PABLO HORCAJADA GONZALEZ

Ejercicio 3

Contenido

[Enunciado 2](#_Toc88387982)

[Variables 3](#_Toc88387983)

[Funciones 3](#_Toc88387984)

[Bucles 4](#_Toc88387985)

[Final 4](#_Toc88387986)

[Anexo 5](#_Toc88387987)

[Figura 1 5](#_Toc88387988)

[Figura 2 5](#_Toc88387989)

[Figura 3 5](#_Toc88387990)

[Figura 4 5](#_Toc88387991)

[Figura 5 5](#_Toc88387992)

[Figura 6 5](#_Toc88387993)

[Figura 7 5](#_Toc88387994)

[Figura 8 5](#_Toc88387995)

[Figura 9 6](#_Toc88387996)

[Figura 10 6](#_Toc88387997)

[Figura 11 6](#_Toc88387998)

[Figura 12 6](#_Toc88387999)

[Figura 13 6](#_Toc88388000)

[Figura 14 6](#_Toc88388001)

[Figura 15 6](#_Toc88388002)

# Enunciado

Realiza un aplicativo web en HTML5 y JavaScript en el cual se pida lo siguiente:

* Pedir por pantalla el nombre de usuario y contraseña mediante entradas texto.
* Comprobar que el usuario solo esté formado por minúsculas y números.
* Si no se cumple el paso anterior, se producirá un error y se podrá de nuevo el nombre.
* La contraseña de contener una letra minúscula y otra mayúscula.
* La contraseña debe contener un número y un carácter que no sea ni letra ni número.
* Si la contraseña no cumple con lo indicado, pediremos que vuelva a introducirla.
* SI cumple todo lo anterior, diremos que ha sido almacenado el usuario y la contraseña.
* Ante cualquier cancelación, la aplicación se termina y se notifica al usuario.

# Variables

Para el programa necesitaremos un total de 8 variables. (Figura 1):

* Dos variables para el texto que introduce el usuario. (Figura 2)
* Dos variables que contendrán las letras del abecedario en minúsculas y mayúsculas. (Figura 3)
* Cuatro variables que servirán para comprobar si la validación está bien o no. (Figura 4)

# Funciones

-Necesitaremos un total de 3 funciones para realizar el programa:

1. **Function contieneNumyMinus ()** (Figura 5). Esta función, recibirá como parámetro un texto (Figura 6), pondrá las variables (correspondiente a las validaciones) a false (Figura 7), recorrerá el texto recibido (carácter a carácter) comprobando si es un número o una minúscula (si el String “letras” contiene el carácter o dicho carácter es un número), y si alguna de las dos lo cumple, cambiara su determinada validación a true. (Figura 8)
2. **Function contieneMayus ()** (Figura 9). Esta función sería prácticamente igual a la función anterior, salvo cambiando el String que recorrerá para la validación (comprobará el String “mayus”).
3. **Function noContiene ()** (Figura 10). Esta función funcionaria igual que la 1º función, pero en este caso, ejecutamos las dos funciones anteriores (pasándoles como parámetro cada carácter del texto) para comprobar si el carácter es una mayúscula, minúscula y número (Figura 11), añadiendo además que, si no se cumple ninguna validación de las funciones, significa que contiene un carácter que no sea ni letra ni número (Figura 12).

# Bucles

Necesitaremos un total de 2 bucles para realizar el programa:

1º Bucle 🡪 Pedirá por pantalla el nombre de usuario, ejecutará la función contieneNumyMinus pasándole como parámetro el nombre del usuario que se ha escrito, comprobará si el usuario no cumple las condiciones (mostrando un mensaje) y comprobará si el usuario ha cancelado el programa hasta que las condiciones del usuario sean correctas. (Figura 13)

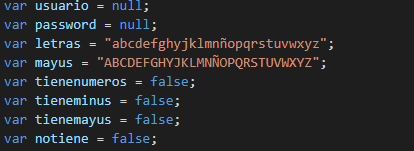
2ºBule 🡪 Pedirá por pantalla la contraseña, ejecutará las funciones noContiene (), contieneNumyMinus () y contieneMayus () pasándoles como parámetro la contraseña escrita por el usuario, comprobará si las condiciones de la contraseña no son correctas (mostrando un mensaje) y comprobará si el usuario ha cancelado el programa hasta que las condiciones de la contraseña se cumplan. (Figura 14)

# Final

Para acabar el programa, necesitaremos de un If (que se ejecutará después de todos los bucles) que comprobará si el usuario y la contraseña escritas no están vacías, si es el caso, mostrará un mensaje por pantalla e imprimirá el usuario y la contraseña que el usuario ha escrito. (Figura 15)

# Anexo

## Figura 1



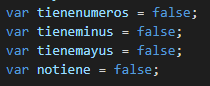
## Figura 2



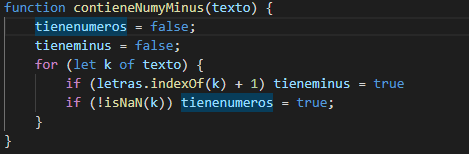
## Figura 3



## Figura 4



## Figura 5



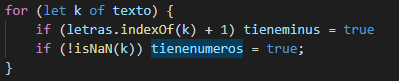
## Figura 6



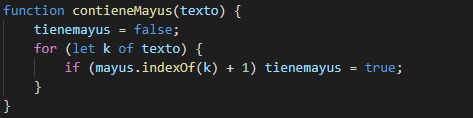
## Figura 7



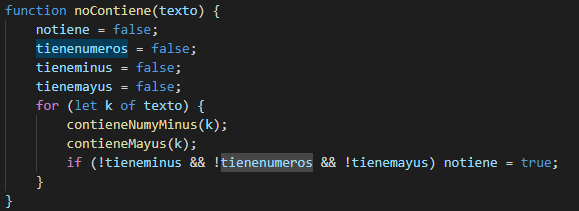
## Figura 8



## Figura 9



## Figura 10



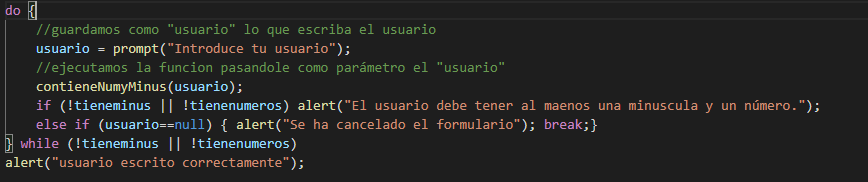
## Figura 11



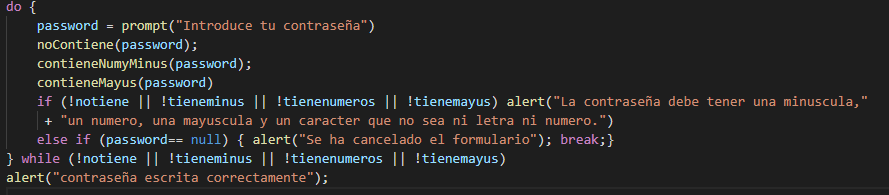
## Figura 12



## Figura 13



## Figura 14



## Figura 15

