

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN**  
**FACULTAD DE PRODUCCIÓN Y SERVICIOS**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**CURSO:**

Programación Web 1

**APELLIDOS Y NOMBRES:**

Riveros Vilca Alberth Edwar

**ASIGNACIÓN:**

Ejercicios en Perl

### Restricciones:

- Deberá intentar resolver los problemas usando sólo el tutorial indicado, no deberá consultar material adicional en internet u otra fuente.
- Se calificará el proceso de solución y no la solución en sí. Si sabes usar git y github, puedes usarlo (por esta tarea no es obligatorio), GIT y todos los commits que sean necesarios, para demostrar el proceso que usted siguió hasta llegar a la solución. Si no llega a una buena solución, aún puedo obtener una muy buena nota si su proceso (commits) demuestran esfuerzo.
- Suba por este medio los archivos de su último intento (Los .pl, un archivo por cada problema), estos archivos servirán para constatar cuál fue la versión final de sus intentos.
- Si desea presente un informe en pdf con la ejecución de los programas, los códigos y pantallazos de sus respuestas.

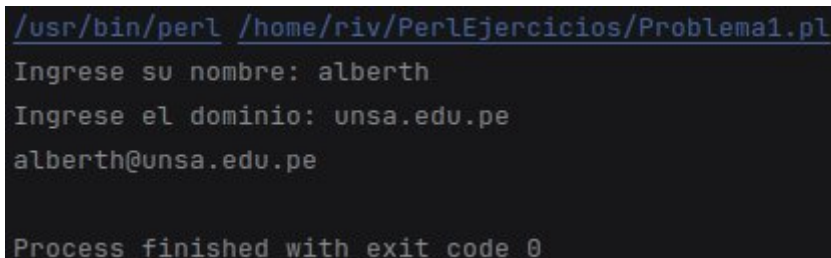
### Ejercicio 01:

Escriba un programa perl que pida el nombre de usuario y luego el dominio, luego el programa deberá imprimir el correo electrónico juntando el nombre de usuario y el dominio y usando el caracter @ en medio. Como restricción no podrá usar el operador punto(.) para concatenar strings. Si el usuario ingresa como nombre "alumno" y como dominio "pweb1", deberá imprimir: "alumno@pweb1".

A screenshot of a terminal window with a dark background and light-colored text. The terminal shows a Perl script being executed. The script starts with a shebang line, followed by strict and warnings modules. It then prompts the user for their name and domain, reads the input, and prints the concatenated email address. The output shows the user entered 'alberth' and 'unsa.edu.pe', resulting in 'alberth@unsa.edu.pe'.

```
#!/usr/bin/perl
use strict;
use warnings FATAL => 'all';
print "Ingrese su nombre: ";
my $nombre = <STDIN>;
chomp($nombre);
print "Ingrese el dominio: ";
my $dominio = <STDIN>;
print "$nombre@$dominio";
```

### EJECUCIÓN:

A screenshot of a terminal window showing the execution of the Perl script. The command to run the script is shown at the top. Below it, the prompts and user input are shown, followed by the output of the script. The output is the email address 'alberth@unsa.edu.pe'. At the bottom, it says 'Process finished with exit code 0'.

```
/usr/bin/perl /home/riv/PerlEjercicios/Problema1.pl
Ingrese su nombre: alberth
Ingrese el dominio: unsa.edu.pe
alberth@unsa.edu.pe

Process finished with exit code 0
```

2.Estamos al final del semestre y para calcular su promedio el profesor del curso desea eliminar la peor nota y duplicar la mayor nota. De este modo, si sus notas fueran 12, 15, 17 y 14; el profesor eliminaría el 12 y duplicaría el 17, entonces sus notas serían 17, 15, 17 y 14 y sobre ellas calcularía el promedio. Usted deberá programar una función que reciba las notas y devuelva el promedio, según se explicó y para una cantidad de notas no determinada. Para programar su función no podrá usar ningún tipo de condicionales, sólo las funciones max y min de perl.

```
#!/usr/bin/perl
use strict;
use warnings FATAL => 'all';
use List::Util qw(max min);

sub calcularPromedio {
    my @notas = @_;

    print "Notas originales: @notas\n";

    my $nota_minima = min(@notas);
    my @notas_sin_minima;

    my $encontrada = 0;
    foreach my $nota (@notas) {
        if ($nota == $nota_minima && !$encontrada) {
            $encontrada = 1;
        } else {
            push @notas_sin_minima, $nota;
        }
    }

    my $nota_maxima = max(@notas_sin_minima);
    push @notas_sin_minima, $nota_maxima;

    print "Notas modificadas: @notas_sin_minima\n";

    my $suma = 0;
    $suma += $_ for @notas_sin_minima;
    my $promedio = $suma / @notas_sin_minima;

    return $promedio;
}

print "Ingrese el número de notas a ingresar: ";
my $numNotas = <STDIN>;
chomp($numNotas);

my @notas = ();
for (my $i = 1; $i <= $numNotas; $i++) {
    print "Ingrese la nota N°$i: ";
    my $notaIngresada = <STDIN>;
    chomp($notaIngresada);
    push(@notas, $notaIngresada);
}

my $promedio = calcularPromedio(@notas);
print "El promedio calculado es: $promedio\n";
```

## EJECUCIÓN:

```
/usr/bin/perl /home/riv/PerlEjercicios/Problema2.pl
Ingrese el número de notas a ingresar: 5
Ingrese la nota N°1: 13
Ingrese la nota N°2: 15
Ingrese la nota N°3: 18
Ingrese la nota N°4: 16
Ingrese la nota N°5: 07
Notas originales: 13 15 18 16 07
Notas modificadas: 13 15 18 16 18
El promedio calculado es: 16

Process finished with exit code 0
```

## USO DE GITHUB Y GIT:

```
git init
hint: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
hint: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
hint: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
hint: git config --global init.defaultBranch <nombre>
hint:
hint: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
hint: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
hint: git branch -m <nombre>
Inicializado repositorio Git vacío en /home/riv/Escritorio/CGI-Perl/.git/

echo "# CGI-Perl" >> README.md
git add README.md
git commit -m "pweb-teoria"
[master (commit-raiz) 9ba6112] pweb-teoria
1 file changed, 1 insertion(+)
create mode 100644 README.md

git add .
git commit -m "pweb-teoria"
[master de43d7f] pweb-teoria
18 files changed, 512 insertions(+)
create mode 100644 CGI-Google/Dockerfile
create mode 100644 CGI-Google/README.txt
create mode 100644 CGI-Google/cgi-bin/busquedaAvanzada.pl
create mode 100644 CGI-Google/cgi-bin/busquedaImagenes.pl
create mode 100644 CGI-Google/cgi-bin/busquedaSimple.pl
create mode 100644 CGI-Google/html/busquedaAvanzada.html
create mode 100644 CGI-Google/html/busquedaImagenes.html
create mode 100644 CGI-Google/html/css/images/background.png
create mode 100644 CGI-Google/html/css/style.css
create mode 100644 CGI-Google/html/index.html
create mode 100644 Perl-Ejercicios/.idea/.gitignore
create mode 100644 Perl-Ejercicios/.idea/PerlEjercicios.iml
create mode 100644 Perl-Ejercicios/.idea/inspectionProfiles/profiles_settings.xml
create mode 100644 Perl-Ejercicios/.idea/misc.xml
create mode 100644 Perl-Ejercicios/.idea/modules.xml
create mode 100644 Perl-Ejercicios/.idea/perl5local.xml
create mode 100644 Perl-Ejercicios/Problema1.pl
create mode 100644 Perl-Ejercicios/Problema2.pl

git
git branch -M main
git remote add origin https://github.com/rivX241/CGI-Perl.git
```

```
git push -u origin main
Enumerando objetos: 31, listo.
Contando objetos: 100% (31/31), listo.
Compresión delta usando hasta 12 hilos
Comprimiendo objetos: 100% (27/27), listo.
Escribiendo objetos: 100% (31/31), 7.95 MiB | 2.17 MiB/s, listo.
Total 31 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:rivX241/CGI-Perl.git
 * [new branch]      main -> main
rama 'main' configurada para rastrear 'origin/main'.
```

LINK: <https://github.com/rivX241/CGI-Perl/tree/main/Perl-Ejercicios>