

Redes de Computadoras
Trabajo práctico N°5 - 2023
Capa de aplicación. Servidores web.

Objetivo

- Comprender el funcionamiento del modelo cliente-servidor utilizado por muchas aplicaciones de la capa 5 del modelo TCP/IP.
- Instalar y configurar un servidor web.
- Crear páginas web sencillas con secuencias de instrucciones que se ejecuten del lado del cliente y del lado del servidor.
- Utilizar servicios SSH, FTP y VNC.
- Analizar el intercambio de paquetes de consultas DNS, peticiones HTTP y establecimiento y funcionamiento de conexiones TCP.

Metodología

Trabajo individual o grupal. 2 estudiantes por grupo máximo.

Tiempo de realización: 2 clases.

Aprobación

- Mostrar en clase la aplicación web (Actividad 2) funcionando correctamente.
- Subir los códigos fuente a Moodle.
- Elaborar un informe y subirlo a través de Moodle. El informe debe incluir:
 - Capturas de pantallas de las actividades realizadas en la actividad 1. Una captura por ítem.
 - Captura de pantalla de la página web creada en la Actividad 2.
 - Las capturas de pantalla solicitadas en la actividad 3 y respuestas a las preguntas planteadas en dicha actividad.

Materiales necesarios

- Dos computadoras con acceso a Internet. Una se utilizará como servidor, y deberá poseer sistema operativo Linux (altamente recomendado) o Windows. Se sugiere trabajar en grupos de a dos. También puede utilizarse como cliente un teléfono celular que cuente con cualquier navegador web.
- Servidor y cliente SSH (en el Anexo 1 se proveen instrucciones de instalación del servidor y cliente **OpenSSH**).
- Servidor FTP (en el Anexo 2 se proveen instrucciones de instalación del servidor **vsftpd** y el cliente **Filezilla**).

- Servidor y cliente VNC (en el anexo 3 se proveen instrucciones de instalación del servidor y cliente **RealVNC**).
- Servidor HTTP Apache y complementos para PHP (en el Anexo 4 se proveen instrucciones de instalación).
- Navegador web con soporte para Javascript (todos los navegadores modernos para computadoras o teléfonos celulares soportan Javascript).
- Analizador de tráfico Wireshark.

Actividad 1

1. Instale un servidor SSH y un cliente SSH en diferentes computadoras. Se sugiere realizar esta actividad en grupos de a dos, conectándose usted al servidor en la computadora de su compañera/o, y su compañera/o al servidor en su computadora. Una alternativa de trabajo es instalar un cliente SSH en su celular y conectarse al servidor en su computadora.

Se sugiere fuertemente instalar el servidor SSH en una computadora con sistema operativo Linux.

Ejecute varios comandos, entre ellos:

- a. Obtener IP, ver archivos en la carpeta descarga, etc. de la computadora donde se ejecuta el servidor.
- b. Cerrar programas que su compañero esté usando (sugerencia, utilice los comandos top, kill y grep entre otros).
- c. Apagar la computadora de su compañero.

Nota 1: En las computadoras del laboratorio de informática de la Facultad de Ingeniería deberá crear un nuevo usuario de Linux, ya que con el usuario estudiante no podrá realizar la conexión (ver en Anexo 6 instrucciones de como crear usuarios en Linux).

Nota 2: Si utiliza un servidor SSH en su computadora, desactive el servicio cuando ya no lo utilice. Desactívalo también desde el arranque. Un servidor SSH activo es un riesgo de seguridad.

2. Instale un servidor FTP y un cliente FTP en diferentes computadoras y ejecute varios comandos, entre ellos, transferir archivos (Vea instrucciones de instalación en el anexo 2).
3. Instale un servidor VNC y un cliente VNC en diferentes computadoras y ejecute varias acciones (Vea instrucciones de instalación en el anexo 3).

Actividad 2

1. Instale y configure Apache web server (Vea el Anexo 4 para instrucciones de instalación sobre Linux o Windows. Se sugiere fuertemente utilizar Linux).
2. Cree un archivo llamado “index.php” (o el nombre que quiera) en la carpeta de trabajo de Apache (Vea el Anexo 4 para instrucciones).
3. Edite el archivo anterior de manera de crear una página web que muestre lo siguiente: (Vea el Anexo 5 para instrucciones sobre cómo escribir una página web empleando lenguaje HTML. Si tiene conocimientos sobre algún Framework, puede usarlo) (ver Figura 1):
 - a. El logo de la UNCuyo, obtenido desde la página web de la UNCuyo (no como archivo en la computadora servidor).
 - b. Un formulario con un campo para ingresar un e-mail.
 - c. Una serie de opciones para que el usuario pueda elegir su equipo de fútbol favorito. Muestre el nombre del equipo y el escudo del mismo.
 - d. Un botón de “Enviar” .
 - e. Agregue una hoja de estilo simple que dé formato a su página web (puede utilizar algún estilo de los ejemplos que encontrará en: https://www.w3schools.com/css/css_form.asp).
4. Cuando el usuario presione el botón “Enviar”, funciones escritas en JavaScript deberán verificar, del lado del cliente, que el e-mail ingresado tiene el formato correcto, es decir, debe cumplir:
 - i. Tener al menos 7 caracteres.
 - ii. Tener una @, pero no al principio ni al final.
 - iii. Tener un punto, pero no al principio ni al final.
 - iv. No tener caracteres especiales (#, !, %, \$, etc.).Además, debe verificar que el usuario seleccionó una de las opciones. En caso de que el e-mail ingresado no cumplan alguno de estos requisitos, o el usuario no haya seleccionado una opción, la página web deberá indicar la condición no cumplida y solicitar corregir el error.
5. Si se ingresó un e-mail con formato correcto, se debe llamar a un procedimiento (o conjunto de procedimientos) escritos en PHP que:
 - a. Verifiquen que el e-mail ingresado no respondió anteriormente la encuesta. Si el e-mail ya ha respondido la encuesta, no contabilizar su voto y mostrar la leyenda “Usted ya ha respondido la encuesta”.
 - b. Si el e-mail ingresado no ha respondido la encuesta, contabilizar su voto.
 - c. Lleve una estadística de la cantidad de votos dados a cada equipo.

d. Muestre por pantalla los resultados de la encuesta.

Figura 1: Página web a construir

 127.0.0.1/index.php



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

Elija su equipo favorito

Ingrese su e-mail:

Boca Juniors



River Plate



San Lorenzo



Racing



Independiente



Otro

Enviar

Actividad 3

1. Ingrese a su página web desde otra computadora o teléfono celular. Utilice un analizador de tráfico de red (como Wireshark) para analizar los paquetes intercambiados. Busque los paquetes HTTP, específicamente:
 - a. La petición GET inicial y su respuesta.
 - b. La petición POST al enviar el formulario y su respuesta.

Tome captura de pantalla de la información mostrada por Wireshark para los 4 paquetes para incluir en el informe.

2. Conteste en el informe:
 - a. ¿Puede ver el contenido de las peticiones GET y POST?
 - b. ¿Puede ver las respuestas enviadas por el servidor?

Anexo 1: SSH

SSH es una aplicación cliente-servidor. La aplicación cliente permite ingresar comandos por consola que serán ejecutados en el servidor SSH. Los comandos son enviados sobre una conexión TCP, por lo que es necesario:

1. Que el servidor SSH esté corriendo y esperando conexiones en la computadora en la cual se ejecutarán los comandos.
2. Ejecutar el cliente SSH, indicando la IP (o dns) de la computadora en la cual corre el servidor SSH. Si no se indica el puerto, se asume que el servidor SSH espera conexiones en el puerto 22.
3. Ingresar el nombre de usuario y contraseña de la computadora en la cual corre el servidor SSH.

El cliente y el servidor pueden correr en diferentes sistemas operativos y ser programas de distintos desarrolladores.

Linux usualmente posee por defecto un cliente SSH que puede ejecutarse desde una terminal.

Linux:

1) Instalar e iniciar un servidor SSH sobre Linux

Para instalar un servidor ssh:

```
sudo apt install openssh-server
```

Para activar el servidor ssh:

```
systemctl start ssh
```

Para desactivar el servidor ssh:

```
systemctl stop ssh
```

Para desactivar el servidor ssh desde el arranque:

```
systemctl disable ssh
```

Para verificar el estado del servidor SSH

```
systemctl status ssh
```

2) Instalar un cliente SSH sobre Linux

Para instalar un cliente ssh:

```
sudo apt install openssh-client
```

3) Iniciar una conexión SSH

Desde una terminal, ejecute: `sudo ssh nombre_usuario_servidor@IP` (o DNS)

Ejemplo: `sudo ssh pablo@192.168.0.104` o `sudo ssh cnuncuyo@sdf.org`

Importante!: Finalice el servidor ssh cuando ya no lo use (sobre todo en equipos con Windows). Dejarlo abierto implica un riesgo de seguridad.

Windows 7

1) Instalar e iniciar un servidor SSH en Windows 7:

Descargue Bitvise Server desde <https://www.putty.org> (Instale la versión personal).

Inicie la aplicación "Bitvise SSH Server" y configure las siguientes opciones:

- En la pestaña Server, en "Instance type", elija "Master".
- En la misma pestaña, en "Settings", vaya a "Open easy settings", luego:
 - En la pestaña "Server settings" elija "Open port(s) to any computer"
 - En la pestaña "Windows accounts" elija "Allow login to any Windows account".
- Inicie el servicio yendo a "Bitvise SSH Server service is Stopped" eligiendo "Start Server".

Importante!: Finalice el servidor ssh cuando ya no lo use (sobre todo en equipos con Windows). Dejarlo abierto implica un riesgo de seguridad.

2) Cliente Windows 7:

Dos opciones: PuTTY o Bitvise Client. Puede encontrar ambos en <https://www.putty.org>.

PuTTY: En la categoría Session, indique la IP (o DNS) y haga clic en "Open". Se le pedirá el usuario y contraseña del servidor.

Bitvise Client: En la pestaña Login, indique la IP y el nombre de usuario (los demás datos se pedirán durante el proceso de conexión). Luego haga clic en "Log in". Una vez establecida la conexión, vaya a "New terminal console".

Windows 10:

1) Servidor SSH: Siga las instrucciones dadas en la página de Microsoft:

https://docs.microsoft.com/en-us/windows-server/administration/openssh/openssh_install_firstuse

O instale "Bitvise SSH Server" de la misma manera que con Windows 7.

2) Cliente SSH: Windows 10 posee un cliente SSH ya instalado que puede acceder desde la consola de comandos de Windows.

Si no está instalado por defecto, puede instalarlo como en Windows 7.

Cientes SSH para Android e iOS

Existen varios clientes SSH para Android e iOS que puede encontrar en las respectivas tiendas virtuales de dichos sistemas operativos .

Para Android se recomienda Mobile SSH (basado en PuTTY SSH).

Anexo 2: Servidor y cliente FTP

Servidor **vsftpd** para Linux

Para instalar el servidor FTP vsftpd ejecute:

```
sudo apt install vsftpd
```

Para activar el servidor FTP:

```
systemctl start vsftpd
```

Para desactivar el servidor:

```
systemctl stop vsftpd
```

Para desactivar el servidor desde el arranque:

```
systemctl stop vsftpd
```

Para verificar el estado del servidor SSH:

```
systemctl status vsftpd
```

Para permitir escrituras en la computadora donde está instalado el servidor FTP:

Por defecto, el servidor vsftpd no permite escribir en la computadora en la cual se ejecuta. Para permitir escrituras, debe editar el archivo `/etc/vsftpd.conf` (si accede por SSH, puede utilizar el programa `vim` con permisos de superusuario). En dicho archivo, debe descomentar la línea `write_enable=YES` (para descomentar, quite el símbolo `#` delante de la línea).

Nota: Si instala el servidor FTP en su computadora, desactívelo desde el arranque cuando ya no lo use.

Cliente FTP **Filezilla** para Linux y Windows

Como cliente se recomienda el cliente Filezilla.

Para Linux, puede instalarse con: *sudo apt install filezilla*.

Para Windows, descargarlo de su web oficial:

<https://filezilla-project.org/download.php?platform=win64>

Nota: Archivos con acentos o caracteres extraños podrían no transferirse o generar problemas.

Anexo 3: Servidor y cliente VNC

VNC (Virtual Network Computing) es un protocolo que permite acceder al escritorio de una computadora (servidor) desde otra computadora (cliente).

La computadora cliente puede ejecutar comandos sobre la computadora servidor. Existen varios softwares que permiten ejecutar VNC. Uno muy popular es RealVNC. Descargue e instale el cliente VNC (para Linux o Windows) desde <https://www.realvnc.com/es/>. En la pestaña “Descargar” encontrará el cliente (VNC Viewer) y el servidor (VNC Connect).

RealVNC también posee un servidor VNC, pero pedirá crear una cuenta. Para evitar este paso, se sugiere utilizar otro servidor VNC, por ejemplo x11vnc para Linux (se puede descargar desde los repositorios de Ubuntu).

Anexo 4: Instalación y puesta a punto de Apache en Linux y Windows

Instalación y puesta a punto de Apache y complementos para PHP en Linux (recomendado):

1 - Instale Apache y complementos para PHP con:

```
sudo apt install apache2
```

```
sudo apt install php libapache2-mod-php
```

2 - Una vez instalado inicie el servicio con: *sudo /etc/init.d/apache2 start*

o también con: *systemctl start apache2*

3 - Verifique el buen funcionamiento accediendo desde un navegador web en otra computadora o la misma computadora escribiendo en la barra de direcciones la URL:

```
http://<IP server>/index.html
```

Si utiliza una computadora conectada a una red Wifi, y posee otra computadora (puede ser un teléfono celular conectado a la misma red Wifi), puede acceder al servidor desde el navegador escribiendo la url indicada arriba.

Si la instalación fue correcta, deberá ver una página de Apache que dice: “[It works!](#)”

4 - El directorio con archivos del servidor Apache es: `/var/www/html`

Para más información, visite:

<http://httpd.apache.org/docs/2.4/>

La carpeta `/var/www/html/` posee permisos de escritura solo para superusuario. Se recomienda cambiar los permisos de modo que cualquier usuario pueda leer y escribir archivos en dicha carpeta (Para cambiar permisos, ejecutar `sudo chmod 777 archivo`).

5 - Importante!: Si trabaja en su computadora, no olvide detener el servicio cuando finalice sus trabajos.

Instalación y puesta a punto de Apache y complementos para PHP en **Windows** (No recomendado):

1 - Descargue el instalador de Apache con complementos para PHP desde:
<http://bitnami.com/stack/wamp>

Bitnami es un programa que incluye un servidor HTTP con soporte para PHP entre otras herramientas.

Luego instale Bitnami con el instalador descargado.

Si posee un Firewall instalado, deberá permitir acceso desde redes locales (el Firewall se lo preguntará).

2 - Una vez instalado inicie el servicio desde la aplicación Bitnami. Abra la aplicación y luego, en la pestaña “**Manage Servers**” haga clic en “**Start all**”.

3 - Verifique el buen funcionamiento accediendo desde un navegador web en otra computadora o la misma computadora escribiendo en la barra de direcciones la URL:

<http://<IP server>/index.html>

Si utiliza una computadora conectada a una red Wifi, y posee otra computadora (puede ser un teléfono celular conectado a la misma red Wifi), puede acceder al servidor desde el navegador escribiendo la url indicada arriba.

Si la instalación fue correcta, deberá ver la página de “Welcome” de Bitnami.

4 - El directorio con archivos del servidor Apache es:

</Bitnami/wampstack-7.3.17.1/apache2/htdocs>

Anexo 5: Escritura de páginas web empleando lenguaje HTML, CSS, Javascript y PHP

Nota: este anexo presenta un breve resumen de HTML, CSS, Javascript y PHP. Puede encontrar tutoriales completos y gratuitos de dichos lenguajes y muchos otros en

<https://www.w3schools.com/>

5.1 - Estructura de una página web escrita con HTML

HTML utiliza etiquetas para delimitar secciones. Por ejemplo, la etiqueta `<head>` inicia el código del encabezado de la página web, y `</head>` termina el código del encabezado. La siguiente es la estructura general de una página web:

```
<html>
```

```
<head>
```

Código de configuración, funciones de JavaScript, PHP, etc.

```
</head>
```

```
<body>
```

Código que implementa la página web (texto, botones, imágenes, PHP, etc.)

```
</body>
```

```
</html>
```

5.2 - Algunas etiquetas HTML que puede incluir en su página web

Párrafo: `<p>Texto a incluir en el párrafo</p>`

Imágenes: `` o

``

Salto de línea: `
`

Enlace: `texto del enlace`

`<center>elementos a centrar</center>`

`<H1>Títulos de mayor tamaño</H1>`

`<H2>Títulos de segundo mayor tamaño</H2>`, y así hasta `<H6></H6>`

`<div> elementos </div>` Añade una división lógica a su página web. Todos los elementos entre las etiquetas `<div>` `</div>` serán tratados como una unidad lógica.

Para una lista completa de etiquetas, visite <https://www.w3schools.com/tags/default.asp>

5.3 - Formularios HTML

Un formulario permite que el usuario interactúe con la página web y/o el servidor web. Permite agregar botones, cuadros de texto, controles de diferente tipo, etc. Los datos recolectados a través de un formulario pueden procesarse en el cliente a través de lenguajes del lado del cliente, como Javascript, o enviarse al servidor, para que sean procesados por lenguajes del lado del servidor, como PHP.

Usualmente los datos de un formulario serán analizados por un programa en Javascript en el mismo navegador del cliente, y si cumplen requisitos de formato esperados, serán enviados para ser procesados en el servidor por un programa escrito en PHP, ASP, etc.

Un formulario tiene la siguiente forma:

```
<form name="formulario1" id="formulario1_id" action="verificar.php" method="POST"
onsubmit="return validar();">
  <label for="nombre">Ingrese su nombre: </label>
  <input type="text" name="nombre" id="nombre_en_formulario1">
  <p><input type="submit"></p>
</form>
```

Veamos línea por línea:

```
<form name="formulario1" id="formulario1_id" action="verificar.php" method="POST"
onsubmit="return validar();">
```

name identifica de manera única al elemento dentro de un formulario (puede haber varios formularios en una página web y formularios dentro de formularios).

El *id* del formulario (en el ejemplo: *formulario1_id*) identifica de manera única al formulario dentro de un documento HTML. Es necesario para acceder un elemento desde funciones de JavaScript u otro lenguaje encriptado.

action indica la url del documento donde se encuentra el código que implementa los procedimientos a ejecutar en el servidor (en este caso, un archivo php llamado "verificar.php"). Si se omite, los códigos se encuentran en el mismo archivo.

method="POST". Indica la forma o método a través del cual se enviarán los datos al archivo que contiene el código que procesará los datos. Los métodos pueden ser "GET", que indica que los datos se incluyen dentro de la url o "POST", que indica que los datos se incluyen en el cuerpo del paquete http.

onsubmit="return funcion();" Indica que al apretar el botón "Enviar", antes de hacer el envío, debe llamarse una función de JavaScript llamada "funcion". Si dicha función devuelve "true", se llamará al archivo indicado por "action". Si devuelve "false", no se llamará al archivo indicado por "action". Es común llamar una función en JavaScript para verificar que los datos cumplen los formatos esperados antes de enviarlos al servidor.

```
<input type="text" name="nombre" id="id_del_elemento">
```

input type: agrega un elemento al formulario. Los elementos pueden ser: cuadros de texto para que el usuario ingrese un texto (input type="text"), un botón para enviar el formulario (input type="submit"), campo de texto para que ingresar contraseñas (input type="password"), etc.

Buscar otros elementos en:

https://www.w3schools.com/html/html_form_elements.asp

https://www.w3schools.com/html/html_form_input_types.asp

name identifica al elemento dentro del formulario. No puede haber dos elementos con igual nombre dentro de un mismo formulario, pero si pueden haber dos elementos con igual nombre en diferentes formularios.

id identifica al elemento dentro del documento HTML. No puede haber dos elementos con el mismo id dentro de un documento HTML, aunque estén en distintos formularios.

value se utiliza para dar o recuperar el valor del elemento. El valor dependerá del tipo de elemento. Por ejemplo, para un cuadro de texto, *value* será una cadena de caracteres.

```
<label for="nombre">Ingrese su nombre: </label>
```

Asigna una etiqueta (texto descriptivo) a un elemento. En el ejemplo, nombre puede ser un cuadro de texto, un botón, un cuadro para contraseñas, etc.

```
<p><input type="submit"></p>
```

Botón de envío. Al apretarlo, primero se llamará a la función indicada por *onsubmit="return funcion();"*, y si dicha función devuelve "true", se llamará al archivo indicado por *action*.

Nota: Al llenar o enviar el formulario puede aparecer un cartel que diga: "No seguro" o "Datos no cifrados". Verá en la unidad 7 cómo cifrar datos y hacer la conexión segura.

5.4 - Añadir una hoja de estilo

Cree un archivo con formato ".css". Dentro de dicho archivo, copie los estilos a agregar. Utilice los ejemplos que encontrará en https://www.w3schools.com/css/css_form.asp.

Si utiliza una hoja de estilo .css en un archivo separado del archivo principal de su página web, en el archivo .css no debe agregar las etiquetas `<style></style>`.

Para indicar al navegador que utilice la hoja de estilo creada, agregue la línea:

```
<link rel="stylesheet" type="text/css" href="estilo.css">
```

5.5 - Ejemplo código JavaScript

Para una introducción a Javascript, repase los conceptos vistos en clase.

Un curso completo y gratuito de Javascript puede encontrarse en <https://www.w3schools.com/js/default.asp>

Las funciones de JavaScript pueden insertarse dentro del documento principal (.html) o en un archivo separado (recomendado).

En caso de insertar las funciones de JavaScript dentro del documento principal, deben insertarse entre las etiquetas:

```
<script language="JavaScript">
```

.....

Funciones Javascriptp

.....

```
</script>
```

En caso de insertar las funciones de Javascript en un archivo separado, debe crear un archivo con extensión **.js** donde deberá incluir las funciones. Luego, en el documento principal, debe insertar:

```
<script src="archivo_con_funciones_javascript.js"></script>
```

pudiendo invocarse desde el cuerpo del documento (entre las etiquetas `<body></body>`) o desde el encabezado (entre las etiquetas `<head></head>`).

Si las funciones de Javascript se incluyen en un archivo separado, no son necesarias las etiquetas `<script language="JavaScript">` y `</script>` al comienzo y al final del código.

Función escrita en JavaScript

```
function mi_funcion(){
    var nombre_ingresado=document.getElementById("usuario_id").value;
    var clave_ingresada=document.forms["formulario1"]["clave"].value;
    if(variable_1<10){
        return true;
    }
    else{
        alert("El valor de variable_1 es mayor o igual a 10. Ingrese un valor menor a
10");
        return false;
    }
}
```

Los programas en JavaScript usualmente se insertan entre las etiquetas `<head></head>`. Generalmente consisten en una serie de funciones que se llaman desde distintas partes de un documento HTML. La sintaxis es similar a C.

Algunas sentencias y funciones son:

`var x=5;` declara una variable y le asigna el valor 5. Las variables pueden ser numéricas, texto, booleanas, listas, objetos o funciones. Todas se declaran con "var".

`alert("texto del mensaje" + variable1 + "otro texto");` es una función predefinida de JavaScript que muestra un mensaje por pantalla.

`var clave_ingresada=document.forms["nombre_formulario"]["nombre_elemento"].value;`

```
var nombre_ingresado=document.getElementById("elemento_id").value;
```

Las dos sentencias anteriores permiten recuperar el contenido de distintos elementos de formularios declarados dentro del documento HTML. Los elementos de un formulario pueden accederse por su nombre (name) o por su ID. El parámetro “name” de un elemento distingue elementos dentro de un formulario, por lo que si se usa este método, deberá indicarse el formulario en el cual está el elemento (como en el caso de la primera sentencia). Mientras que el ID distingue el elemento dentro de un documento HTML, por lo que si se usa este método, no es necesario hacer referencia al formulario (como en la segunda sentencia).

Algunas funciones para trabajar con cadenas:

cadena.length: devuelve la longitud de una cadena llamada “cadena”.

cadena.charAt(x) o ***cadena[x]***: devuelve el carácter en la posición x de la cadena “cadena”.

cadena.indexOf('x'): devuelve la posición del carácter x dentro de la variable “cadena” (indexOf es usada para buscar un elemento dentro de un array).

5.5.1 Expresiones regulares:

Se requiere conocimiento de programación orientada a objetos para comprender como funcionan las expresiones regulares.

Una expresión regular es un “patrón” contra el cual comparar. Se declaran entre barras: ***/expresión regular/***. JavaScript trata las expresiones regulares como objetos, por lo que ***/abc/*** declara un objeto del tipo expresión regular. Este objeto tiene varios métodos, como:

- ***expresion_regular.test(string)***: Busca coincidencias entre una expresión regular y una cadena de caracteres. Retorna True si hay al menos una coincidencia. Retorna False si no hay ninguna coincidencia.
- ***expresion_regular.exec(string)***: Busca coincidencias entre una expresión regular y una cadena de caracteres. Retorna un array con la coincidencia encontrada. Si no encuentra ninguna coincidencia, retorna **null**.
- ***string.search(expresion_regular)***: Busca coincidencias entre una expresión regular y una cadena de caracteres. Retorna la posición de la primera coincidencia.

Algunos caracteres, tales como * y \$ tienen funciones especiales, por lo que es necesario utilizar \ como secuencia de escape (esto indica que se va a usar la letra *, y no su función).

Los corchetes [] permiten especificar una lista de caracteres.

Ejemplos:

/[abc]/.test(usuario): devuelve true si la cadena usuario contiene una "a", una "b" o una "c".

/[A-Z]/.test(usuario): devuelve true si la cadena usuario contiene una letra mayúscula.

/[a-z]/.test(usuario): devuelve true si la cadena usuario contiene una letra minúscula.

/[1-9]/.test(usuario): devuelve true si la cadena usuario contiene un número del 1 al 9.

/@#|\\$!|%|*|.test(usuario): devuelve true si cadena contiene alguno de los símbolos @, #, \$, !, % o *.

/\D/.test(cadena): busca cualquier carácter que no sea un dígito (números del 0 al 9). Devuelve true si encuentra al menos 1.

/\d/.test(cadena): busca cualquier dígito (números del 0 al 9). Devuelve true si encuentra al menos 1.

5.5.2 Consola del navegador

`console.log("mensaje",variable1,variable2);`

Permite escribir en la consola del navegador (en Chrome: Herramientas del desarrollador ->Console). Muy útil en tiempo de depuración.

6 - Ejemplo de PHP

El código PHP se inserta entre los limitadores "`<?php`" y "`?>`". Puede agregarse en cualquier parte del documento HTML. Es un lenguaje muy potente, que permite realizar casi cualquier acción que realizan lenguajes como C, Python o Java. Puede incluirse el código PHP en un solo bloque o en varios bloques distribuidos en diferentes partes del documento. Dos formas de agregar código php en un programa son:

1) Código que elige que bloque de código HTML utilizar entre varios bloques de código HTML posibles.

```
<?php if ($clave==$clave_correcta): ?>
```

.....código HTML a generar en caso de que se cumpla la condición..... Nota, aquí puede agregarse código HTML o JavaScript.

```
<?php else: ?>
```

....código HTML a generar en caso de que no se cumpla la condición....

```
<?php endif ?>
```

Notar que en PHP las variables se escriben anteponiendo el símbolo \$ adelante. No es necesario declarar las variables antes de asignarles un valor.

2) Código que no elige entre distintas secuencias de código html.

```
<?php
```

```
    $clave_correcta="abcde";
```

```
    $clave = $_POST["clave"];
```

```
    echo "clave: ".$clave." clave_correcta: ".$clave_correcta;
```

```
if($clave==$clave_correcta){  
    header("Location: http://www.uncuyo.edu.ar");  
    $file = fopen("registro.txt","a");  
    fwrite($file,"Usuario ".$usuario." acceso correcto"."n");  
    fclose($file);  
}  
else{  
    .....  
}  
?>
```

En el ejemplo:

```
$clave = $_POST["clave"];  
$clave = $_GET["clave"];
```

Para recuperar datos enviados desde un formulario con el método POST o enviados mediante el método GET. Se utiliza `$_POST["nombre_elemento"]`; donde *nombre_elemento* es el nombre (name) de un campo de un formulario del cual se desea recuperar un dato, por ejemplo, un cuadro de texto. En el caso de utilizar el método GET, las variables deben pasarse a través de la url con el formato `nombre_variable=valor`.

```
echo "clave: ".$clave." incorrecta";
```

Escribe por pantalla. Similar a "print" de otros lenguajes. En PHP las cadenas de texto se concatenan con un punto (.). En el ejemplo se concatenan 3 cadenas: 1) La palabra "clave: ", 2) el valor de una variable llamada clave y 3) la palabra " incorrecta".

Usualmente "echo" se utiliza en tiempo de desarrollo y depuración, ya que una página web en su versión final se utiliza HTML o JavaScript para brindar mensajes "visualmente elegantes" al usuario.

```
header("Location: http://www.uncuyo.edu.ar");
```

Direcciona a otra página web. En el ejemplo, la página web de la UNCuyo.

```
$file = fopen("registro.txt","a");  
fwrite($file,"Usuario ".$usuario." acceso correcto"."n");  
fclose($file);
```

Estas tres sentencias permiten abrir, escribir y cerrar un archivo, de manera similar a otros lenguajes. Notar que "Usuario ".\$usuario." acceso correcto"."n" son 3 cadenas y

una variable concatenadas. El archivo a escribir debe tener permisos de escritura por cualquier usuario.

`$_SERVER['argumento']`

Es una variable definida por PHP, en la cual almacena información sobre la conexión. Argumento puede tomar valores como `SERVER_ADDR`, `SERVER_PORT`, `REMOTE_ADDR`, `REMOTE_PORT`, etc.

Puede encontrar más información sobre PHP u otros lenguajes en <https://www.w3schools.com/php/default.asp>

Anexo 6: Crear un usuario nuevo en Linux Ubuntu

6.1 Crear un nuevo usuario

1. Ir a configuración, luego usuarios.
2. Haga click en desbloquear, luego vaya a “agregar usuario”
3. Elija un nombre de usuario que recuerde.
4. Haga clic en “establecer una contraseña ahora” y agregar una contraseña. Se sugiere la contraseña **computer132***. Luego haga click en añadir.
5. En la siguiente pantalla, habilite la opción “administrador”.

6.2 Cambiar de usuario

1. Ir a apagar, luego “cambiar usuario”.
2. Elija el nuevo usuario creado e ingrese.