



Departamento de
Informática

Little Bricks

Plataformas e Serviços X-Ops

David Gomes | 48537
Karina Trompak | 48059

Índice

Introdução	3
Contextualização da aplicação selecionada	4
Metodologia	6
Requisitos Funcionais e Não Funcionais	6
Construção do Projeto	8
Tutorial e configuração da Google Cloud Platform	11
Adaptações no projeto original	14
Resultados e Dificuldades	16
Webgrafia	20

Índice de Ilustrações

Figura 1 – Página inicial do Site Antigo	4
Figura 2 - Loja do Site Antigo	4
Figura 3 - Criação do Repositório no GitHub	8
Figura 4 - Configuração de um ambiente local com o XAMPP	8
Figura 5 - Criação da base de dados no HeidiSQL	9
Figura 6 - Criação da tabela "clientes"	9
Figura 7 - Criação da tabela "contactos"	10
Figura 8 - Criação da tabela "encomendas"	10
Figura 9 - Criação da tabela "encomenda_produto"	10
Figura 10 - Criação da tabela "produtos"	11
Figura 11 - Criação do Repositório	11
Figura 12 - Repositório	11
Figura 13 - Repositório com alguns ficheiros	12
Figura 14 - Criação de um Trigger	13
Figura 15 - Ativação dos Serviços	13
Figura 16 - Criação do Bucket	14
Figura 17 - Nova Página da Little Bricks	14
Figura 18 - Loja da Little Bricks	15
Figura 19 - Novo Rodapé da Little Bricks	15
Figura 20 - Código de Login	17
Figura 21 - Código para buscar informações da encomenda	18
Figura 22 - Ativação da função segurança	18
Figura 23 - Código para dar submit do contacto	19

Introdução

A evolução tecnológica e a adaptabilidade são pilares fundamentais no cenário atual, onde a transformação digital redefine constantemente os paradigmas do mercado. Este projeto, desenvolvido no âmbito da cadeira Plataformas e Serviços O-Xps, emerge como um reflexo dessa necessidade de inovação contínua e agilidade operacional.

Partindo de uma premissa desafiadora, este trabalho propõe-se a reinventar e reestruturar uma aplicação web originalmente concebida como uma loja de automóveis. Numa reconfiguração desafiante, essa plataforma foi remodelada para se tornar um ambiente vibrante e acessível para entusiastas de legos, demonstrando não apenas a versatilidade do código-fonte original, mas também a capacidade de adaptação e ampliação do escopo funcional.

A base deste projeto reside não apenas na transformação do conteúdo e da funcionalidade da aplicação, mas também na modernização da sua infraestrutura. A migração para a *Google Cloud Platform* foi um passo crucial, proporcionando um ambiente robusto e escalável para a aplicação. Apesar de inicialmente, como grupo, tentarmos adaptar os nossos conhecimentos e habilidades no ambiente do *Microsoft Azure*, este não foi possível de realizar por uma série de questões técnicas. Além disso, a implementação de práticas avançadas de CI/CD tem sido um elemento vital neste processo de atualização, aproveitando tecnologias como pipelines, Docker e *Kubernetes* para garantir a consistência, confiabilidade e eficiência operacional.

Para manter a transparência, a colaboração e o controle de versões durante o desenvolvimento do projeto, foi estabelecido um repositório no GitHub, tornando-se o centro de gerenciamento do código-fonte e facilitando o trabalho colaborativo do nosso grupo.

Assim, este projeto não é apenas uma atualização superficial de uma aplicação, mas sim uma demonstração concreta do potencial transformador da tecnologia quando aliada à visão estratégica e à prática técnica. Neste documento, exploraremos em detalhes as etapas, os desafios enfrentados e as soluções adotadas para alcançar os objetivos propostos, ressaltando não apenas o resultado final, mas também o processo iterativo e inovador que impulsionou esta jornada.

Contextualização da aplicação selecionada

A seleção da aplicação, proveniente do contexto da disciplina de Composição Web relativa ao primeiro ano da licenciatura de Informática Web, foi o resultado de uma avaliação prudente entre diferentes projetos disponíveis. Optamos por trabalhar com este projeto em particular, pois consideramos uma das propostas mais interessantes e promissoras do conjunto disponível. Antes de começar as modificações da aplicação, esta apresentava uma página inicial com pouca utilidade ou funcionalidade relevante. As únicas possibilidades existentes eram realizar o login e fazer compras, carecendo de informações úteis para os potenciais clientes.

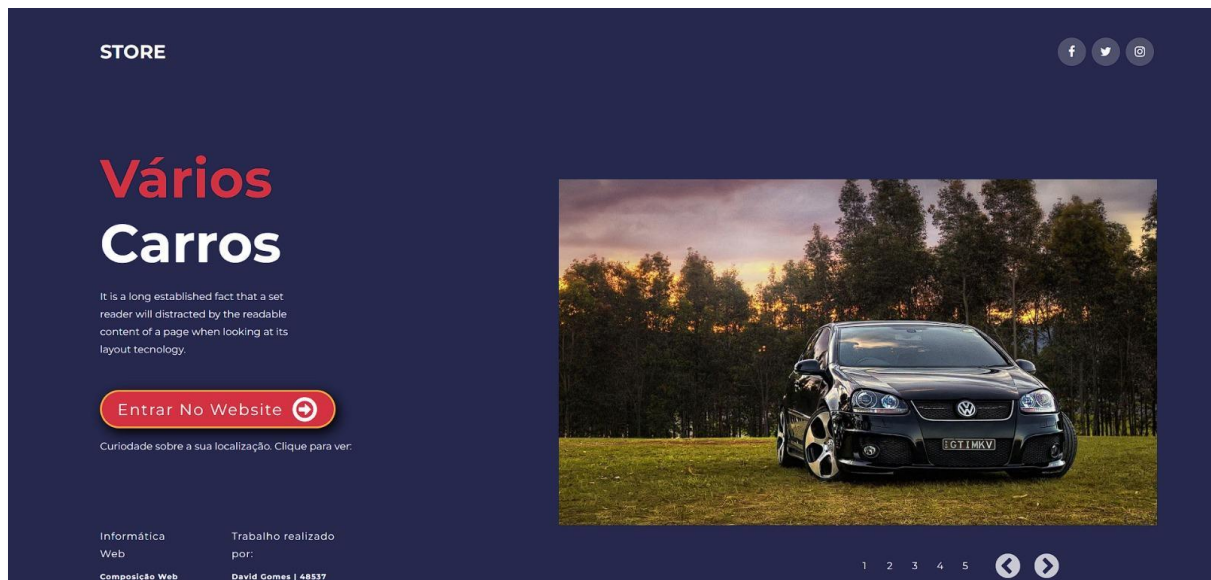


Figura 1 – Página inicial do Site Antigo

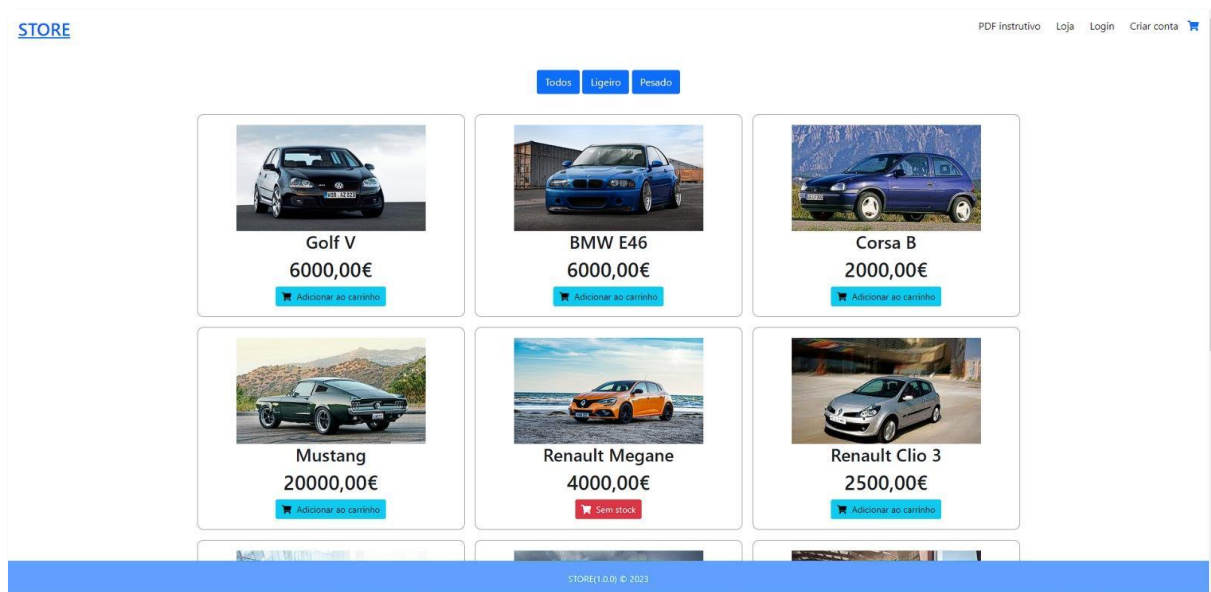


Figura 2 - Loja do Site Antigo

A decisão foi influenciada por diversos fatores. Inicialmente, identificamos que esta aplicação se destacava pela sua estrutura sólida e pela utilização de tecnologias familiares, como PHP, HTML, CSS, Composer e a base de dados desenvolvida na plataforma HeidiSQL. Essa familiaridade com as linguagens e ferramentas utilizadas na sua construção proporcionou uma base sólida para as modificações propostas, simplificando o processo de atualização e otimização.

Para mais, embora a aplicação original tenha sido bem executada, percebemos um potencial considerável para melhorias significativas. A ideia inicial da loja de automóveis era robusta, porém, acreditamos que uma mudança para um conceito de loja online de legos oferecia um campo mais criativo e inexplorado. Este redirecionamento foi motivado pela construção, após extensivas pesquisas, que mostraram a escassez de lojas online que oferecessem uma seleção de brinquedos de qualidade, aliada a um design e funcionalidades bem desenvolvidas.

Ao longo do processo de atualização, uma série de modificações significativas foi implementada, visando aprimorar e adaptar a aplicação original às metas definidas para chegar ao resultado desejado. As principais alterações incluíram:

1. **Alteração do conteúdo do site para venda de legos:** A mudança fundamental na área de negócio, passando de uma loja de carros para uma loja de venda de brinquedos, redefinindo não apenas o catálogo de produtos, mas também o propósito e o público-alvo da aplicação.
2. **Reformulação da página inicial (HTML/CSS):** O redesenho e reestruturação da página para refletir a nova proposta de valor da loja, procurando criar uma interface mais apelativa para os clientes.
3. **Alteração do layout da loja:** A revisão completa do layout da loja, adaptando-a à temática dos brinquedos, garantindo uma experiência de compra mais intuitiva e envolvente.
4. **Criação da página de ajuda:** Desenvolvimento de uma secção dedicada para o suporte e esclarecimento de dúvidas dos clientes, contribuindo para uma experiência mais completa e satisfatória.
5. **Implementação de menu com diferentes categorias:** Adição de um menu que categoriza os produtos, simplificando a navegação e a busca por itens específicos.
6. **Aprimoramento do código de login e verificação de conta:** Correções e melhorias no código responsável pela autenticação de usuários e verificação de contas, visando garantir a segurança e a fluidez do processo de login.

Com isto, ao decidirmos manter as linguagens de desenvolvimento originais, optamos por direcionar a aplicação para um nicho de mercado pouco explorado, contemplando uma oportunidade de oferecer uma proposta única e atrativa num segmento insuficientemente representado.

Metodologia

Antes de iniciar a migração para a *Google Cloud Platform*, houve uma tentativa de migração para a *Microsoft Azure*, porém a falta de conhecimentos e funcionalidades da plataforma não permitiu o desenvolvimento do projeto. Foi conduzida uma análise abrangente da infraestrutura existente da aplicação. Este processo envolveu uma série de etapas detalhadas para compreender a arquitetura, recursos e desafios enfrentados pela aplicação na sua configuração original.

Requisitos Funcionais e Não Funcionais

O nosso sistema de e-commerce é projetado para oferecer uma experiência completa e fluida aos utilizadores durante as suas interações. Os requisitos funcionais deste sistema foram detalhados para garantir a funcionalidade abrangente, permitindo que os utilizadores realizem uma série de ações essenciais para uma experiência de compra satisfatória. Entre estas:

1. Autenticação do usuário:

- Os utilizadores podem criar contas pessoais, fornecendo informações básicas e o endereço de e-mail válido.
- O sistema garante aos utilizadores um login realizado de forma segura, garantindo autenticação para aceder às suas contas.
- A palavra-passe do utilizador, ao fazer o seu registo, será automaticamente encriptada como forma de segurança, e assim enviada para a base de dados.

2. Navegação e seleção de produtos:

- Os utilizadores devem poder navegar pelos produtos disponíveis por categorias, facilitando a busca por itens específicos.
- Deve existir a capacidade de os utilizadores visualizarem detalhes dos produtos, incluindo imagens e preços.

3. Gestão do carrinho de compras:

- Os utilizadores devem ser capazes de adicionar produtos ao carrinho de compras e remover itens conforme desejado.
- O sistema deve atualizar dinamicamente o carrinho para refletir as mudanças feitas pelo usuário.

4. Processo de pagamento e emissão de fatura:

- Deve ser possível aos utilizadores prosseguir para o checkout e fornecer informações de pagamento de forma segura.
- Após a conclusão de compra, o sistema deve gerar automaticamente uma fatura ou recibo para confirmação do pagamento.

5. Comunicação e suporte:

- Os utilizadores devem poder entrar em contacto com a equipa de suporte preenchendo um formulário de contacto online.

6. Confirmação de pedido:

- Após revisar os itens no carrinho e as informações de pagamento, os utilizadores devem ser capazes de confirmar a encomenda.

Os requisitos não funcionais estabelecidos para este projeto são fundamentais para garantir a qualidade e a eficácia do sistema como um todo:

1. **Desempenho:** O sistema deve ser responsivo, apresentando um tempo de carregamento de página inferior a 3 segundos para proporcionar uma experiência de navegação fluida e rápida.
2. **Proteção de dados:** Todos os dados dos utilizadores, incluindo informações pessoais, devem ser protegidos com criptografia para garantir confidencialidade e integridade.
3. **Interface intuitiva:** A interface do utilizador deve ser intuitiva e de fácil navegação, possibilitando uma experiência agradável para os usuários de diferentes níveis de familiaridade com a tecnologia.
4. **Responsividade:** A plataforma deve ser responsiva em diferentes dispositivos e navegadores.
5. **Disponibilidade e confiabilidade do sistema:** O sistema deve ter uma alta disponibilidade do tempo, minimizando quaisquer possíveis interrupções.

O percurso de migração de um projeto de loja de carros para uma plataforma de venda de legos foi um processo abrangente e estruturado, visando a adaptar e otimizar a aplicação para atender a novos objetos e necessidades de negócios. Este relato detalha o passo a passo do desenvolvimento do projeto, desde a configuração inicial do ambiente local utilizando o XAMPP e a criação de base de dados no HeidiSQL, até a migração para a *Google Cloud Platform* e a implementação de pipelines automatizados para o gerenciamento contínuo do projeto.

Construção do Projeto

1. Criação do repositório no GitHub

Iniciamos o projeto criando um repositório no GitHub para armazenar o código-fonte do projeto original e as suas versões subsequentes. O repositório foi configurado para controlar as alterações ao longo do tempo, permitindo o versionamento do código.

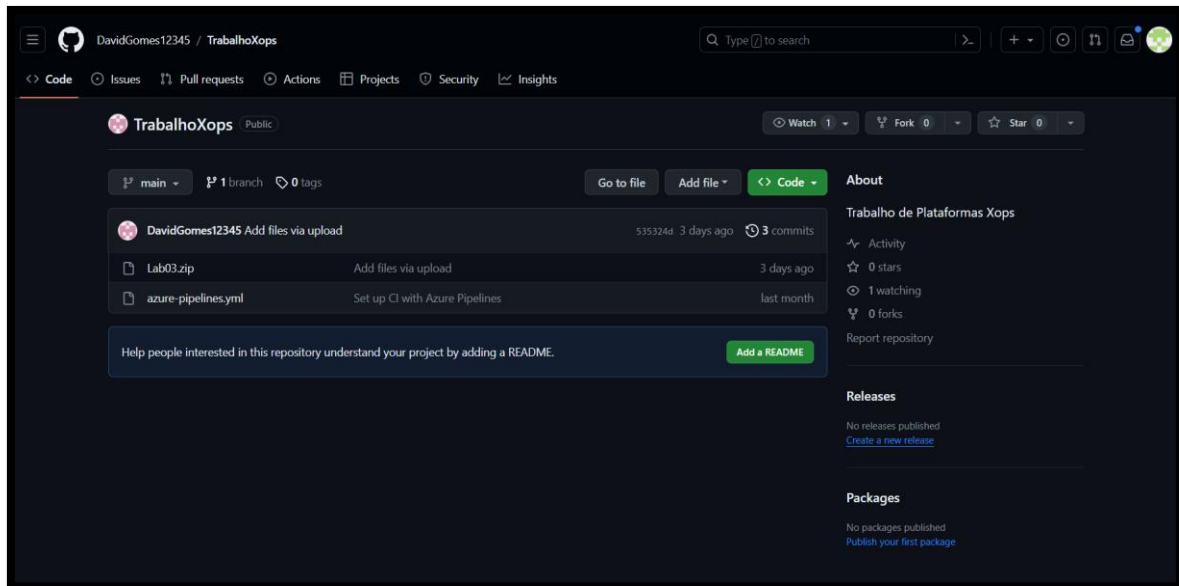


Figura 3 - Criação do Repositório no GitHub

2. Configuração do ambiente local com o XAMPP

Para viabilizar o desenvolvimento local, utilizamos o XAMPP para configurar um ambiente de servidor web, permitindo a execução do projeto localmente nos nossos computadores.

Para isso, colocamos o nosso projeto na pasta htdocs que se encontra dentro da pasta da aplicação XAMPP, para poder abrir e visualizá-lo dentro do localhost de qualquer browser.

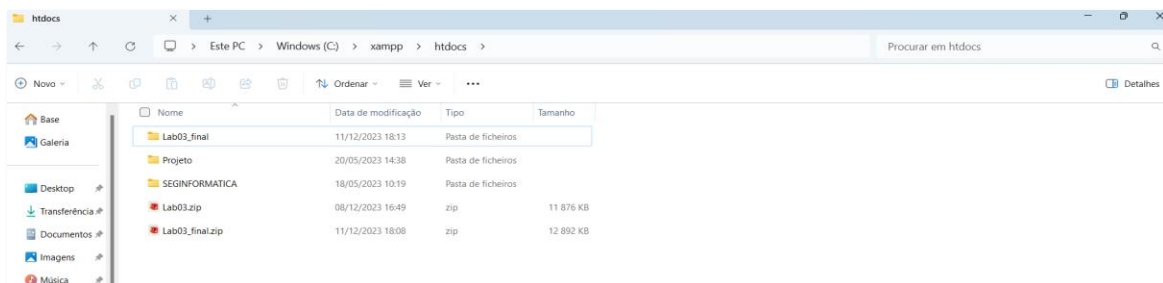


Figura 4 - Configuração de um ambiente local com o XAMPP

3. Criação da base de dados no HeidiSQL

Utilizando o HeidiSQL, criamos uma base de dados “store” para a nossa aplicação. Foram criadas tabelas essenciais como ‘admins’, ‘clientes’, ‘contactos’, ‘encomendas’, ‘encomenda_produto’ e ‘produtos’, estabelecendo a estrutura necessária para armazenar informações relevantes.

A tabela ‘admins’ serviu para guardar as informações de acesso do administrador para a aplicação.

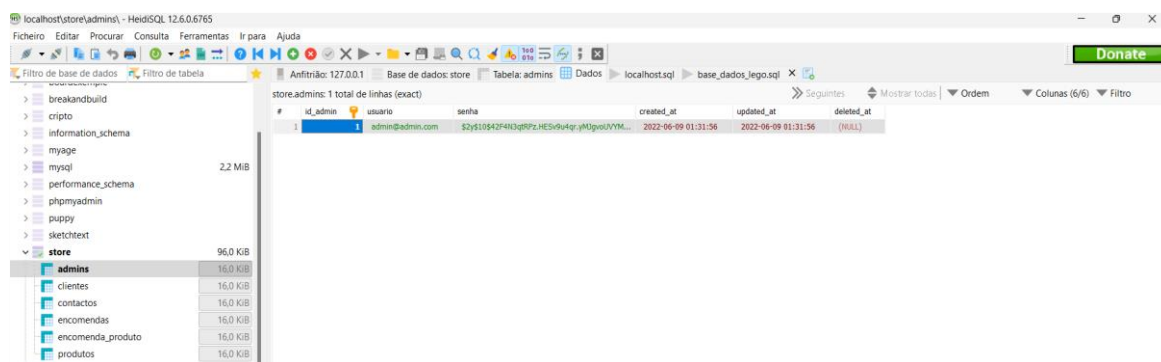


Figura 5 - Criação da base de dados no HeidiSQL

A tabela ‘clientes’ foi criada para armazenar as informações pessoais e dados de acesso à página pessoal dos clientes. Dentro desta temos o id, o email, a senha, o nome completo, a morada, a cidade e o telefone.

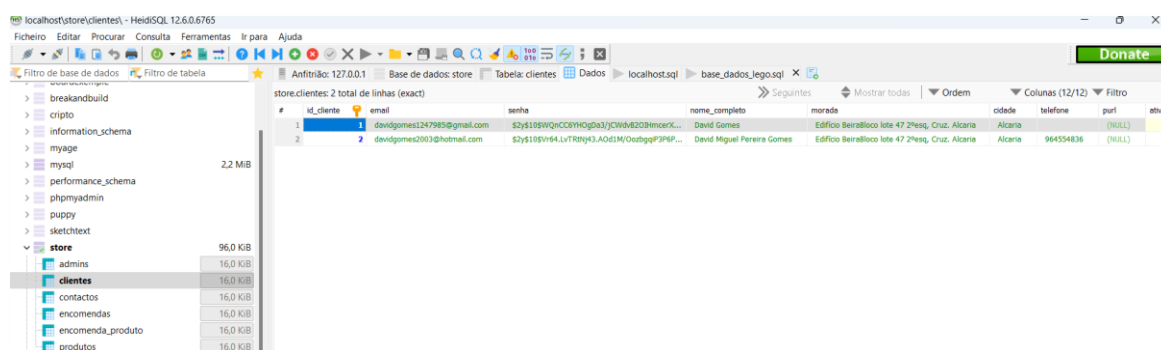
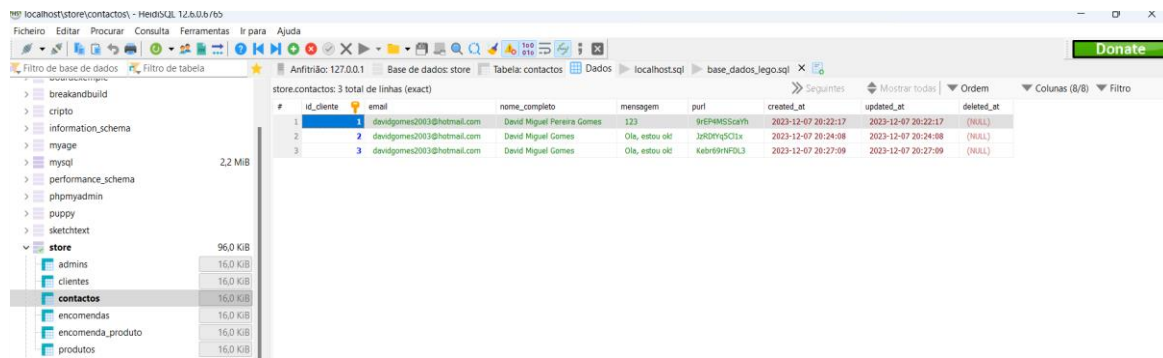


Figura 6 - Criação da tabela "clientes"

A tabela ‘contactos’ tem como propósito preservar as informações preenchidas, pelos clientes, no formulário apresentado na página “Contactos” do nosso site.

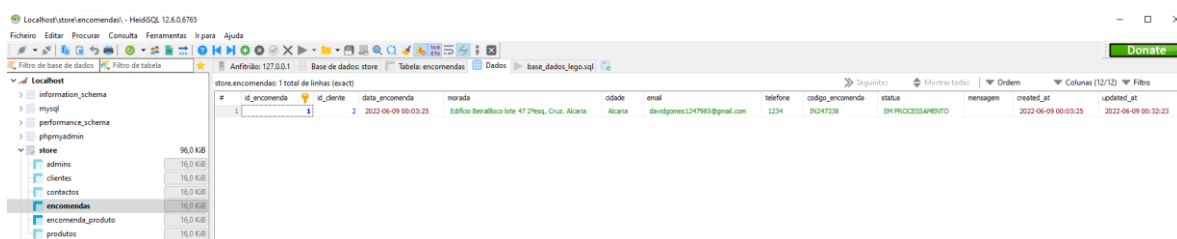


store.contactos: 3 total de linhas (exact)

#	id_cliente	email	nome_completo	mensagem	url	created_at	updated_at	deleted_at
1	1	davidgomes2003@hotmail.com	David Miguel Pereira Gomes	123	9fE4HSSca7h	2023-12-07 20:22:17	2023-12-07 20:22:17	(NULL)
2	2	davidgomes2003@hotmail.com	David Miguel Gomes	Ola, estou ok!	3dScmPc7ta	2023-12-07 20:24:08	2023-12-07 20:24:08	(NULL)
3	3	davidgomes2003@hotmail.com	David Miguel Gomes	Ola, estou ok!	Kab0t9nF0d3	2023-12-07 20:27:09	2023-12-07 20:27:09	(NULL)

Figura 7 - Criação da tabela "contactos"

A tabela ‘encomendas’ tem como utilidade registrar as encomendas que foram feitas na loja, identificando o id da encomenda, o cliente que realizou a encomenda, a data e hora que foi a encomenda foi feita e dados para onde a encomenda será direcionada.

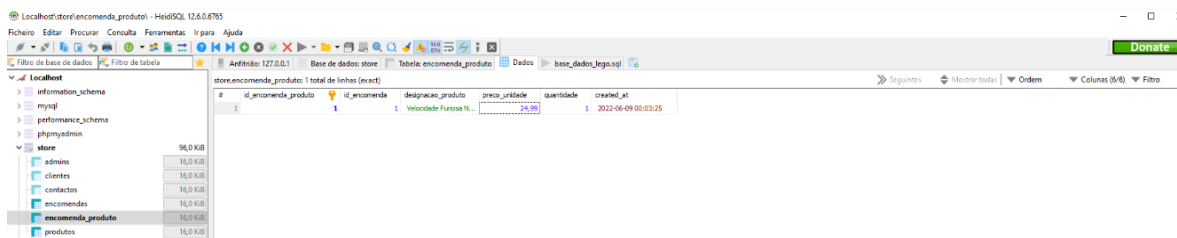


store.encomendas: 1 total de linhas (exact)

#	id_encomenda	id_cliente	data_encomenda	noticia	cidade	email	telefone	codigo_encomenda	status	mensagem	created_at	updated_at
1	1	1	2022-06-09 00:03:25	Edifício Iteraflores lote 47 2º And, Cruz. Alcantara	Alcantara	davidgomes124788@gmail.com	1234	26247338	EM PROCESSAMENTO		2022-06-09 00:03:25	2022-06-09 00:12:23

Figura 8 - Criação da tabela "encomendas"

A tabela ‘encomenda_produto’ auxilia o registo das informações do produto que foi encomendado do stock da loja.



store.encomenda_produto: 1 total de linhas (exact)

#	id_encomenda_produto	id_encomenda	designacao_produto	preco_unitario	quantidade	created_at
1	1	1	Velocidade Futura N...	24,90	1	2022-06-09 00:03:25

Figura 9 - Criação da tabela "encomenda_produto"

Por fim, a tabela 'produtos' foi criada para o guardar o stock da loja, mantendo o id do produto, o seu nome, a sua descrição, a imagem, o preço e a sua quantidade disponível na loja.

store produtos: 26 total de linhas (exact)

#	id_produto	categoria	nome_produto	descricao	imagem	preco	stock	visivel	created_at	updated_at
1	1	Harry Potter	Harry, Hermione, Ron e Hagrid	Recriações LEGO® Brickheadz™ de 4 das pers...	Fig7.png	24,99	1	1	2022-06-06 22:49:15	2022-06-06 2
2	2	Arquitetura	Taj Mahal	Construa e exprime o máximo do amor na arq...	Fig1.png	119,99	1	1	2022-06-06 22:51:18	2022-06-06 2
3	3	Arquitetura	Londres	Celebra a diversidade arquitetónica de Londr...	Fig2.png	39,99	1	1	2022-06-06 22:51:19	2022-06-06 2
4	4	Ícons	Lego Titanic	Prestie homenagem ao navio mais famoso do h...	Fig3.png	679,99	1	1	2022-06-06 22:52:07	2022-06-06 2
5	5	Ícons	Buquê de Flores	Surpreenda um amigo ou alguém de quem go...	Fig4.png	59,99	1	1	2022-06-06 22:53:19	2022-06-06 2
6	6	Ícons	Real Madrid - Estádio Santiago Bernabéu	O rugido da multidão. A alegria de um gelo. En...	Fig5.png	399,99	0	1	2022-06-06 22:54:14	2022-06-06 2
7	7	Ícons	Ghostbusters™ ECTO-1	Prepara-se e equipe-se com a sua arma de pr...	Fig6.png	239,99	1	1	2022-06-06 22:55:04	2022-06-06 2
8	8	Harry Potter	Castelo de Hogwarts	Bem-vindo ao icónico LEGO® Harry Potter™ 71...	Fig8.png	469,99	1	1	2022-06-06 22:57:25	2022-06-06 2
9	9	Harry Potter	Hogwarts Express™ - Edição para Colecionad...	Capture a magia das histórias do Harry Potter...	Fig9.png	499,99	1	1	2022-06-06 22:58:50	2022-06-06 2
10	10	Speed Champions	Velocidade Furiosa Nissan Skyline GT-R (R34)	Este modelo réplica do LEGO® Speed Champ...	Fig10.png	24,99	1	1	2022-06-06 22:59:47	2022-06-06 2
11	11	Speed Champions	Fast & Furious 1971 Dodge Charger R/T	Este modelo réplica colecionável do LEGO® S...	Fig11.png	24,99	1	1	2023-12-05 11:22:50	2023-12-05 1
12	12	Speed Champions	Lamborghini Countach	Prepara-te para experimentar o lendário Lam...	Fig12.png	24,99	0	1	2023-12-05 11:23:31	2023-12-05 1
13	13	Star Wars	Millennium Falcon	Bem-vindo ao modelo LEGO® Star Wars mai...	Fig13.png	849,99	1	1	2023-12-05 11:27:03	2023-12-05 1
14	14	Star Wars	Venator-Class Republic Attack Cruiser	Reviva as memórias de aventuras épicas da G...	Fig14.png	649,99	1	1	2023-12-05 11:27:52	2023-12-05 1
15	15	Star Wars	Capacete de Darth Vader	Ícónico colecionável Star Wars™; Suscite mem...	Fig15.png	79,99	1	1	2023-12-05 11:30:06	2023-12-05 1
16	16	Technic	Gundeste de Legartes Liebherr LR 13000	É grande. É poderoso. É o seu próximo desafio...	Fig16.png	679,99	1	1	2023-12-05 11:31:06	2023-12-05 1
17	17	Technic	3037 Ford GT	Descubra uma atenção numerandente, aos det...	Fig17.png	119,99	1	1	2023-12-05 11:31:54	2023-12-05 1

Figura 10 - Criação da tabela "produtos"

Tutorial e configuração da Google Cloud Platform

Seguimos um tutorial detalhado encontrado no Youtube para configurar o ambiente na *Google Cloud Platform (GCP)*. Passo a passo, configuramos serviços essenciais, como Cloud Storage e Kubernetes, familiarizando-nos com a interface da GCP e suas funcionalidades.

1. Em primeiro lugar, criamos um repositório na plataforma.

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to littlebricks.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
davidgomes1247985@cloudshell:~ (littlebricks)$ gcloud source repos create hello-LittleBricks
API [sourcerepo.googleapis.com] not enabled on project [littlebricks]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y

Enabling service [sourcerepo.googleapis.com] on project [littlebricks]...
Operation "operations/acat.p2-901587450092-f52f6174-2ad4-4b56-b119-6174caa59af1" finished successfully.
Created [hello-LittleBricks].
WARNING: You may be billed for this repository. See https://cloud.google.com/source-repositories/docs/pricing for details.
davidgomes1247985@cloudshell:~ (littlebricks)$
```

Figura 11 - Criação do Repositório

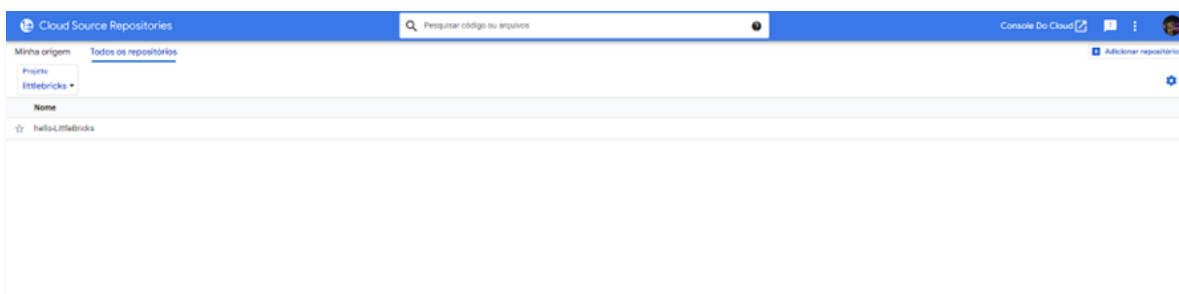


Figura 12 - Repositório

2. Como segundo passo, testamos os nossos conhecimentos na nova plataforma e produzimos uma CI/CD pipeline. Para isso, fomos utilizando certos comandos essenciais: *gcloud source repos create*; *gcloud source repos clone*; *git add .*; *git origin push master*; *git commit -m*; *gcloud app deploy*...
- O comando *gcloud source repos create* é usado para criar um novo repositório no Google Cloud Source Repositories (GCSR).
 - O comando *gcloud source repos clone* é utilizado para clonar um repositório do GCSR para o seu ambiente local.
 - O comando *git add .* é usado para adicionar todas as alterações feitas nos arquivos do repositório git no próximo commit.
 - O comando *git origin push master* é utilizado para enviar as alterações do branch local chamado “master” para o repositório remoto chamado “origin”.
 - O comando *git commit -m* é utilizado para criar um novo commit no Git e incluir uma mensagem de commit diretamente na linha de comando, sem a necessidade de abrir um editor de texto separado para inserir a mensagem.
 - O comando *gcloud app deploy* é utilizado para implantar e atualizar aplicações no GCP App Engine. O app Engine é um serviço gerenciado que permite hospedar e dimensionar automaticamente aplicações da web de forma simples na infraestrutura da Google.

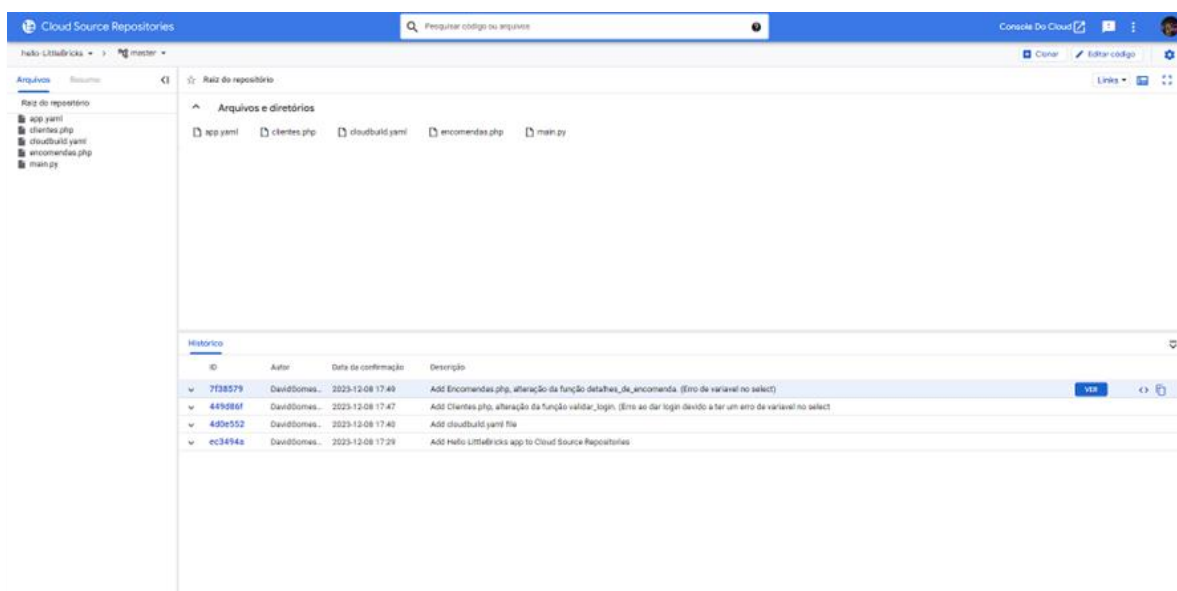


Figura 13 - Repositório com alguns ficheiros

3. De seguida, prosseguimos a explorar o ambiente da GCP e criamos um Trigger.

Este por sua vez, é um mecanismo que aciona automaticamente a execução de uma pipeline de CI/CD quando determinadas condições são atendidas num repositório de código.

Para acionar este mecanismo, antes de tudo procuramos no painel de navegação pela Cloud Build e após isso, acedemos ao marcador trigger no menu lateral.

A seguir, clicamos em “Criar Gatilho” e configuramos o ambiente, selecionando o repositório onde desejávamos acionar o build.

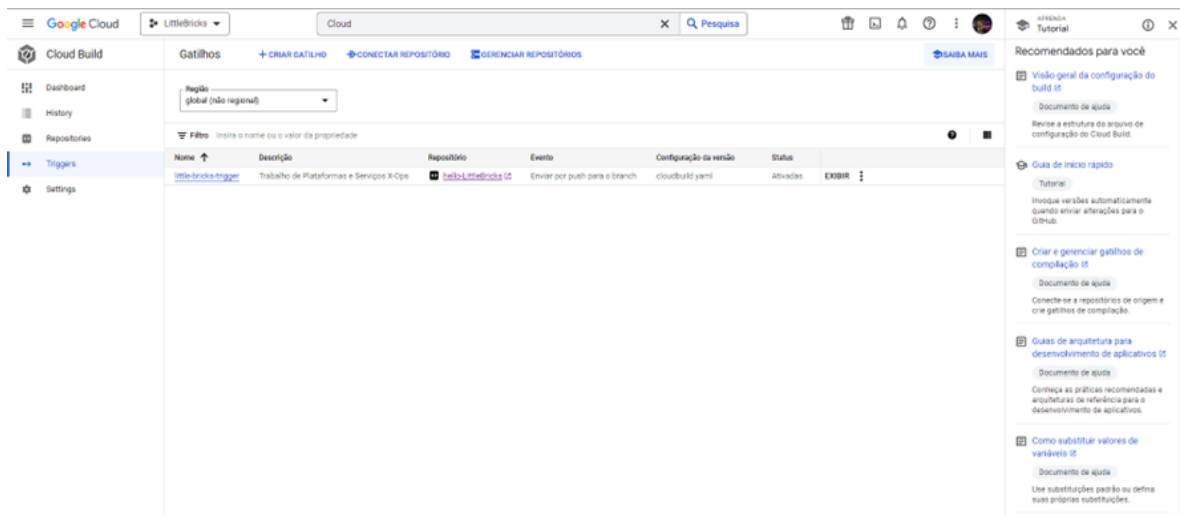


Figura 14 - Criação de um Trigger

4. Aplicamos a ativação de certos serviços

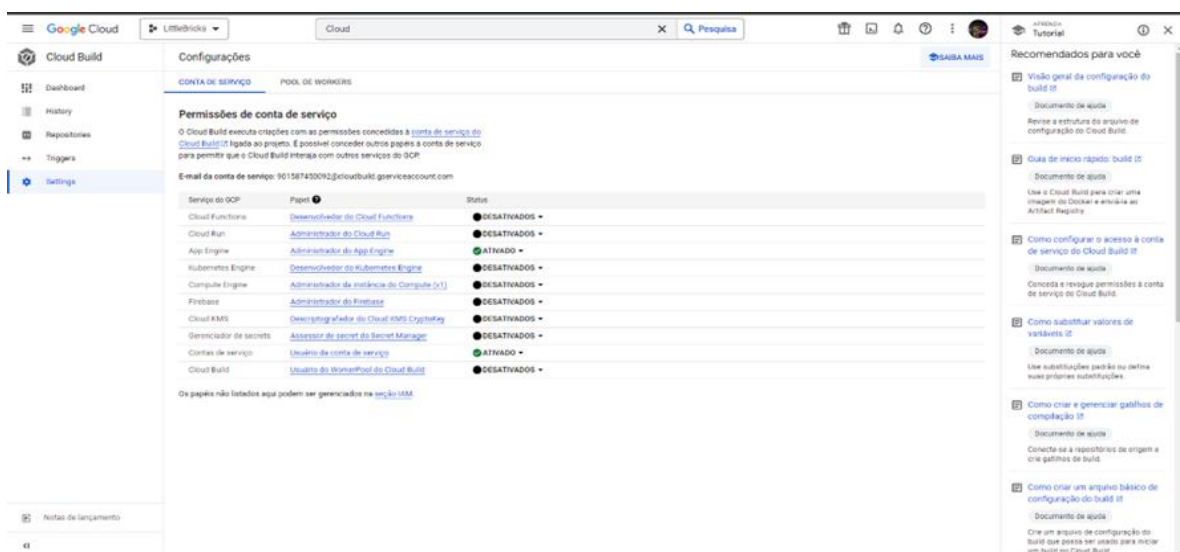


Figura 15 - Ativação dos Serviços

5. Criação do Bucket

Neste passo, procuramos o Cloud Storage e clicamos em “Create Bucket”, preenchendo algumas informações necessárias para a configuração. Este vai permitir guardar todos os ficheiros do nosso projeto necessário para a pipeline.

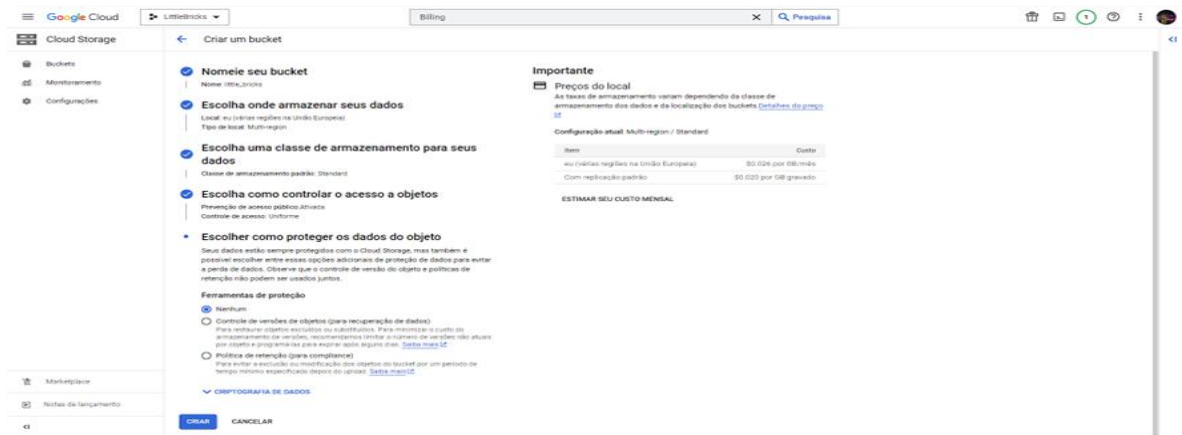


Figura 16 - Criação do Bucket

Adaptações no projeto original

Enquanto configuramos o ambiente na GCP, começamos a fazer adaptações no projeto original para alinhá-lo com a nova direção do negócio, transformando a loja de carros numa plataforma de venda de legos. Isso incluiu na interface do usuário, alterações na lógica de negócios e integração de novos recursos.

A maior alteração ocorreu na interface da página inicial e da página da loja. Foi alterado o conceito e o conteúdo da página.



Figura 17 - Nova Página da Little Bricks

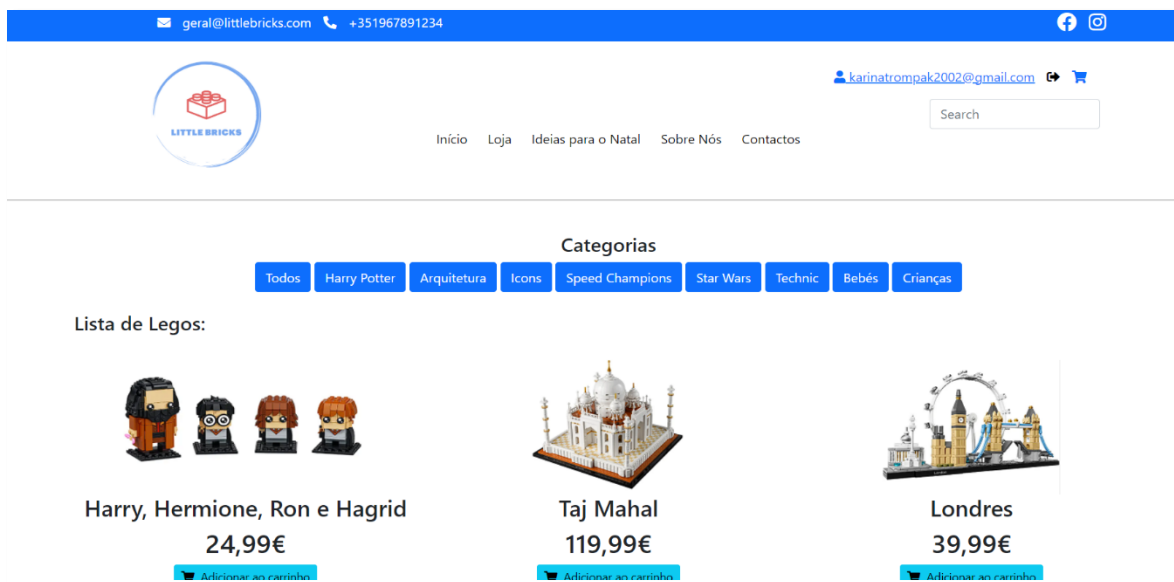


Figura 18 - Loja da Little Bricks

Adicionamos a página “Ideias para o Natal” e “Sobre Nós”. Na primeira decidimos colocar os produtos mais procurados, da época em questão, proporcionando maior interesse por parte dos clientes e contribuição para a evolução do possível negócio. Na segunda página, inserimos uma breve descrição acerca da nossa loja e dos seus produtos.

Em complemento, acrescentamos alguns separadores essenciais e correspondentes às necessidades básicas e formais de uma e-commerce, dentro destes, “Termos e Condições”, “Política de Privacidade” e “Livro de Reclamações”.



Figura 19 - Novo Rodapé da Little Bricks

4. Integração com o GitHub e configuração de pipelines

Integrando o repositório no GitHub com a GCP, configuramos pipelines de integração contínua e implantação contínua. Utilizando ferramentas como *Cloud Build* e *Cloud Deployment Manager*, criamos pipelines automatizadas para compilar e implantar as novas versões do projeto.

5. Implantação das versões atualizadas

Utilizando os pipelines configurados, realizamos implantações sucessivas das versões atualizadas do projeto na GCP. Através da automação proporcionada pelos pipelines, garantimos a consistência e estabilidade das implantações.

Resultados e Dificuldades

Durante o desenvolvimento deste projeto, alcançamos progressos significativos na transição da loja de carros para uma loja de brinquedos. Implementamos com sucesso uma variedade de melhorias, incluindo a reestruturação da interface, a adição de categorias de produtos, formulários de contacto otimizados e um processo mais fluido de seleção de produtos. Além disso, configuramos um ambiente eficaz na *Google Cloud Platform*, aproveitando recursos como Compute Engine, Kubernetes e *Cloud Storage*.

No entanto, apesar dessas realizações, é importante ressaltar dois desafios que ainda estamos a enfrentar:

1. Barra de busca: Apesar do esforço para aprimorar a funcionalidade do site, a barra de busca continua indisponível no momento.
2. Processo de pagamento: Incapacidade de efetuar o pagamento real das compras.

Durante o processo de implementação das alterações na plataforma, deparamo-nos com várias dificuldades:

1. Dificuldade na configuração adequada do sistema de login e registo de usuários.

Resolução: Descobrimos que um dos problemas ocorreu devido a um erro da nomenclatura na base de dados. Enquanto a variável na função de login foi identificada como “deleted_at”, na base de dados foi criada como “delete_at”. Esse desalinhamento causou inconsistências no processo de verificação de login.

Outro desafio enfrentado foi a condição “delete_at IS NULL”. Essa condição foi inicialmente configurada para garantir que os registos não fossem excluídos, no entanto resultou numa lógica incorreta que impedia efetivamente a realização de logins, levando à supressão automática de registos.

Encontramos um problema na variável “\$purl” que é responsável pela geração de um hash aleatório para identificar os usuários ativos e inativos. Durante a execução desta função, descobrimos que a variável “\$purl” não estava a criar corretamente o hash necessário. Isso resultou na incapacidade de distinguir os novos usuários registrados e impossibilitou o envio de e-mails de confirmação para esses novos usuários. Para resolver isso, foram realizadas correções específicas na lógica de geração do hash.


```

public function validar_login($usuario, $senha)
{
    // verificar se o login é válido
    $parametros = [
        ':usuario' => $usuario
    ];

    $bd = new Database();
    $resultados = $bd->select("
        SELECT * FROM clientes
        WHERE email = :usuario
        AND ativo = 1
        AND delete_at IS NULL
    ", $parametros);
}

```

Figura 20 - Código de Login

2. **A exibição precisa das categorias associadas a cada produto foi um desafio, resultando em dificuldades na navegação eficiente pelos diferentes tipos de produtos disponíveis.**

Resolução:

1. Foi desenvolvida uma função para buscar as categorias existentes na base de dados e armazenar essa informação em um array. Isso permitiu ter acesso rápido e organizado às categorias disponíveis.
 2. Na função principal da página da loja, a função “lista_categorias” foi invocada para obter acesso ao array contendo as categorias. Essa integração possibilitou o uso das categorias para criar os botões necessários.
 3. Foi desenvolvido um gerador automático de botões, baseado na quantidade de categorias presentes no array. Isso permitiu a criação dinâmica de botões de categoria, simplificando o processo de inclusão de novas categorias, pois os botões são gerados automaticamente conforme o conteúdo do array é atualizado.
3. **A correta exibição das informações das encomendas nos perfis dos clientes e na página de administração, bem como a visualização precisa dos dados dos clientes.**

Resolução: O problema identificado estava na criação da variável “\$lista_produtos” destinada a realizar a consulta na base de dados. Porém, no momento de retorno da função, ocorreu uma inconsistência na variável utilizada, resultando numa resposta incompleta ou ausente da lista de encomendas dos usuários, juntamente com os seus detalhes.

Após uma análise minuciosa, identificamos a raiz desse problema e procedemos com a correção adequada.

```
// =====
public function buscar_detalhes_encomenda($id_encomenda)
{
    // vai buscar os detalhes de uma encomenda
    $bd = new Database();

    $parametros = [
        ':id_encomenda' => $id_encomenda
    ];

    // buscar os dados da encomenda
    $dados_encomenda = $bd->select("
        SELECT clientes.nome_completo, encomendas.*
        FROM clientes, encomendas
        WHERE encomendas.id_encomenda = :id_encomenda
        AND clientes.id_cliente = encomendas.id_cliente
        ", $parametros);

    // lista de produtos da encomenda
    $lista_produtos = $bd->select("
        SELECT *
        FROM encomenda_produto
        WHERE id_encomenda = :id_encomenda", $parametros);

    return [
        'encomenda' => $dados_encomenda[0],
        'lista_produtos' => $lista_produtos
    ];
}
```

Figura 21 - Código para buscar informações da encomenda

4. A ativação das funções de envio de email para notificar os clientes durante a criação de contas e ao efetuarem encomendas encontrou problemas de integração ou configuração.

Resolução: O problema foi identificado numa configuração específica chamada “segurança a apps menos seguras” no serviço de email usado para enviar notificações aos clientes. Essa opção, quando desativada, estava bloqueando o envio automático de emails através do servidor.

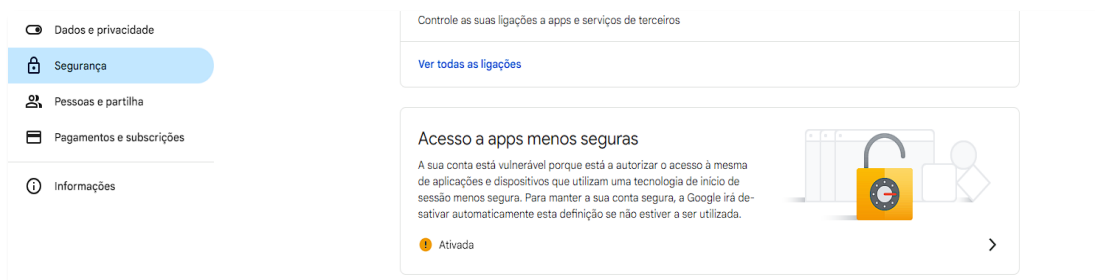


Figura 22 - Ativação da função segurança

5. A implementação do formulário de contacto e a criação adequada da tabela na base de dados para armazenar essas informações foram áreas que precisam de ajustes para funcionar conforme o esperado.

Resolução: Nessa função modificada, “\$purl” não estava a ser gerada adequadamente, mesmo com a função de geração a funcionar corretamente. Após uma análise mais detalhada, descobrimos que o problema residia na má aplicação dessa variável dentro da função. Isso estava a causar a falha na geração correta do identificador único necessário para o processo de registo.

```
// =====
public function contacto_submit()
{
    // verifica se houve submissão de um formulário
    if ($_SERVER['REQUEST_METHOD'] != 'POST') {
        $this->index();
        return;
    }

    $cliente = new Clientes();

    // inserir novo cliente na base de dados e devolver o purl
    $purl = $cliente->registar_contacto();

    // apresenta o layout para informar o envio do email
    Store::Layout([
        'layouts/html_header',
        'layouts/header',
        'criar_contacto_sucesso',
        'layouts/footer',
        'layouts/html_footer',
    ]);
    return;
}
```

Figura 23 - Código para dar submit do contacto

Discussão e Conclusão

Durante o desenvolvimento deste projeto de transformação de uma loja online, várias etapas foram realizadas para migrar de uma plataforma de vendas de carros para uma loja de legos. Ao longo do processo, diversas dificuldades foram enfrentadas, incluindo desafios técnicos e de integração de sistemas. Identificar e resolver esses obstáculos foi fundamental para o sucesso do projeto.

As mudanças implementadas foram significativas, com melhorias na interface do usuário, adição de novas funcionalidades como seleção de produtos por categorias, criação de formulários de contacto e redefinição do sistema de login. No entanto, enfrentamos desafios específicos, como erros na configuração de variáveis, problemas com integração de dados e configuração de serviços externos, como o servidor de email.

Um dos destaques desse projeto foi a transição para a *Google Cloud Platform*, onde configuramos um ambiente robusto para hospedar a aplicação, implementando práticas de CI/CD utilizando pipelines, Docker e Kubernetes. Isso proporcionou maior escalabilidade e eficiência no desenvolvimento e implantação única.

Apesar das dificuldades encontradas, o nosso grupo demonstrou resiliência e habilidades de resolução de problemas para superar os obstáculos. A colaboração e o comprometimento foram fundamentais para atingir os objetivos e entregar um produto final funcional e melhorado.

Em conclusão, o projeto apresentou avanços significativos na transformação da loja online, proporcionando uma plataforma mais moderna e adaptável para atender às necessidades dos clientes. Embora alguns desafios tenham sido encontrados, cada obstáculo superado contribuiu para um aprendizado valioso. O resultado final é um sistema mais robusto e preparado para evoluções futuras, mantendo-se alinhado com as expectativas dos usuários.

Webgrafia

https://www.youtube.com/watch?v=4h8_JyOnRCY&ab_channel=ITkFunde

[CI/CD on Google Cloud - YouTube](#)

<https://www.youtube.com/watch?v=PD83mmyAbs4>