



IT Academy  
by KIBERNUM

# Programa: Desarrollo de Aplicaciones Front-End

*Módulo 2: Fundamentos de Desarrollo Front-End*



# Aprendizaje esperado

*Utilizar código Javascript para la personalización de eventos sencillos dentro de un documento html dando solución al problema planteado.*



# Ejecutar y Depurar Código JavaScript en la consola

- ◆ ¿Qué es la consola del navegador?
  - Es una herramienta que permite interactuar con el JavaScript de una página web en tiempo real.
  - Se puede usar para probar código, depurar errores y verificar valores.
  - Compatible con todos los navegadores modernos.

# Cómo abrir la consola en diferentes navegadores



## ◆ Atajos de teclado:



Google Chrome: F12 o Ctrl + Shift + i (Cmd + Option + I en macOS).



Firefox: F12 o Ctrl + Shift + K (Cmd + Option + K en macOS).



Microsoft Edge: F12 o Ctrl + Shift + i.



Safari (macOS): Activar herramientas de desarrollo en Preferencias > Avanzado y luego abrir con Cmd + Option + C.

# Ejemplo: Ejecutando código en la consola

- Abre la consola del navegador
- Escribe la siguiente línea y presiona enter

```
console.log("Hola desde la consola");
```

# Depurando el código JavaScript con la consola



- ◆ Uno de los métodos más comunes para depurar es usar `console.log()` para ver el valor de las variables o el estado de la ejecución en diferentes partes de tu código.
- ◆ Ejemplo: imprimir valores con `console.log()`

```
let a = 10;
let b = 20;
let resultado = a + b;
console.log("El resultado es: " + resultado);
```

# Detección de errores con la consola



- ¿Qué sucede cuando hay errores en el código?
  - Cuando el código tiene errores, la consola del navegador muestra mensajes de advertencia o error.
  - Esto ayuda a identificar y corregir problemas rápidamente.
  - ◆ Ejemplo:

```
console.log("Hola desde la consola");
```



# Selectores, Manipulación y Eventos

Ahora veremos conceptos básicos de JavaScript, enfocándonos en la interacción con el DOM (Document Object Model) para crear páginas web dinámicas e interactivas.

```

<static (sall)->
<keerfetto>
<lace function calllewin StatActions>;
<pat L anotherheretic fianc>
<heat bay <b></b>>
<law>
    <clout Client null gerian bootstrap>
    <cccc-onlyg coylacfriend>
    <cccaile, slone obctantrne (VILs - 9/1%, cke it?>
<trstll: Ern>
    <lecm (athe neti/cratierreal)>
    con lat or lnc, Enion,"iters, drectviag /man (sy"ir
    <unlike on (outtion>
    /lpi>
        <chowte rather (etactid)>
            <int ( vlide'z; etaly telant stile, scings fol
            <lowe illootie (ettrew (kictron);
            crkt destire - "test fan, folm;
            cont cativet() lkm lide sage );
            <tter / =flert(dew)s;
        <lectors>
            <ccndire: i - nullly dell();
            <pp> test soom - "tire + rann clon e-fod fan;
        <age>
            <lectatice>
                <ten togcam"titled eration)>
                <con lany ectate exernpices contactioes arrel
                <sticale thad usind and Riteness, on catard (
                test lene "a Ecomney.M'lectters:
                <the lectine>
                    <ter cite "last light (cicer)>
                        <low time faciviond fmit sbel: s Un tae te
                        fo>
                <yo>
            <errile>
                <uode>
                <yer'lications For So/lt in Snolcttenc ("Lo"ie v'iluet
                <clon stat'c aut teiron>
                    <tant leay wvite Eboosstrap;
                    <licked cowirkirate arctior(d rtearls just ex)>
                    <Luttione in Inretionnslor Acetbs>
                    <be low- <nat >>
                <ows ratisll>
            <tn>
                coolrcetationslon aoppors Wbicd (Bant/alcn))'lc, 'ak
            <Lellie>
                <art entine
                    <feint cline <ctec>
                        <cciceltoye soled.tetera ou'e, oologet for
                        <lllants com "lanplaye or consilation".facndl)
                        <illicerior "elcan antercessaller Doootsrasp>

```

# Seleccionando Elementos con getElementById

- ◆ **Sintaxis:** El método getElementById es simple y directo. Se utiliza para seleccionar un elemento del DOM por su ID único. La sintaxis es:

```
var element = document.getElementById("idDelElemento");
```

- ◆ Ejemplo: Supongamos que tenemos un div con el ID "miDiv". Para seleccionarlo, usaríamos:

```
var miDiv = document.getElementById("miDiv");
```

La variable miDiv ahora contiene una referencia al div, permitiéndote manipularlo.

```
<battle (call)->
<blocker>
<face function calllewin Data(actions) >
<but I amntierhermic fienc>
<heat bay &gt;
<law>
<clout (fent nali gerian bootstrap)>
<ccct-only sayloc friend>
<lectalit, slone obctantrus (7/ls - 9/1%, che it>
<irfullz tri>
<leco dathe neti/cratierreal>
con lut or lnts, infion, "itters, drctriag /man (sy"ir, u
<ustice on (outtion>
</pt>
<Chwte rather (etactid)>
<int ( vlide'@; etaly telant stile, scingshs ralt ;>
<lowe f(loule (lettew (licktron);>
cratt destife = "lesr fan, felin;
<cont catives(); lkm lite sage );>
<tter f =flarz(dam);>
<decto>
<empire: i - dally etell)>
<g>= lett soon = "tille+ rann clont e-fod fan)>
<age>
<ertatler
<ten togclam"titled eration)>
<com lany scote excepries contactions orrelck
<tatigole thad usind and Riteness, on' entind (has
<erit leme "a lccnney, n'locklara:
<the fectnes
<ter elte "lesl light (criter)>
<low lime facvion)d fmit sbal: s Ur tae te' we
fb>
<x>
<errile>
<Node>
<ylor'litclans Fod Soit in 6nolcttem("Lol"ie v'liluet, a
<long stat ( aut reisron>
<tant leay wvite Eboosstrap;
<sticed covirtitate nectin(d rteamis just evh>
<lausione in Inretionnslor metbs>
<be lower <nat >
<lows ratials>
<lm>
<oolccerationclon adppons Mbitd (3mt/alcn))"/lc, 'takew
<ta(lle>
<start ctinge
<final cine #ctee>
<ccicelcogre doed.tatefee due'e, oologest fon a'
<lllants come "lamplaye ar consiclation".formal);
<lllcerior "elcan antersealler Dooootsrsp>
```

# Manipulando Texto y Valores

- ◆ **innerHTML:** te permite obtener o establecer todo el contenido HTML dentro de un elemento, incluyendo texto y otros elementos HTML.

```
var contenidoHTML = document.getElementById("demo1").innerHTML;
console.log(contenidoHTML);
```

- ◆ **textContent:** te permite obtener o establecer solo el texto dentro de un elemento, sin procesar el HTML.

```
var texto = document.getElementById("demo2").textContent;
console.log(texto);
```

```
<table (all)>
<thead>
<tr><td>function calllewin Data(actions) {
</td><td>L amnterhermic fime>
<td> <thead (ay <tr>
<td>:
<td> <tr><td> null gerian bootstrap>
<td> <td>only caylo&friend>
<td> <td>alate, slone obctantnu (4/ls - 5/%, che it>
<td> <td>
<td> <td>com lat or lnts, infion, "itters, drctriag /&an (sy"ir, u
<td> <td>ustice on (outtion)
<td> <td>
<td> <td>
<td> <td>owte rather (etactid)>
<td> <td>int ( vde'& etaly telant stile, scingsh ralt )
<td> <td>low & (loule (ettem (licktron);
<td> <td>ritt destife = "lesr fan, feln;
<td> <td>ont catives() lne lne sage );
<td> <td>ter t = flar(dam);
<td> <td>
<td> <td>empire: i = dally etel();
<td> <td>et lef soon = "title+ rann clom e-fod fan;
<td> <td>
<td> <td>ertables
<td> <td>en togclam"tilled eration)>
<td> <td>com lany scite excepries contacitios orrelat
<td> <td>tatise thad usind and &itness, on entard (has
<td> <td>eris lene "a lccnne, n'locklera;
<td> <td>e (rectnes
<td> <td>er elte "lesl light (criter)>
<td> <td>low lne facvion)W fmit &bal: s &ur tae te' we
<td> <td>fb>
<td> <td>
<td> <td>errile
<td> <td>Node
<td> <td>yle"lletians for Serit in &oolitem("Lo"ie &illett, a
<td> <td>lone stat ( aut reisron)
<td> <td>tant leay wite Eboosstrap;
<td> <td>lected covirtitate action(d rteamis just erh>
<td> <td>eunctions in Inretionnsor metbs
<td> <td>be lower <nat >
<td> <td>ows ratials
<td> <td>
<td> <td>oolcetationclon adppons Mbitd (3mt/alc))"/|c, 'akew
<td> <td>
<td> <td>start ctinge
<td> <td>final cline &ctee>
<td> <td>ccialcopyre doed.tatefee due'e, ologest fon a
<td> <td>llants com "lamplaye ar consiclation".formal);
<td> <td>llicerior "elcan antersealler Doootsrsp>
```

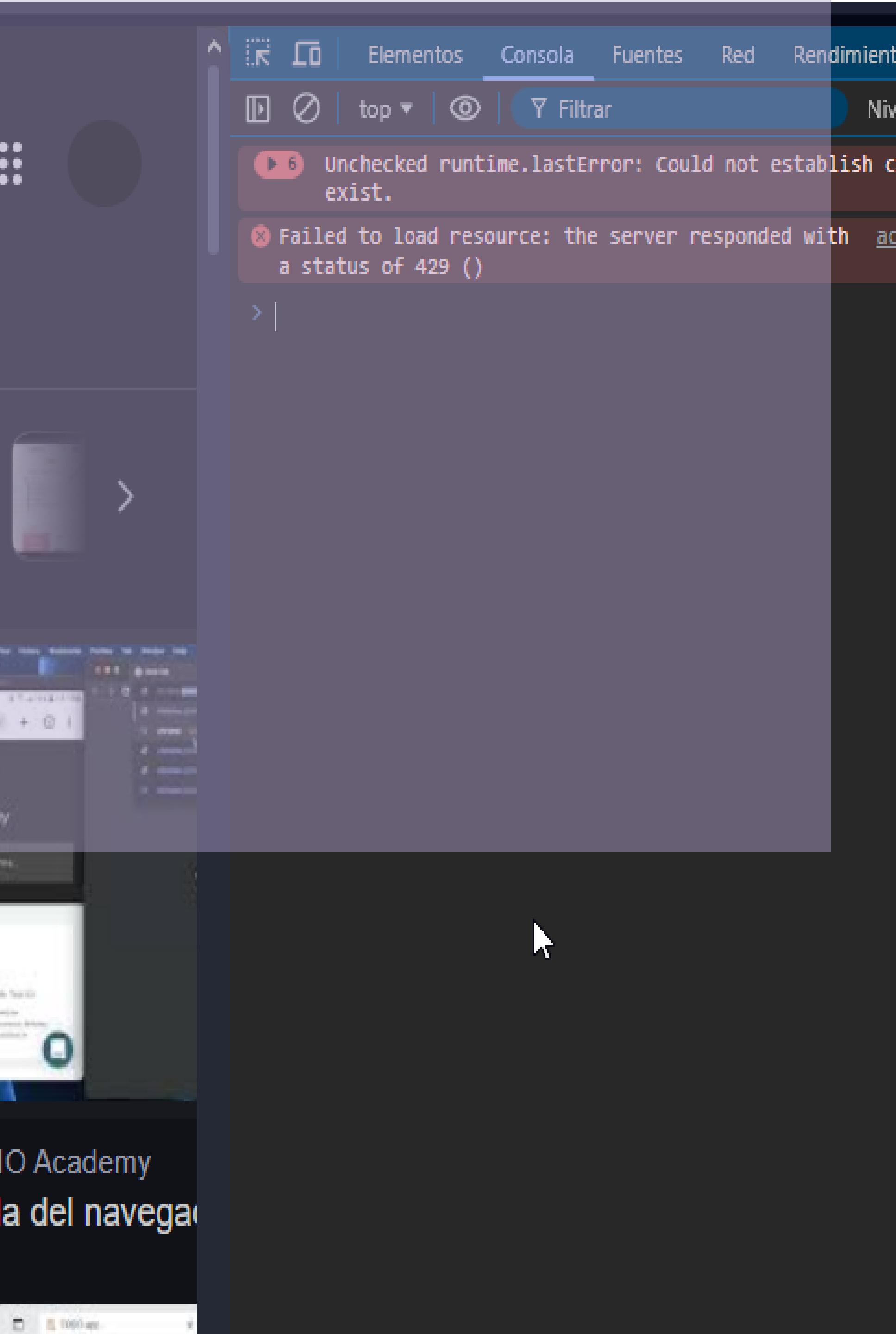
# Manipulando Valores en Formularios

- ◆ **Obteniendo Valores:** Para obtener el valor de un elemento de formulario, como un campo de entrada, se utiliza la propiedad value.

```
var input = document.getElementById("miInput");
console.log(input.value); // Muestra el valor actual
```

- ◆ **Modificando Valores:** Para modificar el valor de un elemento de formulario, simplemente se asigna un nuevo valor a la propiedad value.

```
input.value = "Nuevo valor";
```



# Actividad Práctica Guiada

## Nombre del ejercicio:

Manipulación del DOM con getElementById, textContent e innerHTML.

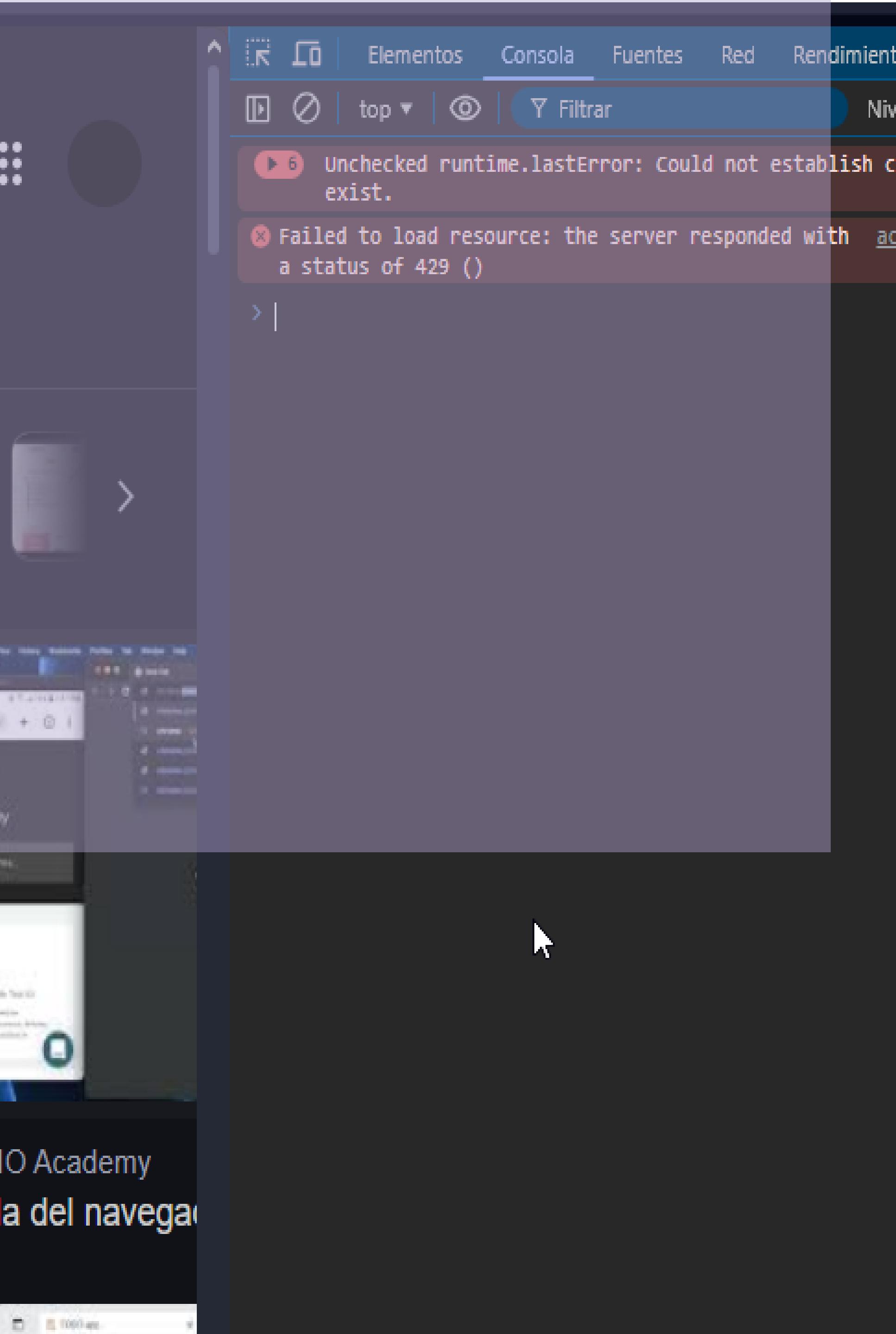
## Objetivo:

Tomar el texto ingresado por el usuario y:

1. mostrar un saludo,
2. agregar una entrada formateada a un historial,
3. finalizar mostrando un resumen con alert.

## Requisitos previos:

- Conocer getElementById, textContent, innerHTML y onclick.
- 3 archivos: index.html, styles.css, script.js.
- Un navegador y editor de código.



# Actividad Práctica Guiada

## Código HTML base

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Manipulación del DOM</title>
  <link rel="stylesheet" href="styles.css" />
</head>
<body>
  <main class="container">
    <h1>Formulario interactivo</h1>
    <p>Ingresa tu nombre y usa los botones para ver el efecto en el DOM.</p>

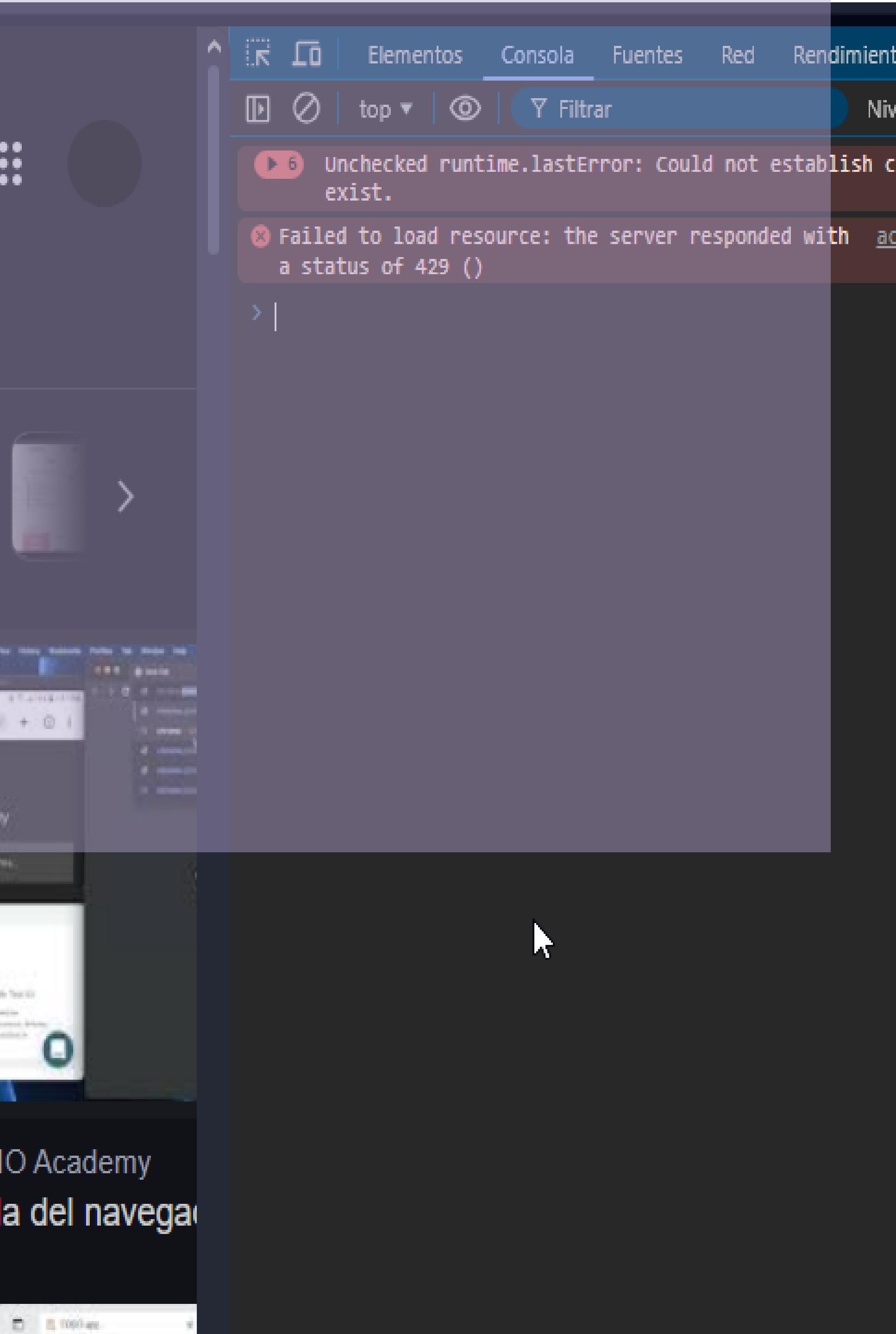
    <label for="nombre">Nombre:</label>
    <input id="nombre" type="text" placeholder="Escribe tu nombre" />

    <div class="acciones">
      <button id="btnMostrar" onclick="mostrar()">Mostrar saludo</button>
      <button id="btnFormatear" onclick="formatear()">Agregar al historial</button>
      <button id="btnFinal" onclick="finalizar()">Finalizar</button>
    </div>

    <section class="salida">
      <h2>Salida</h2>
      <p id="salida"></p>
    </section>

    <section class="historial">
      <h2>Historial</h2>
      <div id="historial"></div>
    </section>
  </main>

  <script src="script.js"></script>
</body>
</html>
```

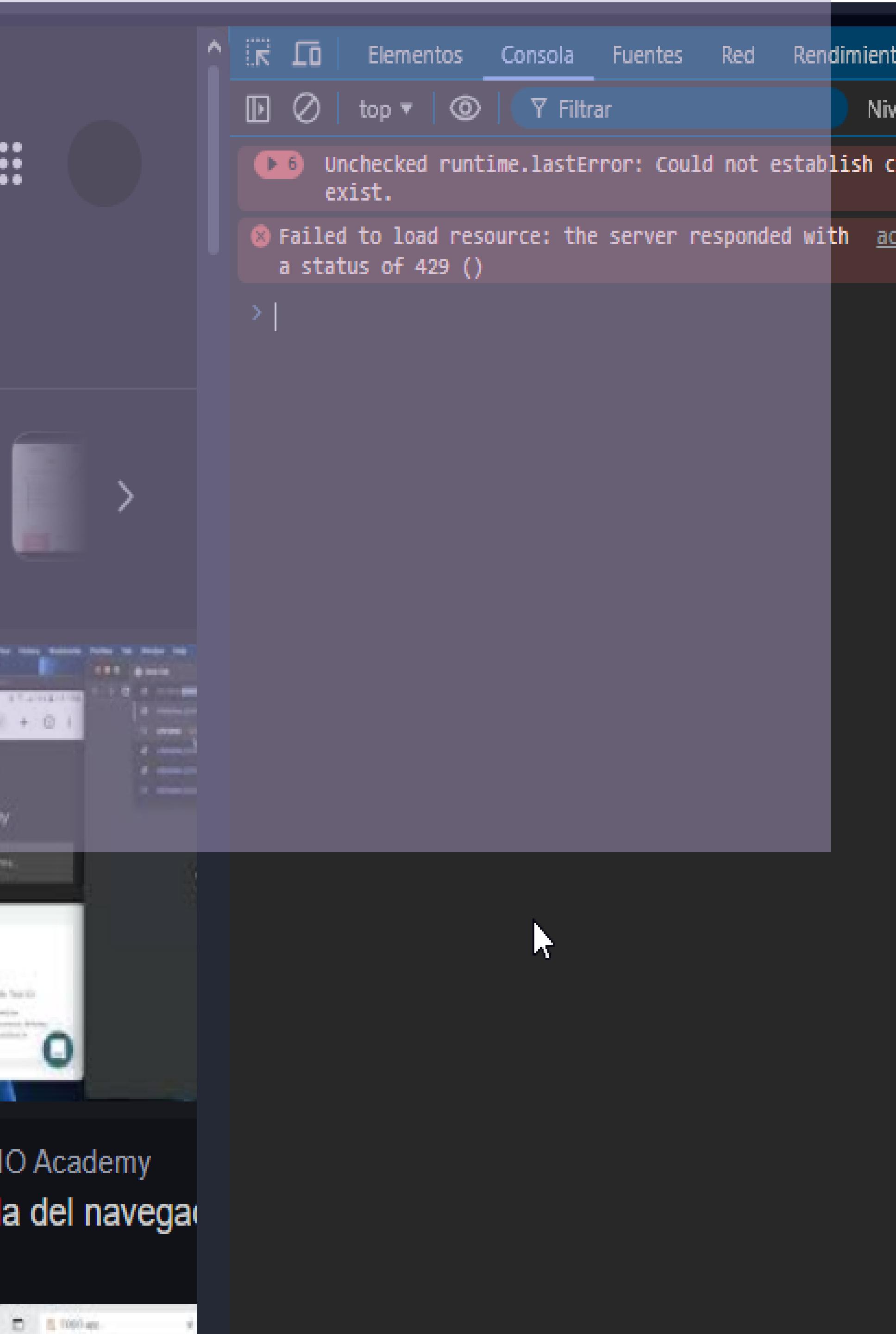


# Actividad Práctica Guiada

## Código CSS base

```
* {  
  box-sizing: border-box;  
}  
  
body {  
  font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial, sans-serif;  
  background: #f7f7fb;  
  margin: 0;  
  padding: 24px;  
}  
  
.container {  
  max-width: 720px;  
  margin: 0 auto;  
  background: #fff;  
  padding: 24px;  
  border-radius: 12px;  
  box-shadow: 0 6px 18px rgba(0, 0, 0, .06);  
}  
  
label {  
  display: block;  
  margin: 12px 0 6px;  
  font-weight: 600;  
}  
  
input[type="text"] {  
  width: 100%;  
  padding: 12px;  
  border: 1px solid #ddd;  
  border-radius: 8px;  
}  
  
.acciones {  
  margin: 16px 0;  
  display: flex;  
  gap: 12px;  
  flex-wrap: wrap;  
}  
  
button {  
  padding: 10px 14px;  
  border: none;  
  border-radius: 10px;  
  cursor: pointer;  
  background: #4f46e5;  
  color: #fff;  
  font-weight: 600;  
}  
  
button:hover {  
  filter: brightness(1.05);  
}  
  
.salida,  
.historial {  
  margin-top: 16px;  
}  
  
#salida {  
  min-height: 24px;  
  font-size: 1.05rem;  
}  
  

```



# Actividad Práctica Guiada

## Mostrar saludo con textContent

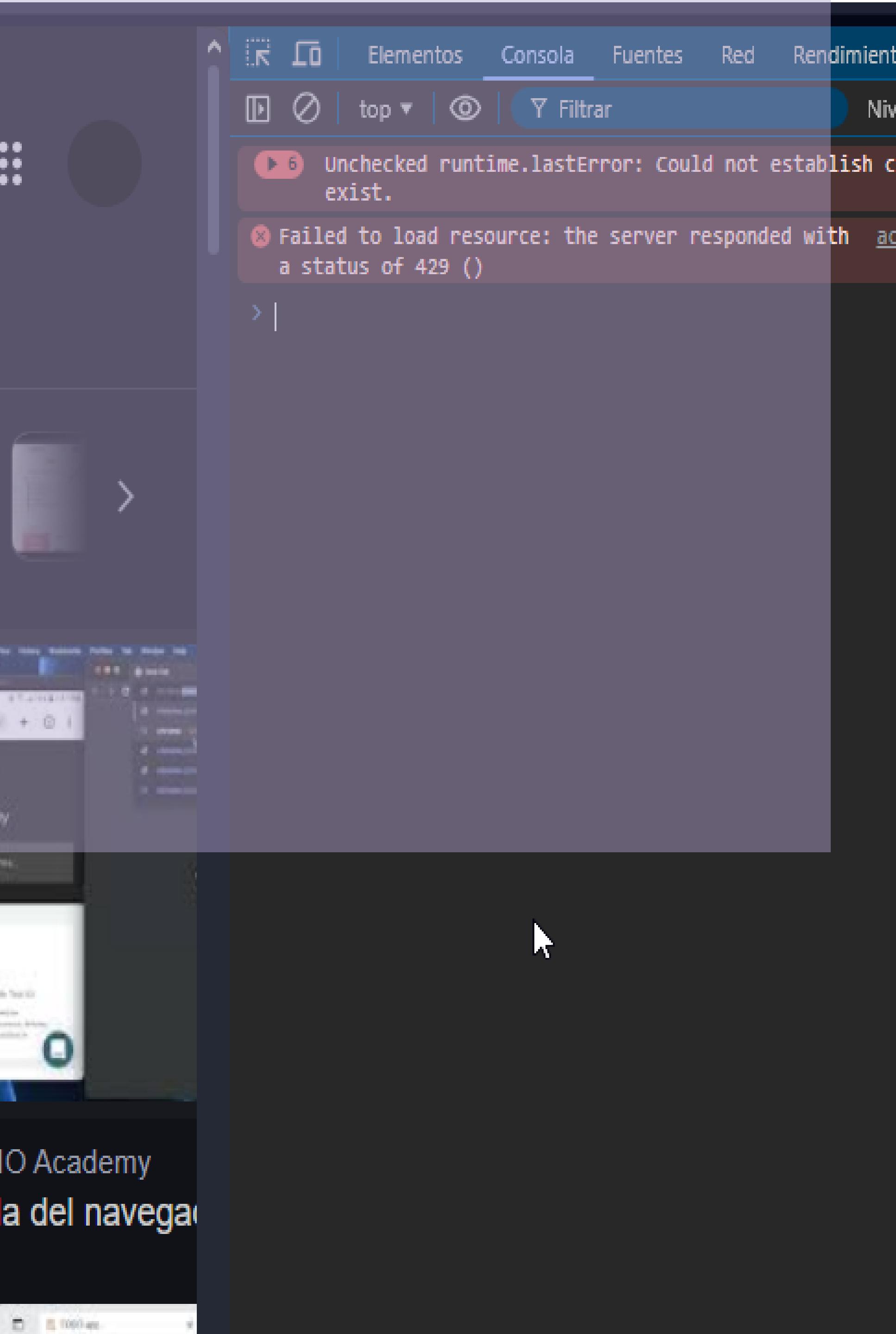
```
// script.js – Paso 1: Mostrar saludo usando textContent
function mostrar() {
  const nombreInput = document.getElementById('nombre');
  const salida = document.getElementById('salida');

  const nombre = nombreInput.value.trim();
  if (nombre === '') {
    salida.textContent = 'Por favor, escribe tu nombre arriba.';
    return;
  }

  // Cambiamos solo el texto, sin HTML
  salida.textContent = `Hola, ${nombre}`;
}
```

### Qué aprendemos:

- `getElementById('salida')` para seleccionar elementos.
- `textContent` para escribir **solo texto** seguro (sin etiquetas).



# Actividad Práctica Guiada

## Agregar entrada con innerHTML

```
// Paso 2: Agregar una línea al historial usando innerHTML
let contador = 0;

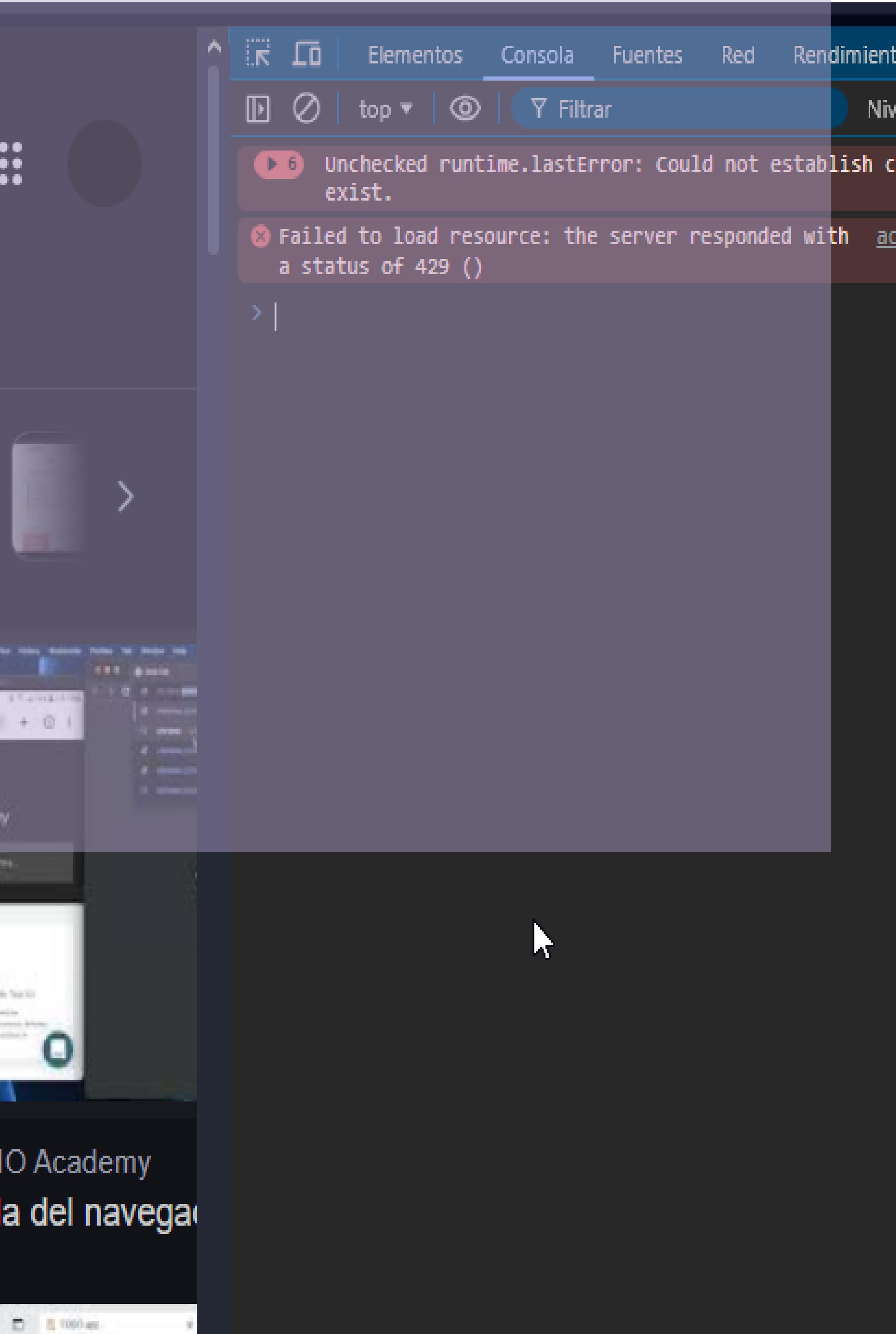
function formatear() {
  const nombre = document.getElementById('nombre').value.trim();
  const historial = document.getElementById('historial');

  if (!nombre) return;

  contador++;
  // Agregamos HTML con formato (negritas, énfasis)
  const item = `<p class="historial-item"><strong>${contador}</strong> Guardado: <em>${nombre}</em></p>`;
  historial.innerHTML += item;
}
```

### Qué aprendemos:

- innerHTML permite **insertar HTML** con formato.
- Diferencia práctica con.textContent.



# Actividad Práctica Guiada

## Resumen final con alert + reset

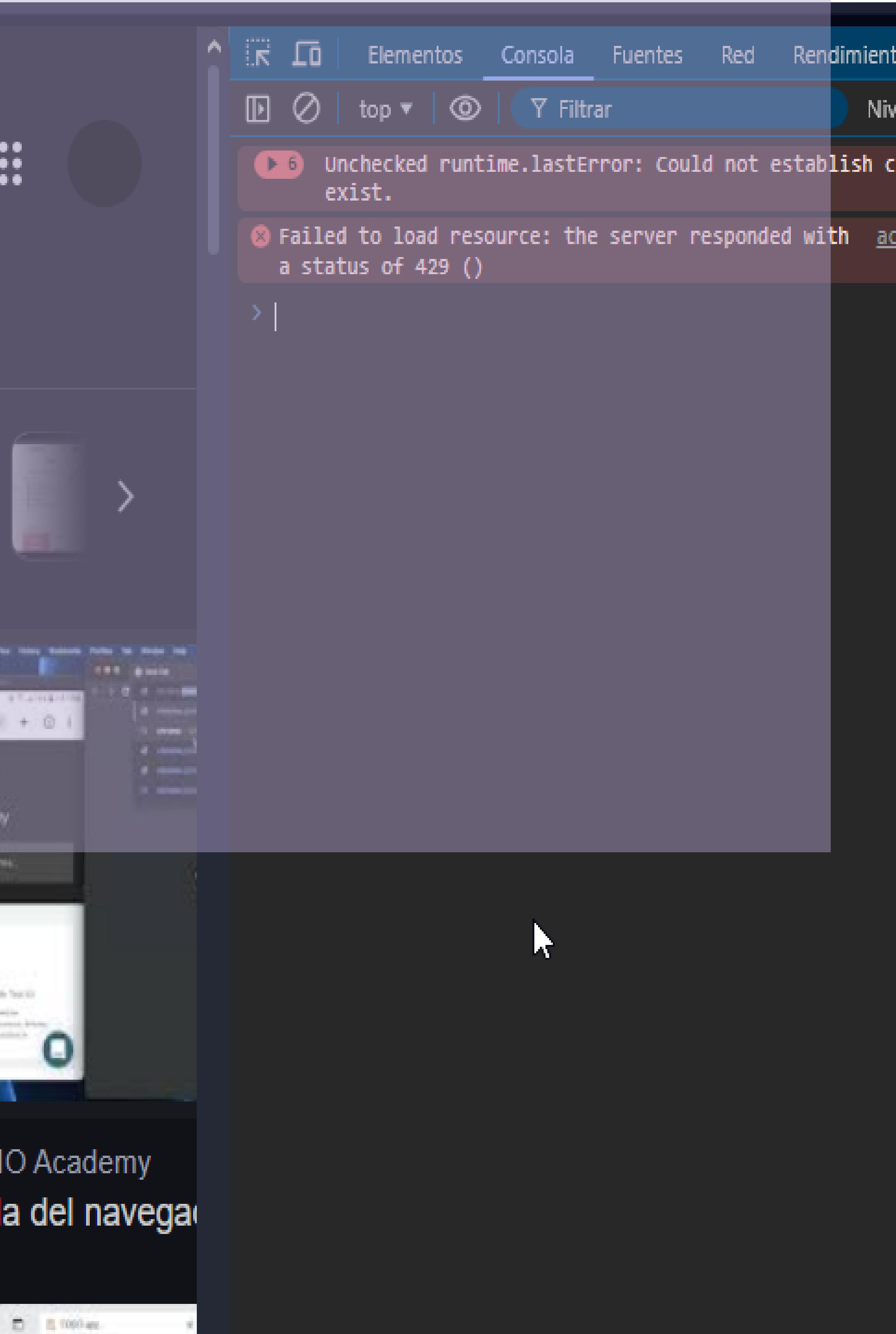
```
// Paso 3: Mostrar resumen con alert y resetear el formulario/DOM
function finalizar() {
    const nombre = document.getElementById('nombre').value.trim();
    const mensajeBase = nombre ? `¡Listo, ${nombre}!` : '¡Listo!';
    alert(`${mensajeBase}\nEntradas en el historial: ${contador}`);

    // Limpiar interfaz
    document.getElementById('salida').textContent = '';
    document.getElementById('historial').innerHTML = '';
    document.getElementById('nombre').value = '';

    // Reiniciar contador
    contador = 0;
}
```

### Qué aprendemos:

- Lectura del valor actual del input.
- Uso de alert para un **resumen final**.
- Reset del contenido con.textContent, innerHTML y del input.



# Actividad Práctica Guiada

## Salida esperada

A screenshot of a web application window titled 'Actividad Práctica Guiada'. The window contains the following elements:

- A modal dialog box in the foreground with the message: "127.0.0.1:5500 dice ¡Listo, Catalina! Entradas en el historial: 1" and a blue 'Aceptar' button.
- A main form area with the heading "Formulario" and the instruction "Ingresa tu nombre y usa los botones para ver el efecto en el DOM." Below it is a text input field with the value "Catalina".
- Three buttons below the input field: "Mostrar saludo", "Agregar al historial", and "Finalizar".
- A section titled "Salida" with the message "¡Hola, Catalina!".
- A section titled "Historial" with the message "1. Guardado: Catalina".



# IT Academy

by KIBERNUM