
 CFIP Majada Marcial	ACTIVIDADES EN EL AULA			 Gobierno de Canarias
		UT2_A3			
		Edición: 1	Fecha: 28-03-2017	Página 1 de 21	

NOMBRE ALUMNO/A _Iván Hernández Fuentes

Curso 2018/2019	ACTIVIDADES	CALIFICACIÓN
Fecha:	Grupo: 2º DEW	

MÓDULO	UNIDAD DE TRABAJO	
DEW	UT2	A3
A TENER EN CUENTA:		
1) Antes de comenzar rellene Nombre y Apellidos .		

La tarea consiste en la implementación de varios scripts en lenguaje Javascript. En caso de no tener clara la interpretación de alguno de los enunciados, se deja al alumno libertad para llevar a cabo la solución que interprete más adecuada, siempre y cuando se justifique convenientemente (preferiblemente mediante comentarios en el código).

1. Crea un script que pida al usuario un número entero entre 1 y 100. A continuación se mostrará un mensaje que indique si el número:
 - a. Es divisible entre 2.
 - b. Es divisible entre 3.
 - c. Es divisible entre 5.
 - d. Es divisible entre 10.

Indicaciones:

- Un número es divisible entre N cuando el resto de dividir el número entre N es 0.
- En caso de que sea divisible entre varios números, se mostrarán varios mensajes.
- En caso de que el número no esté entre 1 y 100 o no se introduzca un número, se mostrará un mensaje de error indicativo.

Función	Recibe	Devuelve	Descripción
esDivisible	dividendo:entero divisor:entero	booleano	Devuelve verdadero si el primer parámetro es divisible entre el segundo y falso en caso contrario

Comprobamos el funcionamiento del programa.

Metemos un numero mayor

Esta página dice

Introduzca un número entre 1 y 100

Aceptar
 Cancelar

Esta página dice

Ha ocurrido un error

Aceptar

Introducimos números validos

Esta página dice

Introduzca un número entre 1 y 100

Aceptar
 Cancelar

Esta página dice

Introduzca un número entre 1 y 100

Aceptar
 Cancelar

Es divisible entre 2

Es divisible entre 2

Es divisible entre 3

Código

```

<script>

var numero = prompt("Introduzca un número entre 1 y 100");

if(numero < 1 || numero > 100 || numero == ""){
  alert("Ha ocurrido un error")
}

var divisor = [2,3,5,10]; //creamos un array de los divisores

function esDivisible(numero,divisor){ //la funcion toma como dos argumentos el numero introducido y el divisor del array
  for(var i = 0; i<= divisor.length;i++){ //creamos un for que recorra el array y mientras el numero / el divisor = 0 mostrara con quien
    //es divisible, sumaremos +1 a la i para que siga recorriendo
    while(numero % divisor[i] == 0){
      console.log("Es divisible entre " + divisor[i]);
      i++;
    }
  }
}

esDivisible(numero,divisor);

```

- Modifica el primer ejercicio repitiendo la petición de datos hasta que se introduzca un número en el rango indicado.

Por otra parte, en lugar de mostrar varios mensajes, se fusionará en uno solo. Además, en caso de que no sea divisible entre ninguno de los citados, se mostrará un mensaje indicativo.

El número 17 no es divisible entre 2, 3, 5 ni 10
☐ Evita que esta página cree cuadros de diálogo adicionales.

El número 50 es divisible entre 2 5 10
☐ Evita que esta página cree cuadros de diálogo adicionales.

Función	Recibe	Devuelve	Descripción
enRango	numero:entero maximo:entero minimo:entero	booleano	Devuelve verdadero si el número está en el rango entre máximo y mínimo, y falso en caso contrario.

```
El numero 24 es divisible entre 2
El numero 24 es divisible entre 3
El numero 24 no es divisible entre 5
El numero 24 no es divisible entre 10
```

```
<script>
  //repite la petición de numero mientras el numero que introduce el usuario no este en los rangos permitidos
  do {
    var numero = prompt("Introduzca un número entre 1 y 100");
  }while(numero < 1 || numero > 100 || numero == "");

  var divisor = [2,3,5,10]; //creamos un array de los divisores

  function esDivisible(numero,divisor){ //la funcion toma como dos argumentos el numero introducido y el divisor del array
    for(var i = 0; i<= divisor.length;i++){ //creamos un for que recorra el array y mientras el numero / el divisor = 0 mostrara con quien
      //es divisible, sumaremos +1 a la i para que siga recorriendo
      while(numero % divisor[i] == 0){
        console.log("El numero " + numero + " es divisible entre " + divisor[i]);
        i++;
      }
      if(numero % divisor[i] != 0){
        console.log("El numero " + numero + " no es divisible entre " + divisor[i]);
      }
    }
    i++;
  }

  console.log(esDivisible(numero,divisor));
}
```

3. Implementa un script que pida un número entre 1 y 100 e indique si el número es primo. Un número primo solo es divisible entre sí mismo y la unidad.

Función	Recibe	Devuelve	Descripción
esEntero	numero:entero	booleano	Comprueba si el número que recibe como parámetro es entero, devolviendo falso en caso contrario.
esPrimo	numero:entero	Booleano	Comprueba si el número que recibe como parámetro es primo, devolviendo falso en caso contrario.

Probamos nuestro programa.

Esta página dice

Introduzca un numero del 1 al 100

Aceptar Cancelar

Esta página dice

Introduzca un numero del 1 al 100

Aceptar Cancelar

```

true
true
>
true
false
>

```

Devuelve true en
ambos(es primo)

Nos devuelve True y false(es
entero , no es primo)

Código

```

<script>

do{
    var numero = prompt("Introduzca un numero del 1 al 100");
}while(numero < 0 || numero > 100 || numero == "");

function esEntero(numero){
    if(numero % 1 == 0){
        return true;
    }
    return false;
}

```

```
function esPrimo(numero){
  for(var i = 2; i < numero; i++){
    if(numero % i == 0){
      return false;
    }
  }
  return true;
}

console.log(esEntero(numero));
console.log(esPrimo(numero));

</script>
```

4. Implementa un script que pida el tamaño de una pantalla de un dispositivo portable (en pulgadas). Habrá que validar que el número sea positivo (no tiene por qué ser entero).

Una vez leído el tamaño se dará un mensaje indicativo del tipo de dispositivo del que se trata:

- En caso de que el tamaño sea menor o igual a 6 pulgadas, se trata de un Smartphone.
- En caso de que el tamaño esté entre 6 y 11 pulgadas (incluido), se trata de una tablet.
- Si estamos entre 11 y 13 pulgadas (incluido), se trata de un netbook.
- A partir de 13 pulgadas se trataría de un portátil.



Función	Recibe	Devuelve	Descripción
pideTamanoPantalla	-	Real	Pide el tamaño de la pantalla y devuelve el mismo, previamente validado.
esPositivo	numero:entero	booleano	Comprueba si el número que recibe como parámetro es positivo.

Probamos con 5.5 pulgadas

Esta página dice

Introduce el tamaño de tu pantalla

Aceptar
 Cancelar



top

es un smartphone

Probamos con 17 pulgadas

Esta página dice

Introduce el tamaño de tu pantalla

Aceptar
 Cancelar





es un portatil

Probamos con 12.5 pulgadas

Esta página dice

Introduce el tamaño de tu pantalla

Aceptar
 Cancelar



top


es un netbook

Probamos con 7 pulgadas.

Esta página dice

Introduce el tamaño de tu pantalla

Aceptar
 Cancelar



Es una tablet

Código

```

<script>

var tamaño = prompt("Introduce el tamaño de tu pantalla"); //pido el numero por pantalla

function esPositivo(tamaño){ //le paso a la funcion el parametro que solicito al usuario y empiezo con las comparaciones

    if(tamaño <= 6){

        console.log("es un smartphone");

    } else if(tamaño >= 6 && tamaño <=11){

        console.log("Es una tablet");

    } else if(tamaño >=11 && tamaño <=13){

        console.log("es un netbook");

    } else if(tamaño > 13){

        console.log("es un portatil");

    }

}

console.log(esPositivo(tamaño)); //llamo a la función y le asigno como argumento la variable que introduce el usuario

</script>

```

5. Implementa un script con una pequeña calculadora que haga las siguientes operaciones con enteros:
- Pedir un operando.
 - Pedir otro operando
 - Pedir la operación a realizar (puede ser suma, resta, multiplicación o división).

Introduce el primer operando

Introduce el segundo operando

☐ Evita que esta página cree cuadros de diálogo adicionales.

Introduce la operación a realizar

☐ Evita que esta página cree cuadros de diálogo adicionales.

A continuación se mostrará por pantalla el resultado de la operación.

Ten en cuenta que será necesario validar:

- Que los operandos y el operador no sean vacíos.
- Que el operador sea uno de los permitidos (+, -, *, /).
- En caso de que el operador sea la división, que el segundo operando no sea nulo.

Si alguna de las validaciones no es correcta, el programa finalizará.

Función	Recibe	Devuelve	Descripción
validaOperador	operador:cadena	booleano	Valida que el operador sea permitido
validaOperando	numero:entero	booleano	Valida que el operando sea válido
divisionValida	dividendo:entero divisor:entero	booleano	Valida que la división sea válida.

Comprobamos realizando la suma.

<p>Esta página dice</p> <p>Introduce el primer operador</p> <input type="text" value="5"/>	<p>Esta página dice</p> <p>Introduce el segundo operador</p> <input type="text" value="5"/>
<p>Aceptar Cancelar</p>	<p>Aceptar Cancelar</p>

<p>Esta página dice</p> <p>Introduce el signo de operacion a realizar</p> <input type="text" value="+"/>	<p>Esta página dice</p> <p>Operador valido</p>
<p>Aceptar Cancelar</p>	<p>Aceptar</p>

<p>Esta página dice</p> <p>10</p>	<p>Aceptar</p>
-----------------------------------	----------------

Introducimos un operador no valido.

<p>Esta página dice</p> <p>Introduce el primer operador</p> <input type="text" value="10"/>	<p>Esta página dice</p> <p>Introduce el segundo operador</p> <input type="text" value="10"/>
<p>Aceptar Cancelar</p>	<p>Aceptar Cancelar</p>

<p>Esta página dice</p> <p>Introduce el signo de operacion a realizar</p> <input type="text" value="o"/>	<p>Esta página dice</p> <p>operador no valido</p>
<p>Aceptar Cancelar</p>	<p>Aceptar</p>

Introducimos parámetros vacíos

<p>Esta página dice</p> <p>Introduce el primer operador</p> <input type="text" value=""/>	<p>Esta página dice</p> <p>Introduce el segundo operador</p> <input type="text" value=""/>
<p>Aceptar Cancelar</p>	<p>Aceptar Cancelar</p>

<p>Esta página dice</p> <p>parametros vacios</p>	<p>Aceptar</p>
--	----------------

Realizamos la división (normal)

Esta página dice

Introduce el primer operador

Aceptar

Cancelar

Esta página dice

Introduce el segundo operador

Aceptar

Cancelar

Esta página dice

Introduce el signo de operacion a realizar

Aceptar

Cancelar

Esta página dice

Operador valido

Aceptar

Esta página dice

0.5

Aceptar

Realizamos la división con un valor nulo.

Esta página dice

Introduce el primer operador

Aceptar

Cancelar

Esta página dice

Introduce el segundo operador

Aceptar

Cancelar

Esta página dice

Introduce el signo de operacion a realizar

Aceptar

Cancelar

Esta página dice




Operador valido

Aceptar

Esta página dice

El operador es nulo no se resuelve la división

Aceptar

	 CIPF Majada Marcial	ACTIVIDADES EN EL AULA			 Gobierno de Canarias
		UT2_A3			
		Edición: 1	Fecha: 28-03-2017	Página 10 de 21	

Código

```

<script>

var operador1 = prompt("Introduce el primer operador"); //pedimos al usuario los dos operadores
var operador2 = prompt("Introduce el segundo operador");

if(operador1 == "" || operador2 == ""){ //comprobamos si alguno de los operadores está vacío en ese caso se notifica el error
    alert("parametros vacios");
}

var operador = prompt("Introduce el signo de operacion a realizar"); //el usuario introduce el signo de la operación a realizar

//validacion del operador

function validaOperador(operador){

    var operadores = ["+", "-", "*", "/"]; //he creado un array con los signos de las operaciones permitidos

    if(operadores.includes(operador)){ //hacemos una comparacion si el operador del usuario se incluye en el array operadores se notifica
        alert("Operador valido");

        return true;
    }else if(operador != operadores.includes(operador) || operador == "") { //en otro caso se le dice que no vale
        alert("operador no valido");

        return false;
    }
}

//valida la operacion

function validaOperando(operador1,operador2){ //entran dos argumentos a la funcion operador1 y operador2

    var calculo = operador; //asigno a una variable el operador y empiezo a comparar
    if(calculo == "*"){

        alert(operador1 * operador2);

    }else if(calculo == "+"){

        alert(parseInt(operador1) + parseInt(operador2));

    }else if( calculo == "-"){

        alert(operador1 - operador2);

    }else if(calculo == "/"){

        divisionValida(operador1,operador2);

    }

    return true;
}

//compruebo divisor
function divisionValida(operador1,operador2){

    if(operador2 == ""){

        alert("El operador es nulo no se resuelve la división");

        return false;

    }else {

        var division = operador1 / operador2;
        alert(division);

    }

    return true;
}

console.log(validaOperador(operador));
console.log(validaOperando(operador1,operador2));

</script>

```

6. Modifica el script anterior para añadir un nuevo operador llamado FIN que permita finalizar el script. Por tanto, una vez realizada una operación se seguirán realizando operaciones hasta que el operador introducido sea FIN. Además, en caso de que alguna de las validaciones sea incorrecta, se seguirá pidiendo el dato hasta que este sea correcto.

En mi caso yo he utilizado un while en donde mientras se cumpla una condición que en este caso es mientras no se introduzca en los operadores1 y operador2 la palabra "fin" continúe el programa sucesivamente.

Esta página dice

Introduce el primer operador

Aceptar

Cancelar

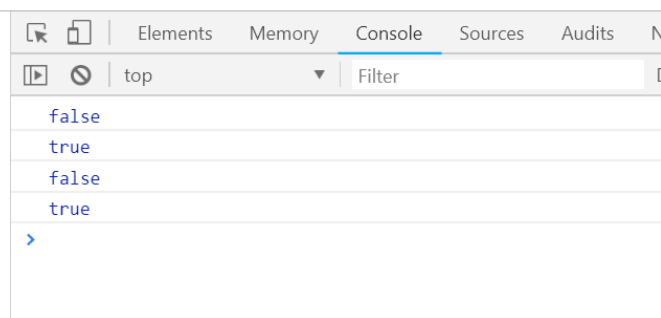
Esta página dice

Introduce el segundo operador

Aceptar

Cancelar

Se acaba el programa



Código

```
while(operador1 != "fin" || operador2 != "fin"){ //mientras el usuario no introduzca fin el script seguira

    var operador1 = prompt("Introduce el primer operador");
    var operador2 = prompt("Introduce el segundo operador");

    if(operador1 == "" || operador2 == ""){
        alert("parametros vacios");
    }else{
        var operador = prompt("Introduce el signo de operacion a realizar");
    }
}
```

7. Implementa un script gestione un control de acceso básico a una aplicación. Se pedirán (en dos ventanas distintas) un login y una contraseña:

Nombre de usuario

Aceptar Cancelar

Contraseña

☐ Evita que esta página cree cuadros de diálogo adicionales.

Aceptar Cancelar

El script tendrá definidas dos constantes que almacenen respectivamente login y una contraseña¹. En función de la información introducida, habrá que informar al usuario acerca de la validez de la misma, teniendo en cuenta las siguientes posibilidades:

El nombre de usuario y la contraseña son válidas
☐ Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

El nombre de usuario no existe
☐ Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

La contraseña no es válida
☐ Evita que esta página cree cuadros de diálogo adicionales.

Aceptar

Función	Recibe	Devuelve	Descripción
existeLogin	login:cadena	Booleano	Valida que el login exista
passwordCorrecto	password:cadena	Booleano	Valida que el password sea correcto.

Probamos el script con parámetros correctos

Esta página dice
 Nombre de usuario

Esta página dice
 Contraseña

Aceptar Cancelar

Esta página dice
 el nombre de usuario y la contraseña son correctos

Aceptar

¹ Sobra decir que es una mala práctica, dado que la contraseña se descarga al equipo del cliente y cualquier persona puede verla. Se plantea el ejemplo únicamente con fines didácticos.

Probamos con datos erróneos

Esta página dice

Nombre de usuario

Esta página dice

Contraseña

Esta página dice

El nombre de usuario no existe

Código

```

var usuario = prompt("Nombre de usuario"); //pedimos el usuario y contraseña al usuario
var contrasenia = prompt("Contraseña");

function existeLogin(usuario){ //comprobamos si coincide el login con el login que hemos creado
  const user = "ivan";
  if(usuario == user ){
    return true;
  }else {
    alert("El nombre de usuario no existe");
    return false;
  }
}

function passwordCorrecto(contrasenia){ //lo mismo con la contraseña
  const password = "1234";
  if(contrasenia == password){
    return true;
  }else {
    alert("la contraseña no coincide");
    return false;
  }
}

if(existeLogin(usuario) && passwordCorrecto(contrasenia)){
  alert("el nombre de usuario y la contraseña son correctos");
}

</script>
</body>

```

8. Modifica el script anterior dando un máximo de 5 intentos para que el usuario introduzca correctamente los datos.

Simplemente he creado una variable que se llama intentos y cada vez que falla se incrementa a uno, mientras los intentos sean menores a 5 puede seguir intentándolo.

```
<script>
var intentos = 0; //creamos una variable de intentos

while(intentos <= 5){ //mientras los intentos sean menor a 5 seguira pidiendo al usuario toda la informacion

    var usuario = prompt("Nombre de usuario");
    var contrasenia = prompt("Contraseña");

    function existeLogin(usuario){

        const user = "ivan";
        if(usuario == user ){

            return true;
        }else {

            alert("El nombre de usuario no existe");
            intentos++; //si no existe el usuario se suma un intento fallido
            return false;
        }
    }

    function passwordCorrecto(contrasenia){

        const password = "1234";

        if(contrasenia == password){

            return true;
        }else {

            alert("la contraseña no coincide");
            intentos++; //si la contraseña no coincide se suma otro intento fallido

            return false;
        }
    }

    if(existeLogin(usuario) && passwordCorrecto(contrasenia)){

        alert("el nombre de usuario y la contraseña son correctos");

    }

}

</script>
```

9. Modifica el script anterior implementando el ejercicio de la siguiente manera:
- En lugar de un único login tendremos un array de 5 logins, definido estáticamente al comienzo del script.

pepe	paco	juan	benito	manuel
------	------	------	--------	--------

- Además tendremos un array de 5 contraseñas. La contraseña en la posición i-ésima del array de contraseñas se corresponderá con el login ubicado en la posición i-ésima del array de logins. Por ejemplo, benito tendrá la contraseña 67890.

12345	23456	09876	67890	666666
-------	-------	-------	-------	--------

El script tendrá funcionamiento análogo al ejercicio anterior, pero habrá que comprobar si el usuario existe en el array de logins y si la contraseña que se introduce es la del usuario correcto.

Queda a criterio del alumno la modificación de las funciones usadas en el ejercicio 8.

Probamos el funcionamiento con el usuario BENITO y su contraseña

Esta página dice

Nombre de usuario

Esta página dice

Contraseña

Esta página dice

usuario valido

Esta página dice

La contraseña 67890 coincide con el usuario benito

Esta página dice

el nombre de usuario y la contraseña son correctos

Probamos con el usuario PACO en donde su contraseña debería de ser 23456

Esta página dice

Nombre de usuario

Esta página dice

Contraseña

Esta página dice
usuario valido

Aceptar

Esta página dice
La contraseña 23456 coincide con el usuario paco

Aceptar

Esta página dice
el nombre de usuario y la contraseña son correctos

Aceptar

Probamos otra vez con BENITO pero con una contraseña diferente.

Esta página dice
Nombre de usuario

benito

Aceptar

Cancelar

Esta página dice
Contraseña

666666

Aceptar

Cancelar

Esta página dice
usuario valido

Aceptar

Esta página dice
La contraseña no coincide con el usuario

Ahora introduciremos un usuario que no existe.

Esta página dice
Nombre de usuario

ivan

Aceptar

Cancelar

Esta página dice
Contraseña

1234

Aceptar

Cancelar

Esta página dice
El nombre de usuario no existe

Aceptar

Esta página dice
La contraseña no coincide con el usuario

Aceptar

Código

```
while(intentos <= 5 && intentos !=0){

    var usuario = prompt("Nombre de usuario"); //pedimos al usuario el login y password
    var contraseña = prompt("Contraseña");

    var nombres = ["pepe","paco","juan","benito","manuel"]; //creo un array nombres y otra de contra
    var contra = ["12345","23456","09876","67890","666666"];

    function existeLogin(usuario){ //una funcion que comprueba que el login del usuario esta en el array usuario

        if(nombres.includes(usuario) ){
            alert("usuario valido");
            return true;
        }else {

            alert("El nombre de usuario no existe");
            intentos++;
            return false;
        }

    }

}
```



```
function passwordCorrecto(contrasenia){ //en la contraseña he creado un doble bucle que recorrerá los dos arrays y comprobará si lo c
//usuario coincide con las posiciones del array
  for(var i = 0; i<nombres.length;i++){
    for(var j = 0; j < contra.length; j++){
      // console.log(nombres[i],contra[i]);
      if(contrasenia == contra[i] && usuario == nombres[i]){
        alert("La contraseña " + contrasenia + " coincide con el usuario " + usuario);
        intentos = 0;
        return true;
      }
    }
  }
  return false;
}

//si las funciones devuelven true se notifica al usuario
if(existeLogin(usuario) && passwordCorrecto(contrasenia)){
  alert("el nombre de usuario y la contraseña son correctos");
}

//si la funcion comprueba password devuelve false se notifica al usuario
if(!passwordCorrecto(contrasenia)){
  alert("La contraseña no coincide con el usuario");
  intentos++;
}
}
```

10. Modifica el script anterior para que funcione en dos fases:

- En una primera fase se van a introducir los logins y contraseñas. Se pedirá por pantalla el número de usuarios que va a tener nuestra aplicación (como máximo podemos tener 10 usuarios). A continuación se irán introduciendo de manera secuencial los login y las contraseñas. Habrá que validar que tanto el login como la contraseña sean no vacíos y tengan una longitud mayor de 5 caracteres.
- Posteriormente se realizará el proceso de login al igual que en el ejercicio anterior.

Queda a criterio del alumno la implementación de las funciones necesarias.

Introducimos un número superior a 10

Esta página dice

Cuantos usuarios vas a introducir

Aceptar

Cancelar

Esta página dice

el numero de usuarios no puede superar los 10 o no puede estar vacío

Aceptar

Introducimos 4 usuarios.

(El primero con una longitud mayor a 5 caracteres)

Esta página dice

nombre de usuario

Aceptar

Cancelar

Esta página dice

nombre no valido

Aceptar

Nos vuelve a pedir un usuario

Esta página dice

nombre de usuario

Aceptar

Cancelar

Esta página dice

nombre valido

Aceptar

Esta página dice

Contraseña

Aceptar

Cancelar

Esta página dice

contraseña valida

Aceptar

Introducimos al usuario tony

Esta página dice

nombre de usuario

Aceptar

Cancelar

Esta página dice

nombre valido

Aceptar

Esta página dice

Contraseña

Aceptar

Cancelar

Esta página dice

contraseña valida

Aceptar

Introducimos al usuario Jesús con una contraseña mayor a 5 caracteres

Esta página dice

nombre de usuario

Esta página dice

nombre valido

Aceptar

Cancelar

Aceptar

Esta página dice

Contraseña

Esta página dice

Contraseña no valida

Aceptar

Cancelar

Aceptar

Iniciamos sesión

Esta página dice

Vas a Iniciar sesion

Aceptar

Esta página dice

Nombre de usuario

Esta página dice

usuario valido

Aceptar

Cancelar

Aceptar

Esta página dice

Contraseña

Esta página dice

usuario valido

Aceptar

Cancelar

Aceptar

Esta página dice

Los datos son correctos

Aceptar

Código

```
<script>
    var numusuario = prompt("Cuantos usuarios vas a introducir");

    if(numusuario > 10 || numusuario == ""){

        alert("el numero de usuarios no puede superar los 10 o no puede estar vacio");

    }else{

        var contador = 1;        //contador para comparar el numero de usuarios
        var nombres = [];        //array de nombres
        var passwords = [];      //array de contraseñas

        function validaNombre(nombre){        //validamos el nombre del usuario

            if(nombre == "" || nombre.length > 5){

                alert("nombre no valido");

                return false;

            }

            return true;

        }

        function validaPass(pass){ //validamos la contraseña

            if(pass == "" || pass.length > 5){

                alert("Contraseña no valida");

                return false;

            }

            return true;

        }

        while(contador <= numusuario){        //pedira el nombre de usuario y contraseña tantas veces como el usuario dijo

            contador++;

            //nombre
            var nombre = prompt("nombre de usuario");

            if(validaNombre(nombre)){        //si la funcion valida nombre devuelve TRUE el nombre es valido y se añade al array con PUSH

                alert("nombre valido");
                nombres.push(nombre);
                var pass = prompt("Contraseña");

                if(validaPass(pass)){        //si la funcion devuelve TRUE la contraseña es valida y se añade al array con PUSH




                    alert("contraseña valida");
                    passwords.push(pass);

                }

            }

        }

    }
}
```

	 CIFP Majada Marcial	ACTIVIDADES EN EL AULA			 Gobierno de Canarias
		UT2_A3			
		Edición: 1	Fecha: 28-03-2017	Página 21 de 21	

```

    }

    //EL USUARIO INICIA SESIÓN

    alert("Vas a Iniciar sesion");

    var login = prompt("Nombre de usuario");
    var contrasenia = prompt("Contraseña");

    function existeLogin(login){    //una funcion que comprueba que el login del usuario esta en el array usuario

        if(nombres.includes(login) ){ //si el login se incluye en el array devuelve true

            alert("usuario valido");
            return true;

        }else {

            alert("El nombre de usuario no existe");

            return false;

        }

    }
  }

```

```

function validaPassword(contrasenia){ //si la contraseña se coincide con el usuario al recorrer el array devuelve true

    for(var i = 0; i < nombres.length; i++){

        for(var j = 0; j < passwords.length; j++){

            if(contrasenia == passwords[i] && login == nombres[i]){

                return true;

            }

        }

    }

    return false;

}

```

```

    existeLogin(login);

    if(validaPassword(contrasenia)){

        alert("Los datos son correctos");

    }

    if(!validaPassword(contrasenia)){

        alert("La contraseña no coincide con el usuario");

    }

} //FIN DEL ELSE GENERAL
</script>

```