



Nombre de la práctica: Empleo de programación del lado del cliente en sitio web.

Nombre del alumno: Magdalena Arana Álvarez.

Lugar y fecha: 03/12/2023.

## Índice

1.	Introducción	3
2.	Desarrollo de la práctica	3
3.	Conclusión	16
4.	Referencias	16

# 1. Introducción

La siguiente práctica muestra la programación del lado del cliente de un formulario de pago, para aplicar la programación del lado del cliente.

La programación del lado del cliente se usa para la integración de sitios web, en la práctica se usa HTML para dar estructura, CSS para dar estilo y Javascript para ejecutar interpretado e interpretar eventos.

La programación del lado del cliente permite validar datos antes de que sean enviados al servidor y así dar a conocer al usuario de manera inmediata un informe de error.

## 2. Desarrollo de la práctica

Descripción: Se modificará el sitio web de una empresa, que servirá para probar código JavaScript y HTML.

1. Crea un nuevo documento HTML con el formulario indicado.
2. Inicializa el repositorio con GIT.
3. Agrega un botón de PAGAR el cual simule la pasarela de pago, al dar clic en él deberás de validar con JavaScript y HTML5 la longitud del número de tarjeta a 16 dígitos sólo números, una fecha válida, código de seguridad a 3 dígitos, todos los datos obligatorios, en caso de que alguno no cumpla con el formato se mandará un mensaje de error, en caso de que todo sea correcto mandar un mensaje de "Pago exitoso".
4. Utiliza colores, fonts, tamaños a tu gusto.
5. Detecta cambios con GIT y actualiza el repositorio.

Se utiliza el lenguaje de programación Javascript y el lenguaje de marcado de hipertexto HTML para la creación del formulario de pago para probar el código.

### Pago con tarjeta.

### Dirección de la tarjeta

## *Todos los campos.*

No se escribe ningún dato y se oprime el botón pagar para visualizar las condiciones establecidas. Si no se cumplen estas características, este es el mensaje que se muestra al usuario para realizar la corrección, además el campo se mostrará en rojo.

-El campo **Número de tarjeta** es requerido, solo ingrese los 16 números de su tarjeta.  
-El campo **Día** es requerido, solo ingrese números.  
-El campo **Mes** es requerido, solo ingrese números.  
-El campo **Año** es requerido, solo ingrese números.  
-El campo **Código** es requerido, solo ingrese los 3 números del código de su tarjeta.  
-El campo **Nombre** es requerido, solo ingrese letras.  
-El campo **Apellido** es requerido, solo ingrese letras.  
-El campo **Domicilio** es requerido, solo ingrese letras.  
-El campo **Tipo de domicilio** es requerido, solo escriba apartamento, suite o edificio.

### Pago con tarjeta.

  

### Dirección de la tarjeta

## Campo número de tarjeta.

El campo número de tarjeta es validado bajo las siguientes condiciones.

```
function validarNumeroTarjeta (){
    if(txtNumero.value.length < 16 || txtNumero.value.length > 16){
        return false;
    }//Regresa falso si el largo del valor recibido es menor o mayor a 16
    if(txtNumero.value.length == 0){
        return false;
    }// Regresa falso si el largo del valor recibido es cero

    if (isNaN(txtNumero.value)) {
        return false;
    } // Regresa falso si no es un numero

    if (parseFloat(txtNumero.value) <= 0) {
        return false;
    }// Regresa falso si es decimal

    return true;
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Número de tarjeta** es requerido, solo ingrese los 16 números de su tarjeta.

### Pago con tarjeta.

### Dirección de la tarjeta

## Campo día.

```
function validarDia (){  
  
    if(txtDia.value.length < 2 || txtDia.value.length > 2 ){  
        return false;  
    }//Regresa falso si el largo del valor recibido es menor o mayor a 2  
  
    if(txtDia.value.length == 0){  
        return false;  
    }// Regresa falso si el largo del valor recibido es cero  
    if (isNaN(txtDia.value)) {  
        return false;  
    } // Regresa falso si no es un numero  
  
    if (parseFloat(txtDia.value) <= 0) {  
        return false;  
    } // Regresa falso si es decimal  
    return true;  
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Día** es requerido,solo ingrese números.

### Pago con tarjeta.

1234567890123456

123

11

1980

123

### Dirección de la tarjeta

Juan

Perez

guanajuato

apartamento

Pagar

## Campo mes.

```
function validarMes (){  
    if(txtmes.value.length < 2 || txtmes.value.length > 2){  
        return false;  
    }//Regresa falso si el largo del valor recibido es menor o mayor a 2  
    if(txtmes.value.length == 0){  
        return false;  
    }// Regresa falso si el largo del valor recibido es cero  
  
    if (isNaN(txtmes.value)) {  
        return false;  
    } // Regresa falso si no es un numero  
  
    if (parseFloat(txtmes.value) <= 0) {  
        return false;  
    }// Regresa falso si es decimal  
    return true;  
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Mes** es requerido,solo ingrese números.

### Pago con tarjeta.

1234567890123456

11

qqqqq

1980

123

### Dirección de la tarjeta

Juan

Perez

guanajuato

apartamento

Pagar



## Campo año.

```
function validarAnio (){  
    if(txtanio.value.length < 4 || txtanio.value.length > 4){  
        return false;  
    }//Regresa falso si el largo del valor recibido es menor o mayor a 4  
  
    if(txtanio.value.length == 0){  
        return false;  
    }// Regresa falso si el largo del valor recibido es cero  
    if (isNaN(txtanio.value)) {  
        return false;  
    }// Regresa falso si no es un numero  
  
    if (parseFloat(txtanio.value) <= 0) {  
        return false;  
    }// Regresa falso si es decimal  
    return true;  
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Año** es requerido,solo ingrese números.

### Pago con tarjeta.

1234567890123456

11

12

19803

123

### Dirección de la tarjeta

Juan

Perez

guanajuato

apartamento

Pagar

## Campo código.

```
function validarCodigo (){  
  
    if(txtNumber.value.length < 3 || txtNumber.value.length > 3){  
        return false;  
    }//Regresa falso si el largo del valor recibido es menor o mayor a 3  
    if(txtNumber.value.length == 0){  
        return false;  
    }// Regresa falso si el largo del valor recibido es cero  
  
    if (isNaN(txtNumber.value)) {  
        return false;  
    }// Regresa falso si no es un numero  
  
    if (parseFloat(txtNumber.value) <= 0) {  
        return false;  
    }// Regresa falso si es decimal  
  
    return true;  
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Código** es requerido, solo ingrese los 3 números del código de su tarjeta.

### Pago con tarjeta.

1234567890123456

11

12

1980

1234

### Dirección de la tarjeta

Juan

Perez

guanajuato

apartamento

Pagar

## Campo nombre.

```
function validarNombre(){  
  
    let regNombre = /^[a-zA-Z]+$/; //Expresión regular solo acepta letras  
    let regNombre1 = /[`!@#$%^&*()_+\-=\[\]{};':"\\|,.<>\/?~]/; //Expresión regular no  
    acepta caracteres especiales.  
  
    if (!(regNombre.test(txtNombre.value.trim())) || txtNombre.value.length < 3 ||  
    txtNombre.value.length == null || txtNombre.value.length == "" ||  
    regNombre1.test(txtNombre.value.trim())) {  
  
        return false;  
    } // Se usa función trim para no aceptar espacios, regresará falso si el largo del valor es  
    menor a 3 o nullo, o un espacio  
    return true;  
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

The screenshot shows a payment form titled "Pago con tarjeta." on a dark blue background. At the top, a pink error message box states: "-El campo **Nombre** es requerido, solo ingrese letras." Below the title, there are input fields for a card number (1234567890123456), an expiration date (11/12/1980), and a CVV (123). The section "Dirección de la tarjeta" includes input fields for the name (Juan1111 and Perez), the address (guanajuato), and the apartment (apartamento). A blue "Pagar" button is at the bottom.

## Campo apellido.

```
function validarApellido (){
    let regApellido = /^[a-zA-Z]+$/; ]+$/; //Expresión regular solo acepta letras
    let regApellido1 = /[`!@#$%^&*()_+\\-=\\[\\\\{};':"\\|,.<>\\/?~]/; ]+$/; //Expresión regular no
    acepta caracteres especiales.

    if (!(regApellido.test(txtApellido.value.trim())) || txtApellido.value.length < 3 ||
    txtApellido.value.length == null || txtApellido.value.length == "" ||
    regApellido1.test(txtApellido.value.trim())) {

        return false;
    } // Se usa función trim para no aceptar espacios, regresará falso si el largo del valor
    es menor a 3 o nullo, o un espacio

    return true;
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Apellido** es requerido, solo ingrese letras.

### Pago con tarjeta.

1234567890123456

11

12

1980

123

### Dirección de la tarjeta

Juan

Perez0097

guajuato

apartamento

Pagar

## Campo domicilio.

```
function validarDomicilio (){
    if(txtDomicilio.value.length < 3 ){
        return false;
    }//Regresa falso si el largo del valor es menor a 3
    if(txtDomicilio.value.length == 0){
        return false;
    }//Regresa falso si el largo del valor es cero

    return true;
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Domicilio** es requerido, ingrese letras y numeros.

### Pago con tarjeta.

1234567890123456

11

12

1980

123

### Dirección de la tarjeta

Juan

Perez

gu

apartamento

Pagar

## Campo tipo.

```
function validarLugar (){
    let regLugar = /^[a-zA-Z]+$; ]+$; //Expresión regular solo acepta letras
    let regLugar1 = /[`!@#$%^&*()_+\\-=\\[\\\\{};':"\\|,.<>\\/?~]/; ]+$; //Expresión regular no
    acepta caracteres especiales.

    if (!(regLugar.test(txtLugar.value.trim())) || txtLugar.value.length < 3 ||
    txtLugar.value.length == null || txtLugar.value.length == "" ||
    regLugar1.test(txtNombre.value.trim())) {

        return false;
    } // Se usa función trim para no aceptar espacios, regresará falso si el largo del valor es
    menor a 3 o nullo, o un espacio

    return true;
}
```

Por lo que el formulario arroja el mensaje de error si no se cumple alguna validación.

-El campo **Tipo de domicilio** es requerido, solo escriba  
apartamento, suite o edificio.

### Pago con tarjeta.

1234567890123456

11

12

1980

123

### Dirección de la tarjeta

Juan

Perez

Guanajuato 12 Col. América

apartamento2222

Pagar

## Botón Pago Exitoso.

Si todos los campos son llenados correctamente el botón indicará “Pago Exitoso.”

**Pago con tarjeta.**

1234567890123456

11 12 1980

123

**Dirección de la tarjeta**

Juan Perez

Guanajuato 12 Col. América

apartamento

Pago Exitoso...

Se inicializa Repositorio

```

MINGW64:/c/Users/PC/arana_magdalena_Empleodeprogramacióndelladod...
Reinitialized existing Git repository in C:/Users/PC/arana_magdalena_Empleodepro
gramacióndelladodelclienteensitioweb/.git/

PC@CO-2101-MT MINGW64 ~/arana_magdalena_Empleodeprogramacióndelladodelclienteens
itioweb (main)
$ git status
On branch main
nothing to commit, working tree clean

PC@CO-2101-MT MINGW64 ~/arana_magdalena_Empleodeprogramacióndelladodelclienteens
itioweb (main)
$ git checkout develop
  
```

### **3. Conclusión**

La programación del lado del cliente es la más utilizada actualmente ya que los usuarios pueden ejecutarla desde sus dispositivos, el usuario puede realizar entrada de información para que posteriormente su información pueda ser validada y verificada para los fines establecidos del sitio web.

### **4. Referencias**

Gauchat, J. (2017). El gran libro de HTML5, CSS3 y Javascript 3ª edición. Págs. 241-356. [https://www.academia.edu/40128302/El\\_gran\\_libro\\_de\\_HTML5\\_CSS3\\_y\\_JavaScript](https://www.academia.edu/40128302/El_gran_libro_de_HTML5_CSS3_y_JavaScript)