

# RELATÓRIO FINAL TRABALHO PRÁTICO 1

---

INTEGRAÇÃO DO INSTAGRAM COM UM BLOG – TODOS OS PASSOS

**TRABALHO DESENVOLVIDO POR: DAVID RODRIGUES Nº14872**

**CONSULTE A PRÓXIMA PÁGINA PARA VISUALIZAR OS CONTEÚDOS.**

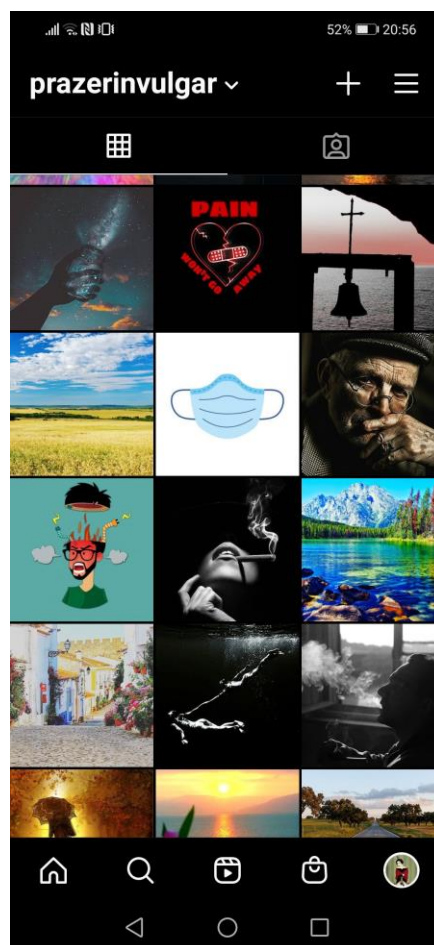
## Introdução

## MOTIVAÇÃO

O presente relatório visa todo o processo de desenvolvimento do projeto de integração, sugerido no âmbito da disciplina “Integração de sistemas de informação”, do curso “Engenharia de sistemas informáticos”, no IPCA.

IDEIA

Após muitas ideias terem surgido, a mais relevante foi tentar explorar uma API existente em larga escala (como uma rede social ou um serviço conhecido) e complementá-la com informação extra, a partir dos dados provenientes dessa API. A ideia final consiste na integração de uma página de textos no Instagram (@prazerinvulgar) num blog. Sendo que o blog tem a sua própria base de dados num servidor e as publicações do Instagram devem encaixar perfeitamente no layout do blog, parecendo que são publicações feitas dentro do mesmo, e não externas. Para a implementação desta ideia, planearam-se vários objetivos, que podem ser consultados no próximo tópico.



**Figura 1** Página @prazerinvulgar utilizada na integração.

## OBJETIVOS

Para conseguir tal feito, diversas configurações tiveram de ser feitas antes de pensar-se sequer na integração em si.

---

### FACEBOOK APP

A API pública do Instagram foi desativada e a única forma de utilizar a mesma como antes é submeter um pedido à Facebook. Uma outra maneira (mas menos ergonómica) de utilizar a API é através de *tokens* de acesso temporários para cada conta cujos dados são acedidos na página. A obtenção de um *token* e dos dados do utilizador não é fácil de automatizar, uma vez que são dados que apenas funcionam temporariamente. Não obstante ao método utilizado para aceder à API, é sempre necessário criar uma “Test App” na página de desenvolvedores do Facebook. Cada página tem direito a um Id, que servirá para comunicar com os *endpoints* da API. As imagens e os passos referentes a este tópico estão disponibilizados mais abaixo.

---

### CONFIGURAÇÃO DE UM SERVIDOR

Nenhum servidor foi comprado para efetuar este trabalho, no entanto o Facebook exige que as aplicações que acedem à sua API estejam num ambiente público, ou seja, acessíveis pela Internet. Foi configurado um servidor caseiro com ligação a um servidor DNS dinâmico e a uma interface de ligação no router. Após isso, e já com o sistema operativo CENTOS 7 instalado (pois a máquina já servia de servidor antes) vários serviços foram configurados para sustentar todo o sistema. Mais informações sobre o servidor configurado podem ser analisadas no respetivo tópico.

---

### INTEGRAÇÃO COM PDI

Foi instalado o *Pentaho Data Integration* para explorar uma ferramenta de ETL nova e para automatizar o sistema através de transformações e jobs. Mais informações no respetivo tópico.

## CONFIGURAÇÃO DA APLICAÇÃO NO FACEBOOK

Para ser possível gerar *tokens* de acesso do utilizador à API do Instagram, foi configurada uma aplicação Facebook:

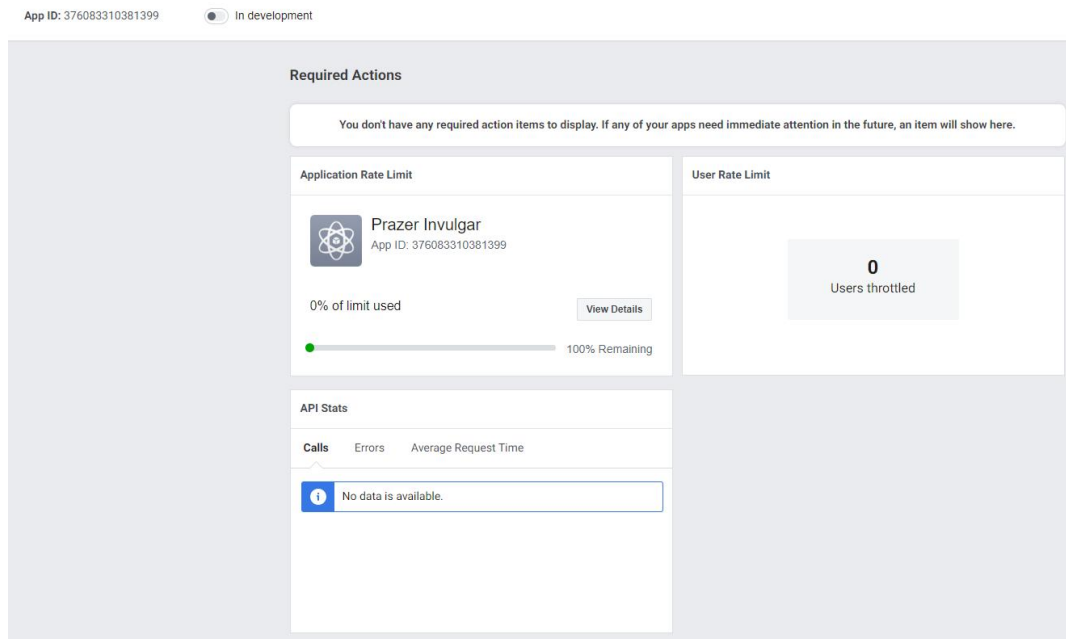


Figura 2 Dashboard da Facebook App

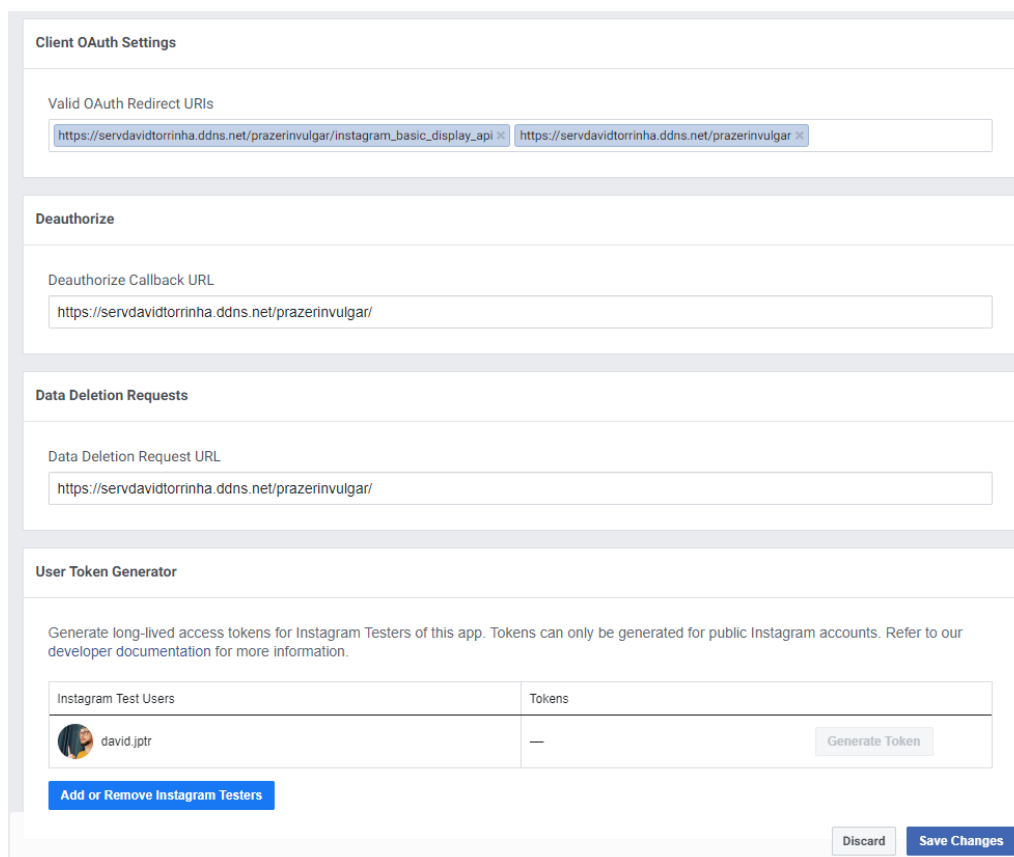


Figura 3 Dashboard Pt.2

Ao lado, no menu lateral, podem-se adicionar serviços, acrescentando-os à aplicação. O serviço utilizado para conectar ao Instagram chama-se “Instagram Basic Display API”. É o serviço que possibilita o acesso mais simplificado aos dados de uma conta (sem ser preciso confirmação por parte do Facebook), no entanto o acesso aos dados é bastante limitado e os *tokens* são sempre temporários. Apesar disso, bastou para a exemplificação de uso.

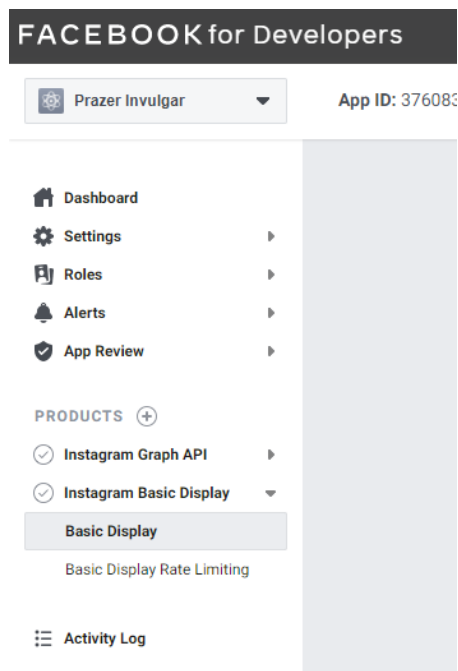


Figura 4 Menu lateral do Dashboard da Facebook App

A aplicação está em modo de teste, pois é necessário submeter um formulário para atualizar a aplicação para uma aplicação de negócio. Por isso mesmo, a única maneira de conectar a página do Instagram com a aplicação foi através das Roles. As roles fornecem ou solicitam acesso a dados, ao adicionar a página à role “Instagram Tester” é possível atribuir posteriormente um “Access Token” para aceder aos dados da mesma. Como neste contexto apenas foi preciso os dados de uma conta, apenas se adicionou a página a esta Role. No entanto, por motivos de teste, tentou-se conectar outra conta para perceber até que ponto seria possível ter duas contas complicadas. No final não acabou por ser testado.

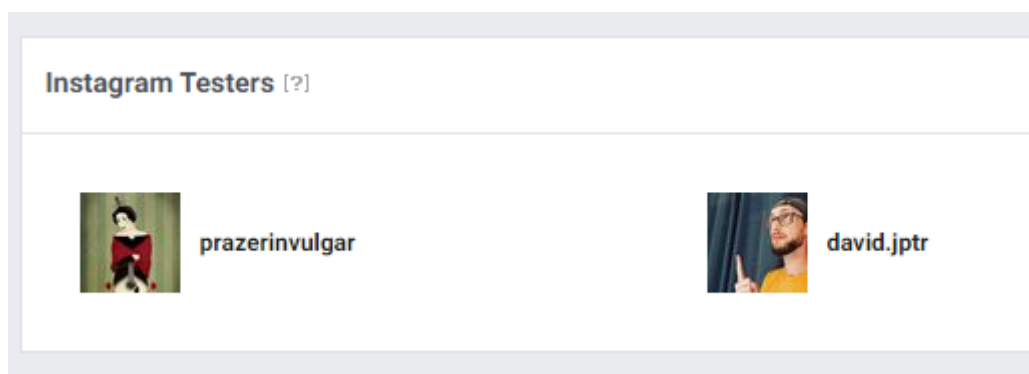


Figura 5 Atribuição da Role necessária.

Feito isto. Já foi possível obter access tokens através do website.

## CONFIGURAÇÃO DO SERVIDOR

Informações do servidor consideradas relevantes para o tema:

- Sistema operativo: CentOS 7
- Serviço HTTP utilizado: Apache (para executar um website)
- Serviço de base de dados: MySQL
- Serviço de gestão de base de dados: PhpMyAdmin
- PentaHo instalado no servidor para trabalhar diretamente com as informações da base de dados e dá página web.

## CONFIGURAÇÃO DA PÁGINA WEB

A página web simula um blog. O *template* foi simplificado de modo a ficar apenas com um layout de publicações simples. Pode consultar o link original do template (encontrado na internet) na bibliografia. A licença é gratuita e pode ser consultada nos ficheiros enviados juntamente com o presente relatório.

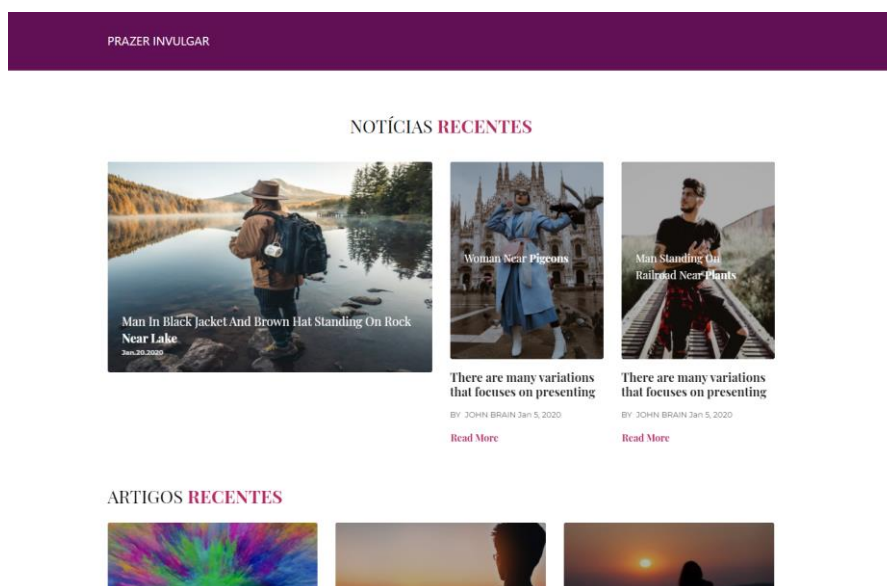


Figura 6 Demonstração do template utilizado (simplificado)

Uma segunda área foi desenvolvida na página web, responsável pela gestão dos tokens. O código foi obtido através de um tutorial, mas sofreu algumas edições por parte do autor. O repositório do tutorial pode ser consultado na bibliografia. Cada vez que o *token* expirava é solicitado um novo pedido.

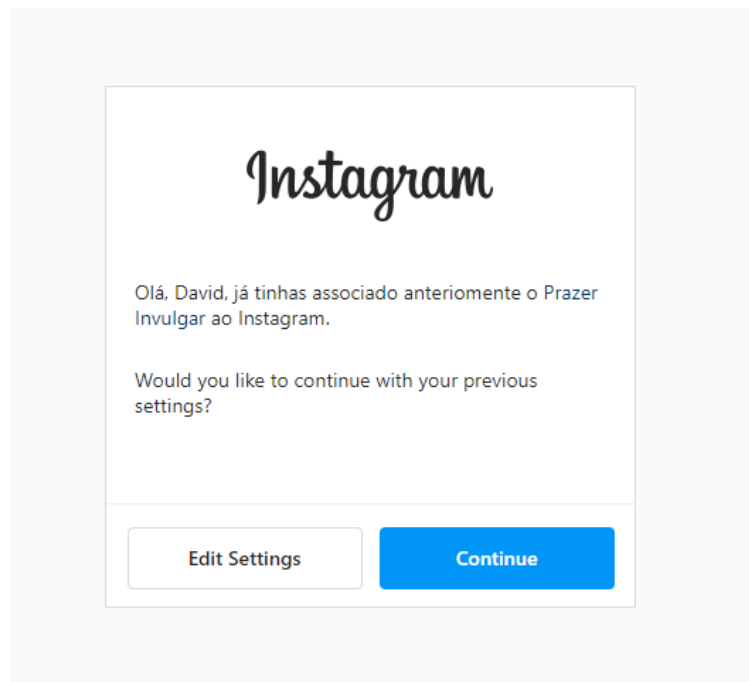


Figura 7 Janela de autenticação.

## CONFIGURAÇÃO DE UM SOFTWARE ETL

O PDI foi instalado no servidor, pois é a única maneira de o mesmo contactar o servidor de base de dados e a página web sem necessidade de nenhum serviço FTP (apesar de também ter sido configurado para trabalho remoto).

**A instalação não foi linear e foi necessário o ajuste de configurações para executar o spoon.sh no sistema operativo servidor.**

## O PROCESSO DE INTEGRAÇÃO

### DIAGRAMA FINAL

A próxima imagem apresenta o diagrama final da transformação realizada para integrar a API e a base de dados com a página web.

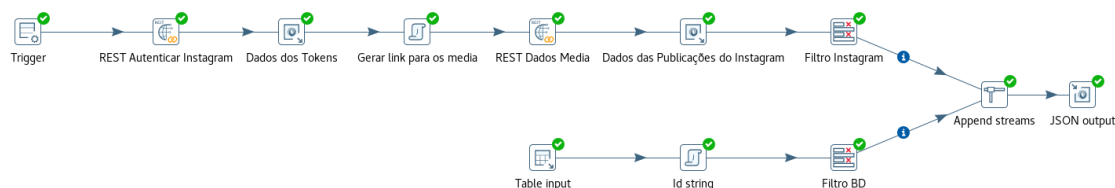


Figura 8 Diagrama da transformação

Olhando sem grandes pistas, pode-se verificar que a transformação trata das duas estruturas de dados independentemente, unindo-as posteriormente com um *Append Stream*.

### EXPLICAÇÃO DOS PASSOS

Tratamento dos dados provenientes da API do Instagram:

Trigger + Autenticar Instagram	Acude ao blog, á área de autenticação, e obtém o <i>access token</i> e o <i>user_id</i> em formato JSON
Dados dos Tokens	Tratamento dos dados obtidos na linha acima, separando os campos.
Gerar link para os media	Separados os campos, cria-se o link para aceder às publicações da página através de uma expressão de java.
Dados Media	Acude à API do Instagram com os dados solicitados, ao <i>endpoint</i> que fornece os dados das publicações da página.
Dados das Publicações do Instagram	Tratamento dos dados obtidos na linha acima, separando os campos.
Filtro Instagram	Dentro de todos os campos obtidos até agora, seleciona apenas os que serão acedidos pelo blog.

Nota: Formato de um endpoint do Instagram.

[https://graph.instagram.com/user\\_id/media?fields=id%2Ccaption%2Cmedia\\_type%2Cmedia\\_url&access\\_token=access\\_token](https://graph.instagram.com/user_id/media?fields=id%2Ccaption%2Cmedia_type%2Cmedia_url&access_token=access_token)



Tratamento dos dados provenientes da base de dados:

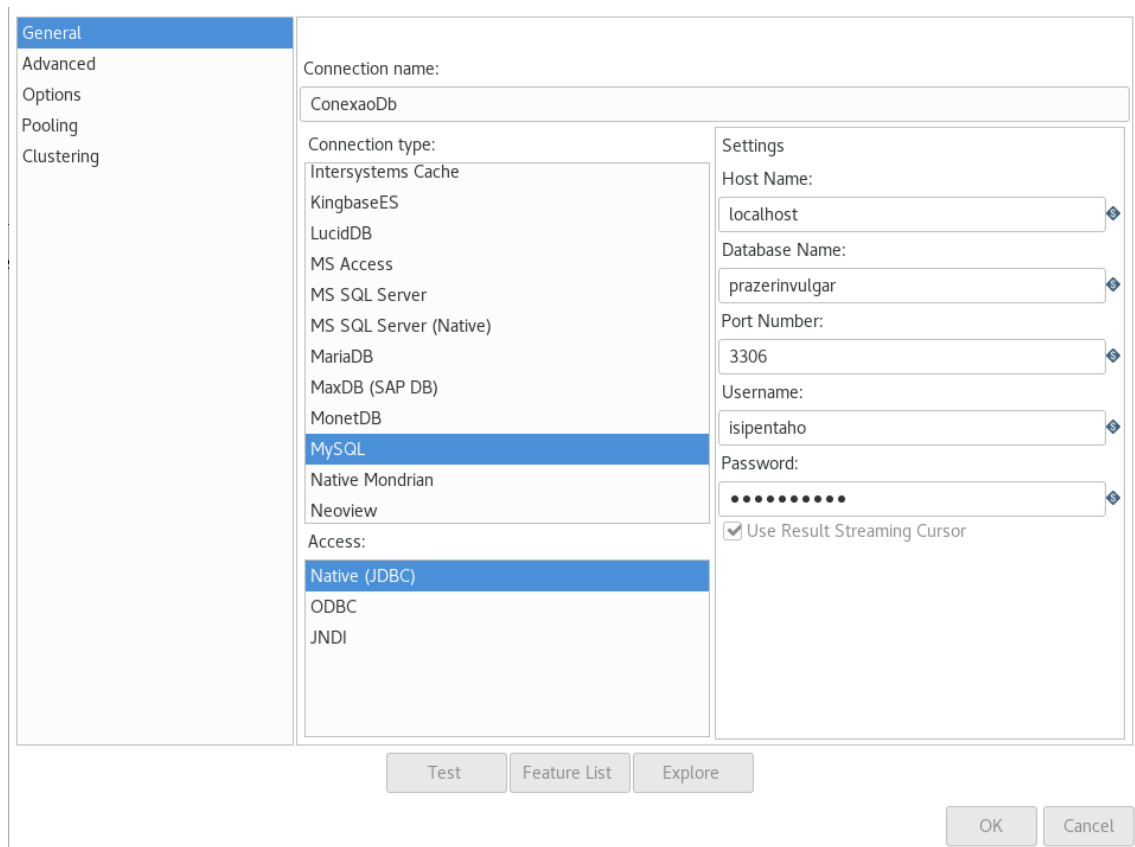


Figura 9 Ligação à base de dados.

Feita a ligação à base de dados, é possível relacionar um *Table input*.

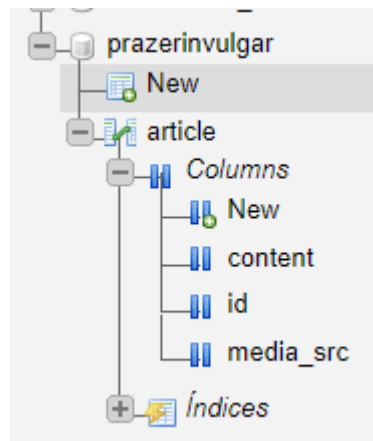


Figura 10 Estrutura da base de dados.

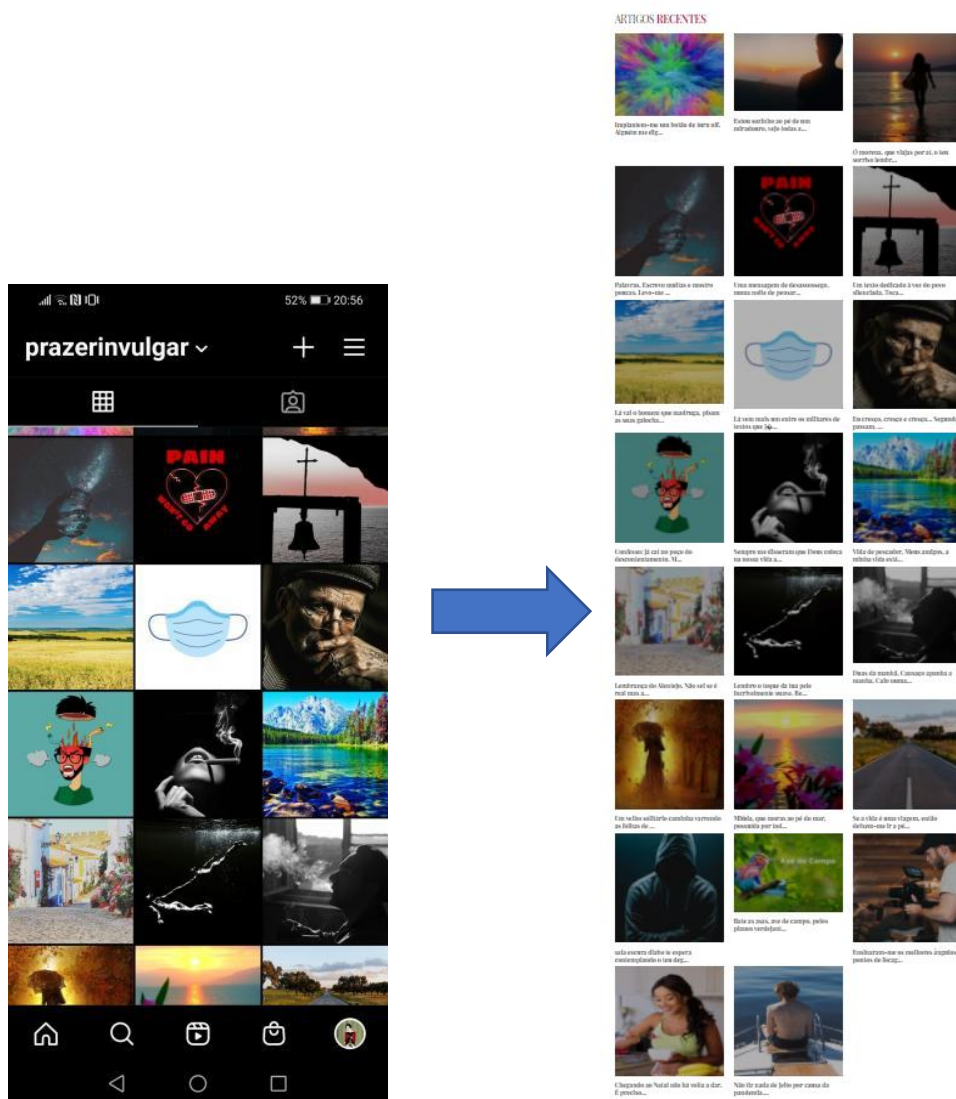
A table input contém o SQL que seleciona todos os campos da tabela. A expressão java que está ligada imediatamente a seguir converte o id para *string*, isto porque mais tarde os dados da base de dados terão de ser compatíveis com os dados da API.

O *Append Stream* exige que as colunas de ambas as streams sejam idênticas. Exatamente por isso, o filtro BD trata da nomeclatura das colunas da base de dados, colocando-as iguais às colunas do JSON da API. Os nomes das colunas da base de dados já foram propositadamente colocados de maneira diferente para testar este passo. Feito isso, o *Append Stream* junta as duas colunas de dados e no final converte tudo em Json, escrevendo num ficheiro localizado na diretoria da página web. O blog acede apenas ao Json gerado pelo PDI para exibir os dados, deixando o tratamento dos mesmos à mercê do *software*.

## CONCLUSÃO

### RESULTADO

Pudemos então aceder a ambos os tipos de publicação numa única página e através de uma única fonte:



Podemos ver que as 3 últimas imagens do lado direito são da base de dados, enquanto que as restantes são da página do Instagram.

---

## O QUE PODIA SER FEITO

Muitas outras coisas foram exploradas e podiam ser implementadas se feitas atempadamente.

- Adicionar utilizadores ao sistema e quando uma publicação do Instagram é publicada, exibir o nome e a imagem do utilizador do Instagram, em vez de tentar localizar o utilizador no sistema;
- Datas foram eliminadas por acidente, por isso as publicações não têm uma ordenação no tempo;
- Criar um Job que envia um email ao dono da página cada vez que alguém subscreve a newsletter. O job poderia abrir a janela a muitas outras funcionalidades de integração;
- Criar um webhook para o discord de gestão da página cada vez que alguém quer entrar em contacto e submete o formulário de contacto.

---

## O QUE PODIA SER MELHORADO

É possível adquirir um *Long Term Access Token* que demora muito mais a expirar do que o *Access Token*. Isso previne a constante preocupação na expiração do *Token*. No entanto, não foi utilizado por não ser necessário num ambiente de teste.

Mais conteúdo podia ter sido adicionado ao Blog. Mas o blog não foi prioritário no desenvolvimento deste trabalho.

## DIFICULDADES

Certamente, a maior dificuldade encontrou-se em compreender como funciona o PDI a um nível mais exigente. Várias tentativas de chegar à solução foram exploradas, incluindo o mapeamento de transformações secundárias, ligações dinâmicas à base de dados, etc.

A procura e estudo de informações relacionadas ao PDI fez-me entender o quão poderosa pode ser uma ferramenta de ETL, certamente será utilizada futuramente nos próximos projetos, pessoais ou não.

## REFERÊNCIAS BIBLIOGRÁFICAS

<https://developers.facebook.com/docs/instagram-basic-display-api/getting-started>

Documentação da API utilizada.

[https://github.com/jstolpe/blog\\_code/tree/master/instagram\\_basic\\_display\\_api](https://github.com/jstolpe/blog_code/tree/master/instagram_basic_display_api)

Código fonte original do meu sistema de gestão de *Tokens*.

<https://w3layouts.com/webzine-website-template/>

Template utilizado no blog.