrinho, é possível adicionar vários produtos diferentes sem complicações. Além disso, ao ligar as tabelas com “foreign keys”, evitam-se repetições desnecessárias e garante-se que os dados entre utilizadores e produtos ficam sempre bem ligados.

Melhoria da Interface e Experiência do Utilizador

Com o objetivo de proporcionar uma navegação fluida e visualmente apelativa, a aplicação foi desenvolvida com recurso ao Bootstrap 5.3.0, complementado com a biblioteca Bootstrap Icons 1.11.1 e estilos personalizados definidos no ficheiro "style/main.css".

A utilização do Bootstrap permitiu uma assegurar a compatibilidade com diversos dispositivos e tamanhos de ecrã. Foram usados componentes como cards, modais, dropdowns, botões estilizados, grids, e formulários validados, que contribuíram para uma interface coerente e profissional.

Destaca-se ainda a implementação de modals de feedback visual (sucesso, erro, avisos), com ícones intuitivos e cores consistentes, que oferecem ao utilizador respostas claras às suas ações. A integração com CSS personalizado permitiu introduzir elementos como fundos com sobreposição de imagem, botões customizados e transições suaves, reforçando a identidade visual da marca.

No conjunto, estas escolhas visam não apenas tornar a aplicação visualmente agradável, mas também melhorar a usabilidade, acessibilidade e confiança do utilizador ao navegar e interagir com a plataforma.

Sistema de Aviso e Erros (Modals)

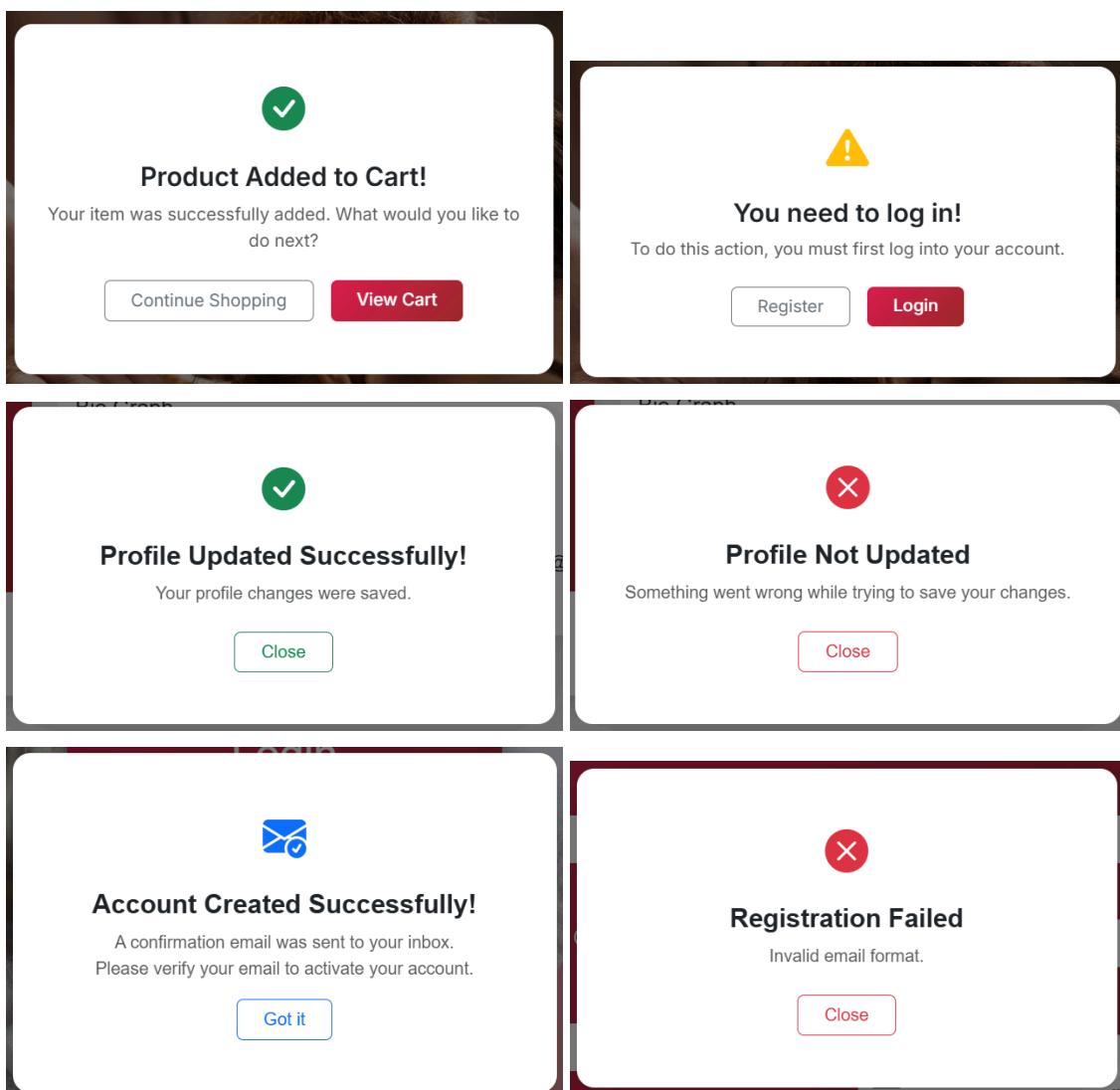
Para melhorar a comunicação com o utilizador, foi criado um sistema de modals de aviso e erro centralizado no ficheiro index.php, que é incluído em todas as páginas da aplicação. Assim, os modals estão sempre disponíveis para serem ativados conforme necessário.

Estes modals são ativados automaticamente através de variáveis de sessão, bastando definir a respectiva variável como true (ou com uma mensagem). Por exemplo:

- “\$_SESSION['s_m_cart']” mostra o aviso de produto adicionado ao carrinho (Figura 18);
- “\$_SESSION['s_m_need_login']” mostra o aviso quando o cliente não autenticado tenta adicionar produtos ao carrinho (Figura 19).
- “\$_SESSION['profile_update']” mostra o aviso a dizer que o perfil foi atualizado (Figura 20).

- “\$_SESSION['profile_update_error']” mostra o erro de atualização de perfil (Figura 21).
- “\$_SESSION['registration_success']” confirma a criação da conta com sucesso (Figura 22);
- “\$_SESSION['registration_error']” mostra o erro de registro (Figura 23);
- “\$_SESSION['login_error']” mostra o erro de login com boas práticas de segurança, apenas referindo que o Email ou a Password está incorreta (Figura 24);
- “\$_SESSION['email_not_verified']” mostra ao cliente a não possibilidade de efetuar o login, por este ainda não estar verificado (Figura 25);
- “\$_SESSION['invalid_card']” mostra ao user que os dados do cartão estão incorretos para efetuar o checkout (Figura 26).

Este sistema é simples de manter e permite dar feedback imediato ao utilizador, com ícones e mensagens visuais claras, melhorando a experiência geral na plataforma.



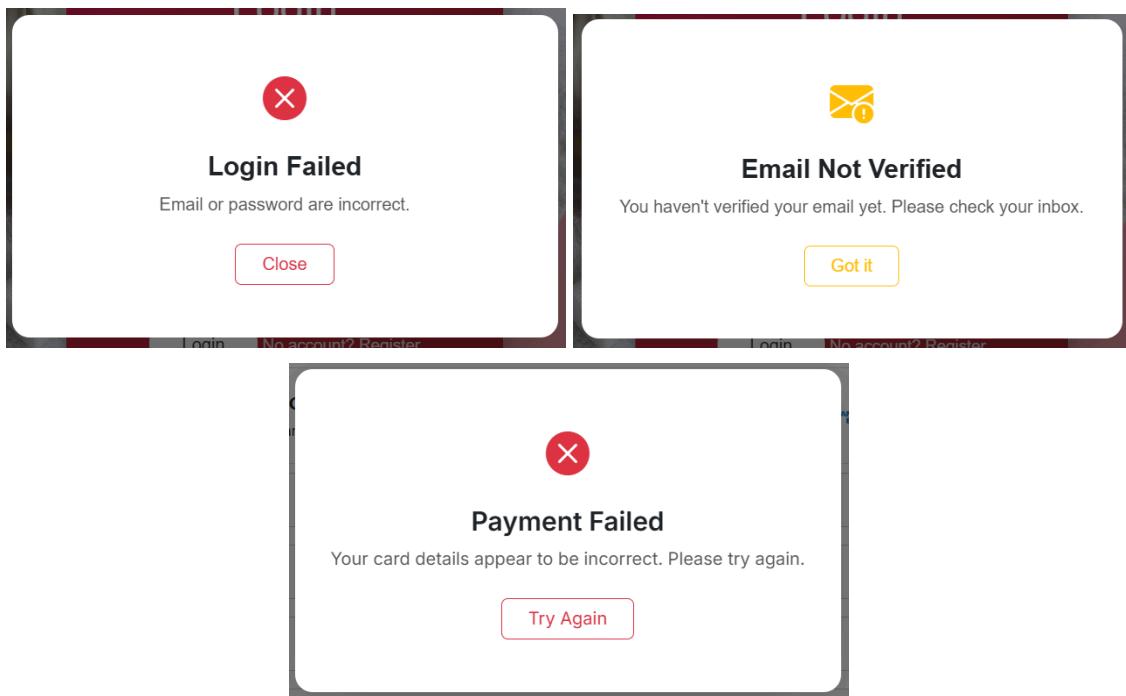


Figura 17 à Figura 25 – Modals.

(Página intencionalmente deixada em branco)

Demonstração da Plataforma

Para ilustrar o funcionamento da aplicação desenvolvida, foi criado um vídeo de demonstração que apresenta as principais funcionalidades e a interação do utilizador com a plataforma web.

A demonstração está disponível no seguinte link:

<https://www.youtube.com/watch?v=NzBTm6lF4g>

Este vídeo permite visualizar de forma clara o percurso do utilizador, desde o registo e login, até à navegação entre páginas, utilização do carrinho de compras, processo de checkout e personalização do perfil.

(Página intencionalmente deixada em branco)

Conclusão

O projeto BodyBliss constituiu uma oportunidade prática para aplicar de forma integrada os conhecimentos adquiridos na unidade curricular de Sistemas Multimédia para a Internet. Através do desenvolvimento de uma aplicação web funcional, modular e segura, foram trabalhados conceitos essenciais como programação em PHP, gestão de sessões, comunicação com base de dados MySQL, integração de WebServices externos e Sistema de Gestão de Conteúdos (SGC).

A plataforma destaca-se pela sua estrutura organizada, interface moderna e intuitiva, e pela implementação de funcionalidades completas como registo e autenticação de utilizadores com verificação por email, gestão de perfis com upload de imagem, repositório colaborativo multimédia, carrinho de compras, sistema de checkout com geração automática de faturas em PDF e envio por email.

A utilização do framework Bootstrap 5, combinada com CSS personalizado, permitiu criar uma experiência de navegação consistente, responsiva e agradável. Além disso, foram aplicadas boas práticas de segurança e reutilização de código, nomeadamente através do uso de queries preparadas, modais de feedback centralizados e componentes reutilizáveis.

Embora o sistema esteja funcional e cumpra os requisitos propostos, foram identificadas oportunidades para evolução futura, como a introdução de pagamentos reais, melhorias ao nível da pesquisa inteligente, maior personalização da interface e reforço da segurança avançada.

Em suma, o desenvolvimento deste projeto permitiu consolidar competências técnicas e reforçar a capacidade de construir soluções web completas, escaláveis e centradas na experiência do utilizador, aproximando-se de um produto real preparado para ser utilizado num contexto profissional.

(Página intencionalmente deixada em branco)

Bibliografia

Durante o desenvolvimento do projeto BodyBliss, foram consultadas diversas fontes de referência que auxiliaram na implementação técnica, integração de bibliotecas e boas práticas de desenvolvimento web. Abaixo encontram-se listadas as principais ferramentas, APIs e documentação oficial utilizadas ao longo do trabalho:

- <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- <https://icons.getbootstrap.com/>
- <https://github.com/PHPMailer/PHPMailer>
- <https://github.com/dompdf/dompdf>
- https://cloud.google.com/security/products/recaptcha?hl=pt_br
- <https://developers.google.com/maps/documentation/embed/get-started?hl=pt-br>
- <https://www.w3schools.com/>
- <https://cssgradient.io/>
- <https://www.freepik.com/>
- <https://bigjpg.com/>
- <https://onlinepngtools.com/create-transparent-png>