

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	PROGRAMACIÓN WEB 1				
TÍTULO DE LA PRÁCTICA:	Laboratorio 04: Búsquedas en Google con CGI Perl				
NÚMERO DE PRÁCTICA:	04	AÑO LECTIVO:	2024 - B	NRO. SEMESTRE:	II
FECHA DE PRESENTACIÓN	13/10/2024	HORA DE PRESENTACIÓN	09:59:00 AM		
INTEGRANTE (s): Riveros Vilca Alberth Edwar				NOTA:	
DOCENTE: M.Sc. Richart Escobedo Quispe					

SOLUCIÓN Y RESULTADOS
<p>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</p> <p>Cree páginas web usando HTML y CSS que permitan realizar los siguientes tipos de búsquedas en google:</p> <ul style="list-style-type: none"> • Búsquedas simples. • Búsquedas de imágenes. • Búsquedas con opciones avanzadas. <p>Condiciones</p> <ul style="list-style-type: none"> • Deberá entregar 4 archivos: • 3 para las páginas HTML y • 1 para el CSS. • Cada página HTML deberá tener un menú en la parte superior con enlaces a las otras páginas de su tarea (todas de manera recíproca). • Para las búsquedas simples deberá crear un botón que diga "<u>Búsqueda en google</u>" y permita mostrar la página de búsqueda de google al presionarlo. De manera similar a la página de google el cuadro de texto deberá tener las esquinas redondeadas y el botón deberá estar debajo del cuadro de texto. Ambos elementos

deberán estar centrados.

- Para las **búsquedas de imágenes**, la página será similar a la de búsqueda simple, pero al apretar el botón de búsqueda deberá abrir la página de búsqueda de imágenes de google: <https://images.google.com/> y mostrar las imágenes asociadas al texto ingresado.
- Para la **búsqueda avanzada**, su página deberá hacer una búsqueda similar a https://www.google.com/advanced_search pero sólo con tres campos: "*todas estas palabras*", "*esta palabra o frase exacta*", "*ninguna de estas palabras*". De manera similar a la página de búsqueda avanzada de google, los campos de texto deberán estar alineados a la izquierda y uno debajo del otro; además el botón deberá estar alineado a la derecha con letras blancas y fondo azul.
- Usted debe escribir su propio CSS, pero en la medida de lo posible, sus formatos deben coincidir con los originales de google.

Tips

- Para saber los nombres que deberían tener sus variables (name) en sus formularios, revise los URLs que se crean al hacer las búsquedas en los respectivos formularios de google.

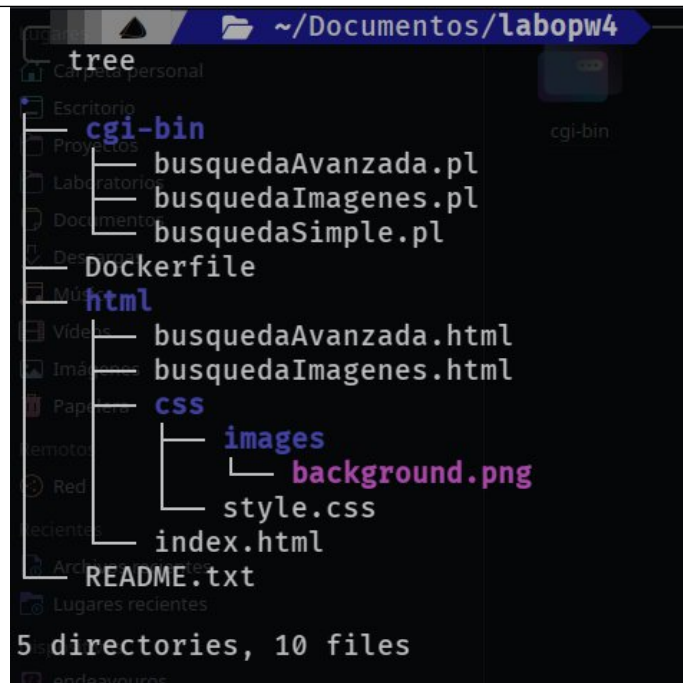
variable1=valor1&variable2=valor2&variable3=valor3...

- También podría ser útil revisar el inspector "**Network**" (En google chrome: Ver->Opciones para desarrolladores->Herramientas del programador)

Esta tarea está basada en el curso CS50's Web Programming with Python and JavaScript de Harvard

II. SOLUCIÓN DEL CUESTIONARIO

Estructura del proyecto:



busquedaSimple.pl:

```
1  #!/usr/bin/perl
2  use strict;
3  use warnings;
4  use CGI;
5
6  my $q = CGI->new;
7  my $simpleSearch = $q->param('simpleSearch')||'';
8
9  my $url = "https://www.google.com/search?q=";
10 $url .= $q->escape($simpleSearch) if $simpleSearch;
11 print $q->redirect($url);
```

busquedaImagenes.pl:

```
1 use strict;  
2 use warnings;  
3 use CGI;  
4  
5 my $q = CGI->new;  
6 my $imageSearch = $q->param('imageSearch') || '';  
7  
8 # Construir la URL de búsqueda de Google Imágenes  
9 my $url = "https://www.google.com/search?tbm=isch&q=";  
10 $url .= $q->escape($imageSearch) if $imageSearch;  
11  
12 print $q->redirect($url);
```

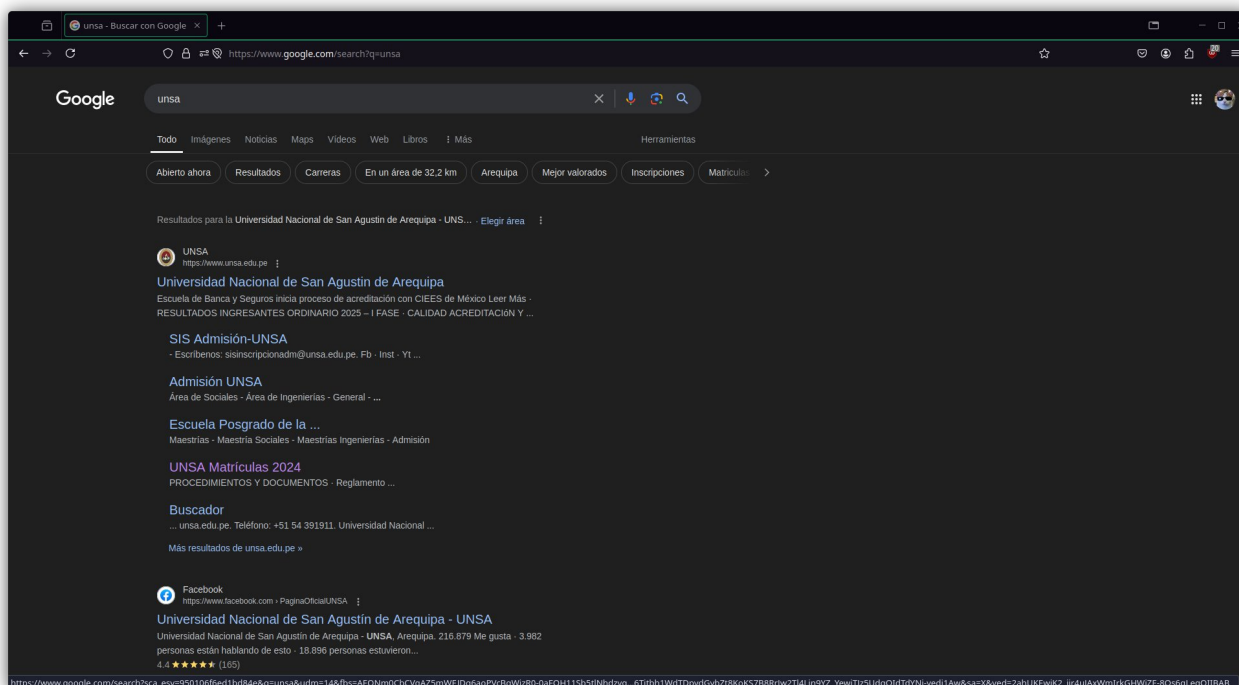
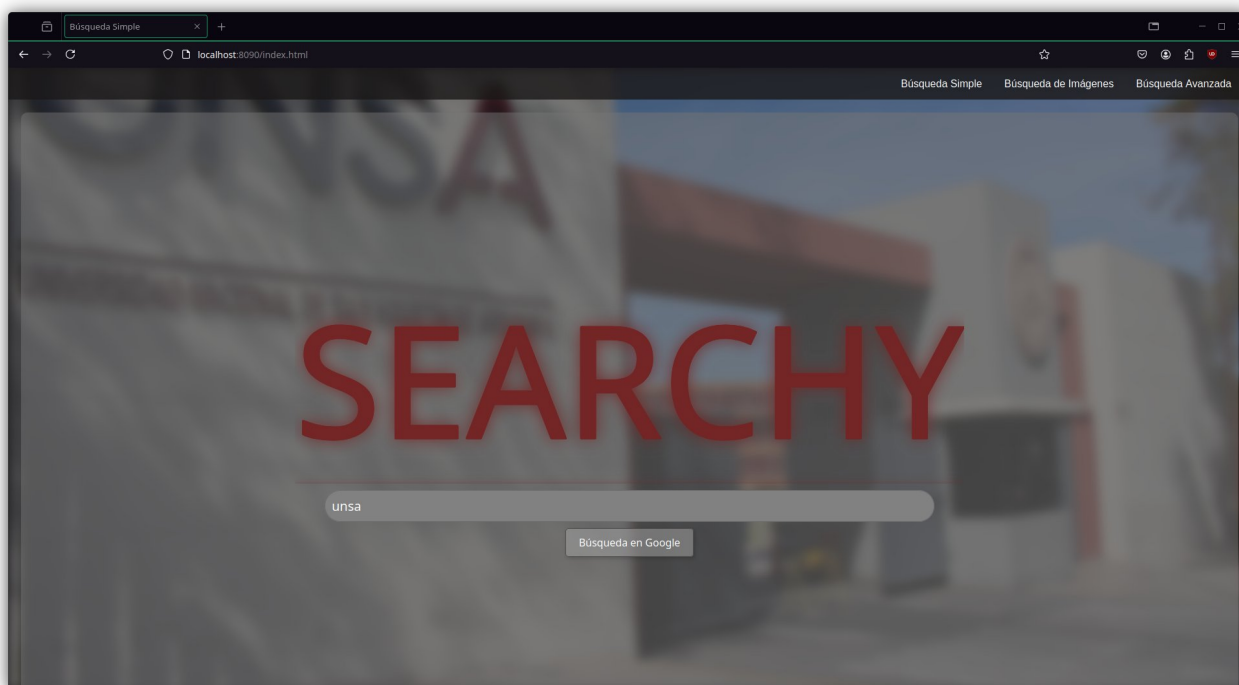
busquedaAvanzada.pl:

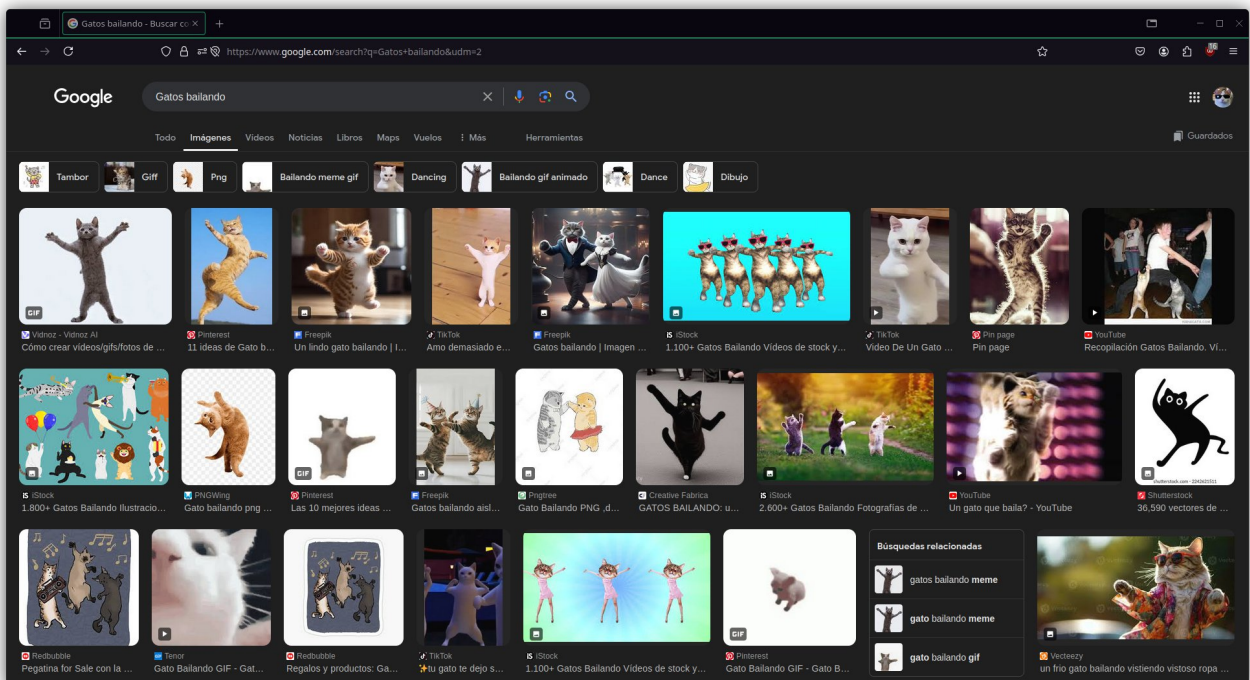
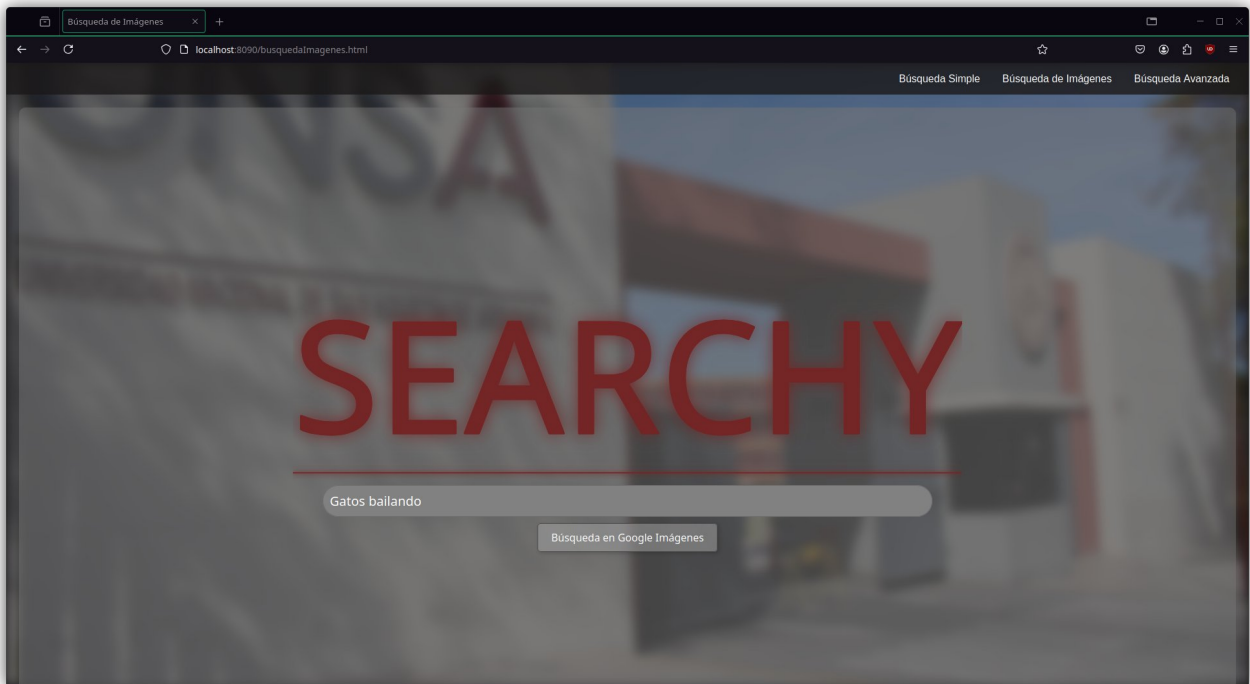
```
1 #!/usr/bin/perl  
2 use strict;  
3 use warnings;  
4 use CGI;  
5  
6 my $q = CGI->new;  
7 my $todasEstasPalabras = $q->param('allwords') || '';  
8 my $estaPalabraFrase = $q->param('thisSentence') || '';  
9 my $ningunaDeEstasPalabras = $q->param('noWords') || '';  
10  
11 my $url = "https://www.google.com/search?q=";  
12 $url .= $q->escape($todasEstasPalabras) if $todasEstasPalabras;  
13 $url .= "+\"" . $q->escape($estaPalabraFrase) . "\"\" if $estaPalabraFrase;  
14 $url .= "+-\" . $q->escape($ningunaDeEstasPalabras) if $ningunaDeEstasPalabras;  
15  
16 print $q->redirect($url);
```

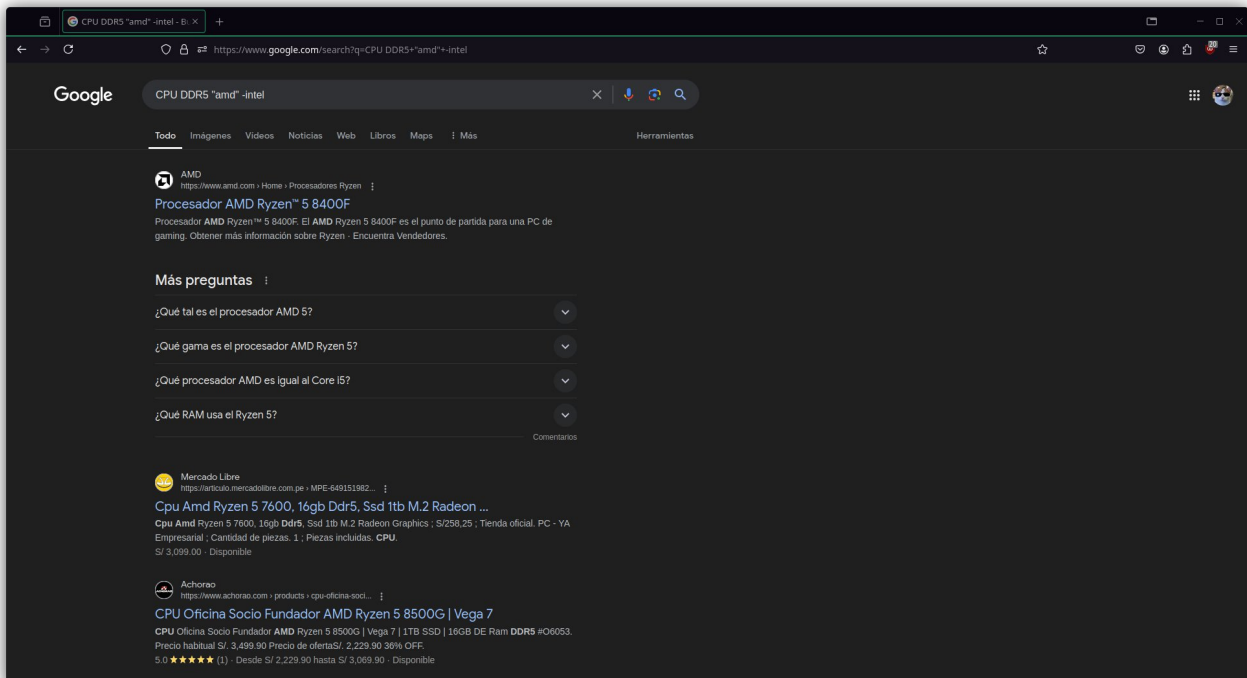
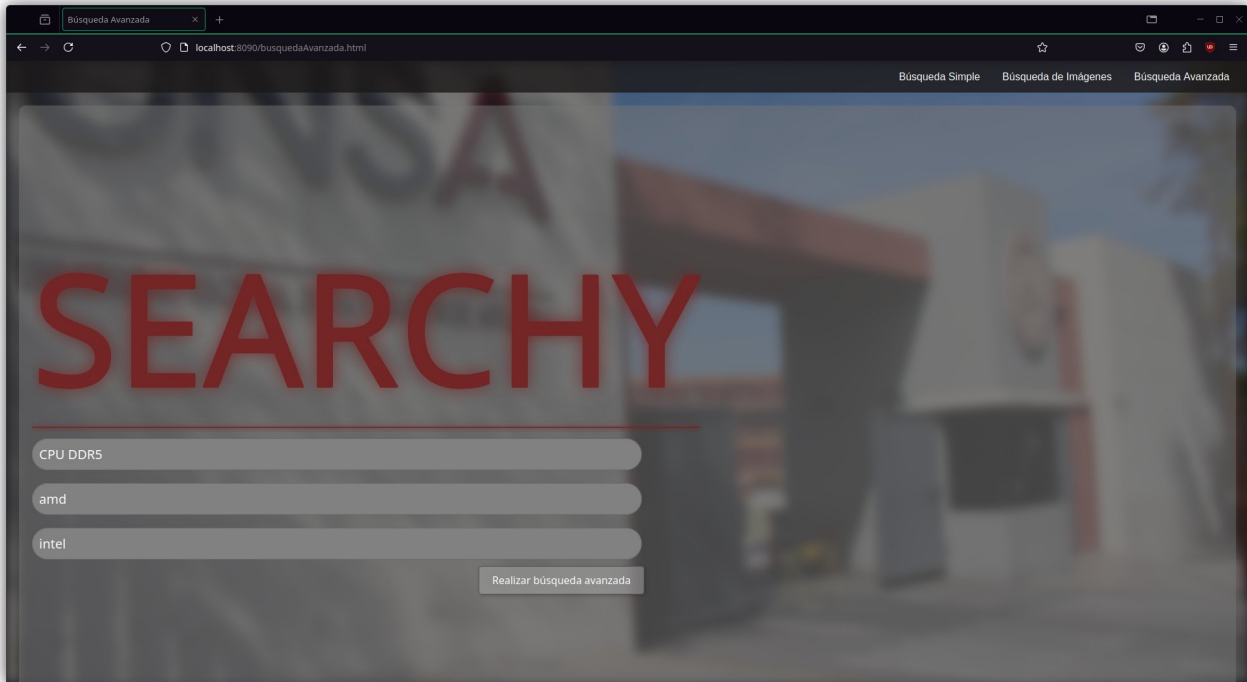
Dockerfile:

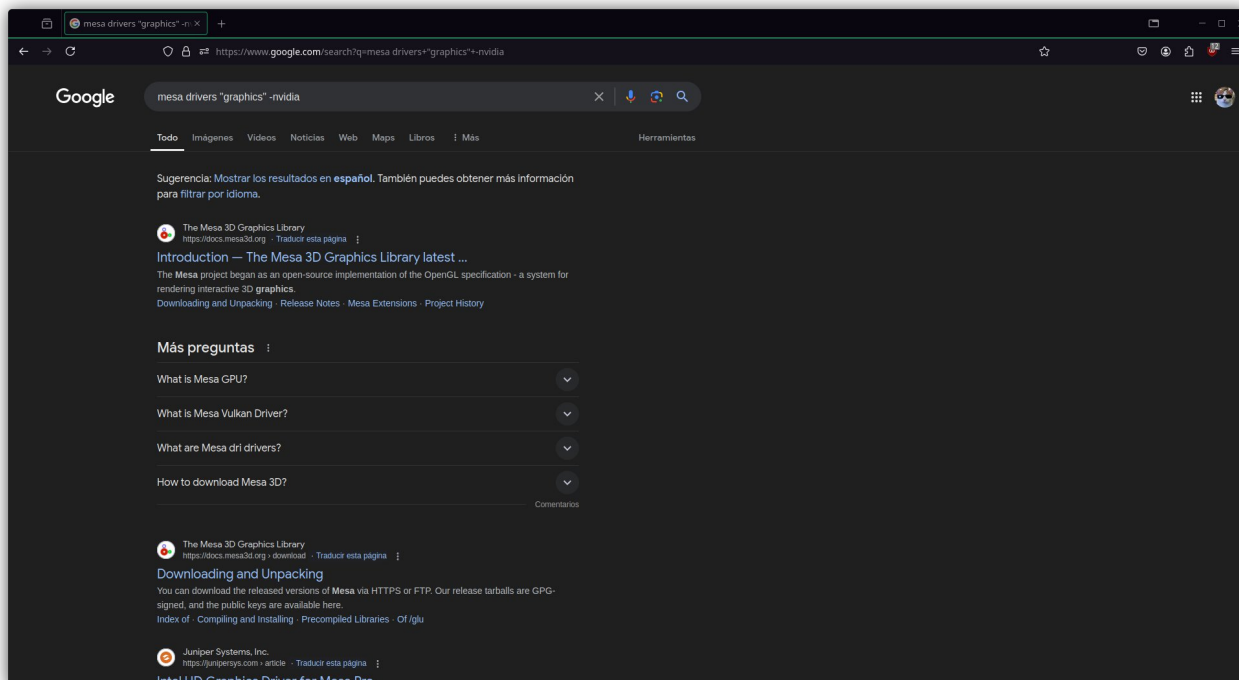
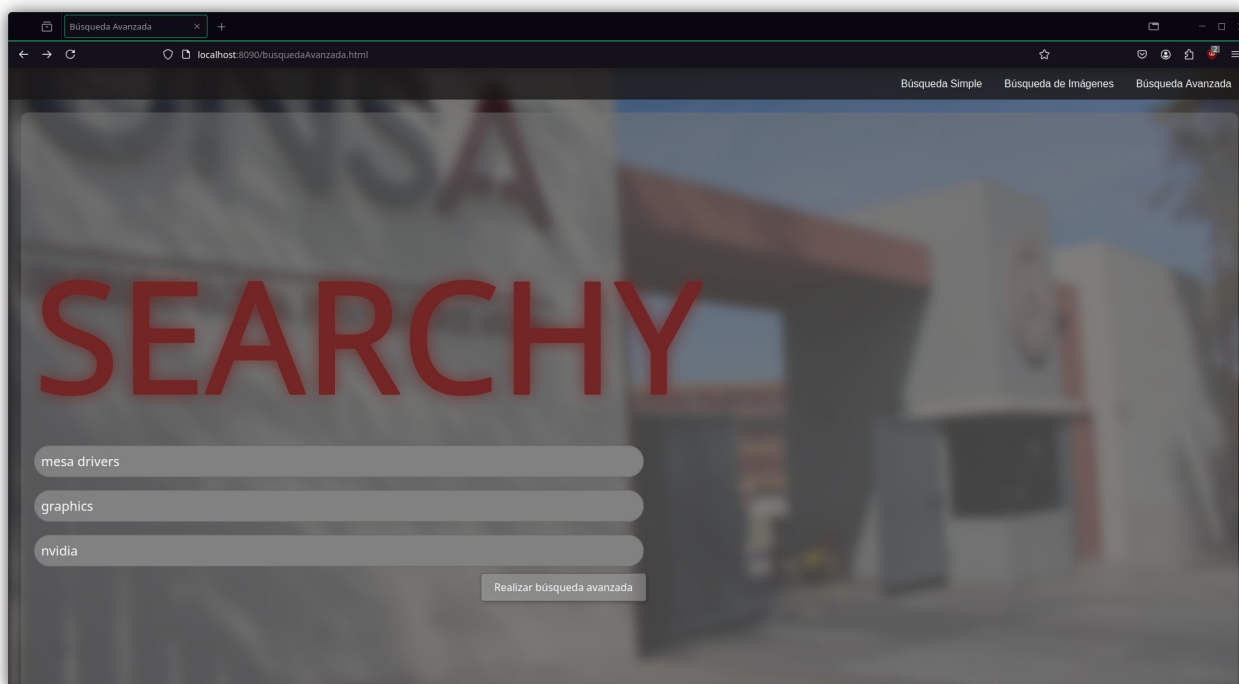
```
1 FROM bitnami/minideb
2
3 ENV DEBIAN_FRONTEND="noninteractive"
4
5 # Actualizar e instalar dependencias
6 RUN apt-get update && \
7     apt-get install -y apache2 perl vim bash locales tree libcgi-pm-perl dos2unix && \
8     apt-get clean && \
9     rm -rf /var/lib/apt/lists/*
10
11 # Configurar locales
12 RUN echo -e 'LANG=es_PE.UTF-8\nLC_ALL=es_PE.UTF-8' > /etc/default/locale && \
13     sed -i 's/^# *(es_PE.UTF-8)/\1/' /etc/locale.gen && \
14     /sbin/locale-gen es_PE.UTF-8
15
16 # Crear directorios y ajustar permisos
17 RUN mkdir -p /var/www/html /usr/lib/cgi-bin && \
18     chown www-data:www-data /usr/lib/cgi-bin /var/www/html && \
19     chmod 750 /usr/lib/cgi-bin /var/www/html
20
21 # Configurar bash
22 RUN echo "export LC_ALL=es_PE.UTF-8" >> /root/.bashrc && \
23     echo "export LANG=es_PE.UTF-8" >> /root/.bashrc && \
24     echo "export LANGUAGE=es_PE.UTF-8" >> /root/.bashrc
25
26 # Copiar y convertir scripts y archivos HTML
27 COPY ./cgi-bin/ /usr/lib/cgi-bin/
28 RUN for file in /usr/lib/cgi-bin/*.pl; do \
29     dos2unix "$file" && \
30     chmod +x "$file"; \
31 done
32 COPY ./html/ /var/www/html/
33
34 # Configurar Apache para CGI
35 RUN echo '<VirtualHost *:80>\n\
36     ServerAdmin webmaster@localhost\n\
37     DocumentRoot /var/www/html\n\
38     ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/\n\
39     <Directory "/usr/lib/cgi-bin">\n\
40         AllowOverride None\n\
41         Options +ExecCGI\n\
42         Require all granted\n\
43     </Directory>\n\
44     <Directory "/var/www/html">\n\
45         AllowOverride None\n\
46         Require all granted\n\
47     </Directory>\n\
48     ErrorLog ${APACHE_LOG_DIR}/error.log\n\
49     CustomLog ${APACHE_LOG_DIR}/access.log combined\n\
50 </VirtualHost>' > /etc/apache2/sites-available/000-default.conf
51
52 # Habilitar el módulo CGI
53 RUN a2enmod cgi
54
55 # Exponer puerto 90
56 EXPOSE 90
57
58 # Comando para iniciar Apache
59 CMD ["apachectl", "-D", "FOREGROUND"]
```

Uso:









Repositorio: <https://github.com/rivX241/PW>

III. CONCLUSIONES

El uso de CGI con Perl tiene historia con el desarrollo web, nos permite ejecutar scripts en el servidor en este caso de Apache2 gracias a esto se puede manejar los datos del formulario y automatizarlo para

que nos redireccione al buscador de Google respectivo con esas entradas, siendo el lenguaje de Perl muy fácil de emplear y sin muchas complicaciones para usarlo con Apache, además comprendí porque en sistemas operativos Windows no corría correctamente debido a CRLF y LF en caso de SO de tipo Unix lo que solucione con el paquete dos2unix.

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA

<https://www.perl.com/article/perl-and-cgi/#why-not-to-use-cgi>

<https://github.com/macaw-cad/react-thingy/issues/6>

<https://github.com/rescobedoulasalle/docker/tree/main/ejercicio5>