



INTRODUCCION A LA PROGRAMACION 2025-3

Retos 1 al 8

Luis Llumiquinga
30 de noviembre de 2025

Ing. Santiago Mosquera
Movilis

Resumen

El presente informe analiza los aprendizajes, soluciones y procesos de resolución aplicados en los retos 1 al 8 del curso “*Krakedev: Primeros Pasos*”. Cada reto se relaciona con fundamentos esenciales del desarrollo web, incluyendo la creación de páginas HTML, integración de hipervínculos, uso de imágenes, interacción mediante alertas, depuración de errores, control de variables y manipulación de elementos visuales dinámicos. Este documento presenta los resultados obtenidos y reflexiona sobre las habilidades adquiridas, así como su contribución en la formación del estudiante en HTML y JavaScript.

Palabras clave: HTML, JavaScript, desarrollo web, depuración, lógica de programación.

Introducción

El curso “*Krakedev: Primeros Pasos*” ofrece una formación inicial en conceptos esenciales del desarrollo web, especialmente orientado a la comprensión y práctica de HTML y JavaScript. En este informe se detalla el trabajo realizado en los retos 1 al 8, abordando la construcción de páginas web, la integración de elementos funcionales y visuales, y el uso de estructuras lógicas para modificar el comportamiento del contenido. Cada reto permitió reforzar la comprensión de la sintaxis, identificar errores, aplicar técnicas de depuración y desarrollar soluciones prácticas.

Desarrollo

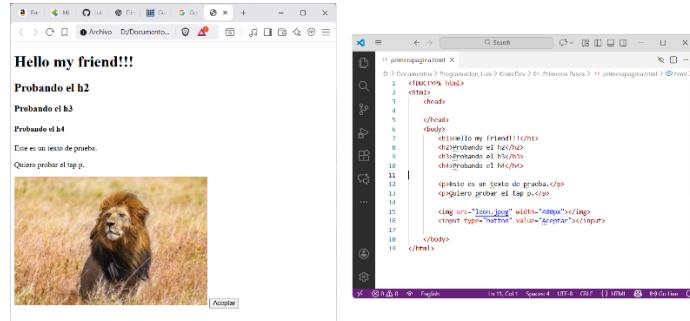
Reto 1: Crear y modificar páginas HTML

Este reto consistió en la creación de páginas HTML básicas y su modificación mediante formularios y estructuras aprendidas durante el curso. Se desarrollaron páginas dinámicas organizadas en secciones utilizando etiquetas fundamentales como `<header>`, `<nav>`, `<section>` y `<footer>`. También se experimentó con formularios simples mediante etiquetas `<form>`, `<input>`, `<label>` y `<button>`, lo que permitió comprender el flujo de datos entre usuario y página.

El aprendizaje principal fue comprender la estructura jerárquica del HTML y la importancia de organizar el contenido para garantizar accesibilidad y claridad. Asimismo, se reforzó el uso de atributos como `id`, `class` y `name`.

Figura 1

Creación de la primera pagina



Reto 2: Agregar hipervínculos

En este reto se integraron hipervínculos mediante la etiqueta `<a>`, conectando diferentes secciones y páginas del sitio web. Se configuraron enlaces internos y externos, además de experimentar con los atributos href, target y rel.

El aprendizaje clave consistió en mejorar la navegación dentro del sitio y comprender la estructura de los recursos web, así como el correcto uso de rutas absolutas y relativas.

Figura 2

Agregación de hipervínculos a la pagina web

LA AVIACION

Introducción

Se entiende por aviación al diseño, desarrollo, fabricación, producción, operación y utilización para fines privados o comerciales de aeronaves, especialmente las más pesadas que el aire

Dirección General de Aviación Civil

Aviación Civil

Aviación Militar

Las Fuerzas Armadas no tienen unidades de la aviación militar, bien a través de organizaciones independientes especializadas, como la fuerza aérea, o bien mediante servicios integrados en otras ramas no estrictamente aeronáuticas, como la aviación naval o la aviación agregada a las fuerzas terrestres.

[Aviación Civil](#)

[Aviación Militar](#)

```

<!DOCTYPE HTML>
<html>
<head>
<title>LA AVIACION</title>
</head>
<body>
  <h1>LA AVIACION</h1>
  <h2>Introducción</h2>
  <p>Se entiende por aviación al diseño, desarrollo, fabricación, producción, operación y utilización para fines privados o comerciales de aeronaves, especialmente las más pesadas que el aire</p>
  <img alt="Logo de la Dirección General de Aviación Civil (DAC)" data-bbox="280 595 310 625" />
  <img alt="Imagen de una avioneta" data-bbox="325 590 425 635" />
  <h2>Aviación Civil</h2>
  <img alt="Imagen de un avión comercial" data-bbox="355 645 445 665" />
  <h2>Aviación Militar</h2>
  <img alt="Imagen de un avión de combate" data-bbox="355 675 445 695" />
  <p>Las Fuerzas Armadas no tienen unidades de la aviación militar, bien a través de organizaciones independientes especializadas, como la fuerza aérea, o bien mediante servicios integrados en otras ramas no estrictamente aeronáuticas, como la aviación naval o la aviación agregada a las fuerzas terrestres.</p>
  <hr>
  <a href="https://www.aviaciuncivil.gob.ec/">Aviación Civil</a>
  <a href="https://www.faa.mil.ec/">Aviación Militar</a>
</body>
</html>

```

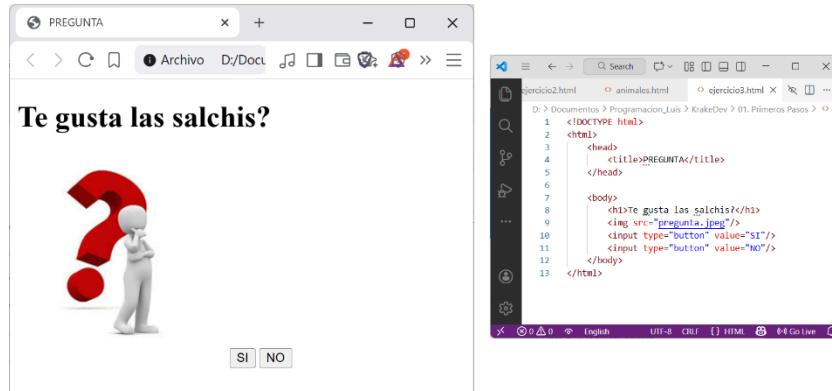
Reto 3: Agregar imágenes con

Se incluyeron imágenes en la página utilizando la etiqueta `` y atributos como `src`, `alt`, `width` y `height`. También se trabajó con rutas locales y URLs externas.

Este reto permitió entender la importancia del texto alternativo para accesibilidad, así como las implicaciones del peso y formato de las imágenes en el rendimiento de la página.

Figura 3

Uso de tag en html



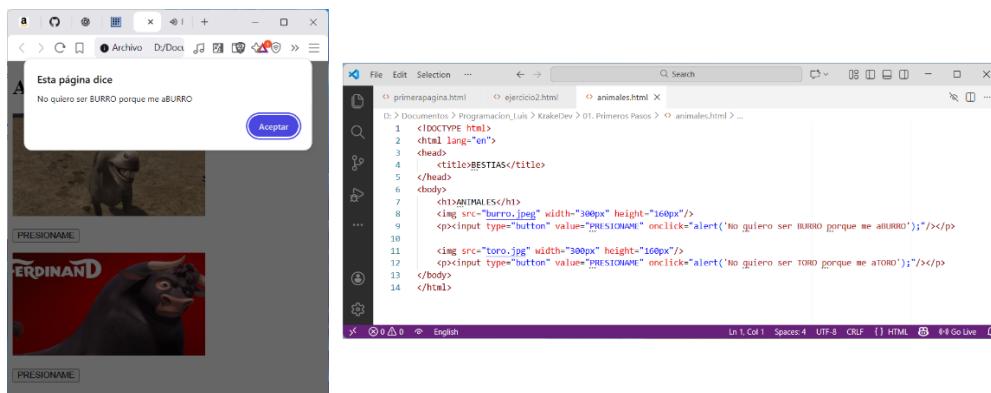
Reto 4: Uso de alert() para mensajes emergentes

En este reto se utilizó la función `alert()` de JavaScript para mostrar mensajes emergentes basados en acciones del usuario. Se integraron scripts en la página mediante la etiqueta `<script>`.

El aprendizaje significativo fue comprender el flujo de ejecución del JavaScript dentro de una página HTML y cómo brindar retroalimentación al usuario de forma simple.

Figura 4

Agregación de ventanas emergentes con alert



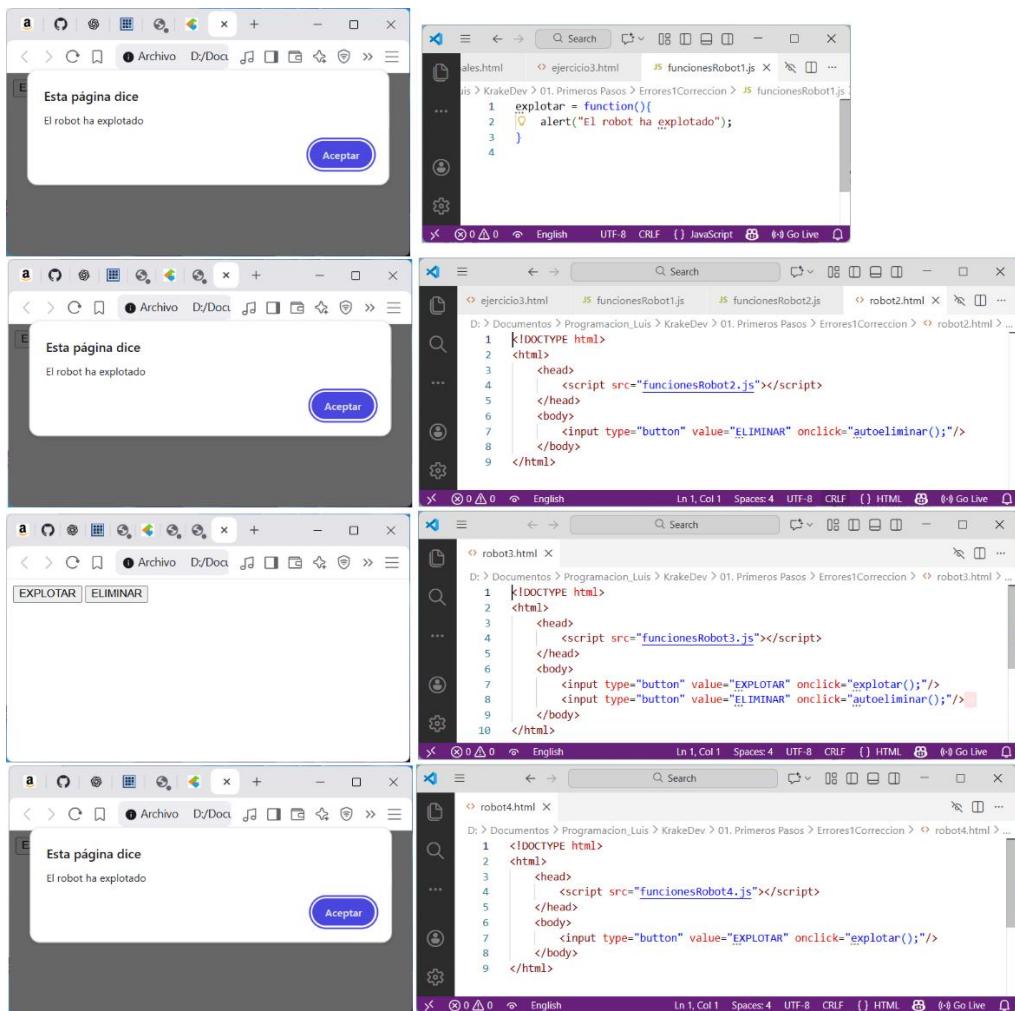
Reto 5: Corrección de errores

Este desafío abordó la identificación y corrección de errores de sintaxis tanto en HTML como en JavaScript. Se revisaron etiquetas mal cerradas, rutas incorrectas, nombres de variables mal escritos y errores en el uso de funciones.

El reto fortaleció las habilidades de lectura crítica del código, además de reforzar la relación entre estructura y funcionalidad. El proceso contribuyó a una mayor atención al detalle.

Figura 5

Corrección de errores



Reto 6: Evidencia de depuración (debug)

Aquí se aplicaron técnicas de depuración, utilizando principalmente la consola del navegador mediante `console.log()`, así como herramientas de inspección.

Se identificaron flujos de valores, ejecución de funciones y errores lógicos. El aprendizaje obtenido fue comprender la importancia del debug como parte esencial del desarrollo, permitiendo encontrar fallos que no son visibles a simple vista.

Figura 6

Evidencia de bug

The screenshot shows the Chrome DevTools debugger interface. The top bar has tabs for 'Rastrear Paquete', 'LuisJumiquinga/Python', 'Eliminar teclados adicionales', 'Guía-Normas-APA-7ma-e', 'BESTIAS', 'Modificar Variables', and 'variables.html'. Below the tabs, there are buttons for 'Archivo' (File), 'D:/Documentos/Programacion_Luis/KrakeDev/01.%20primeros%20Pasos/Capitulos12/variables.html', and a zoom control ('+10 | -10'). A status bar at the bottom says 'Paused in debugger' with icons for back, forward, and search.

The main area shows the DevTools navigation bar with 'Page', 'Workspace >', and 'variables.js X'. The 'Page' tab is selected. The 'Workspace' section shows a tree view with 'top' (containing 'file://'), 'D:/Documentos/Programacion_Luis/KrakeDev/01.%20primeros%20Pasos/Capitulos12/variables.html' (containing 'variables.js'), and 'variables.js'. The code editor on the right displays the following JavaScript code:

```
1 //Declarar la variable puntaje
2 let puntaje;
3 //Asigno 0 a puntaje
4 puntaje = 0;
5
6 cambiar = function(){
7   puntaje = 1000;
8 }
9
10 mostrar = function(){
11   alert("El puntaje es: "+puntaje);
12 }
13
14 incrementar = function(){
15   puntaje = puntaje + 1;
16 }
17
18 disminuir = function(){
19   puntaje = puntaje - 1;
20 }
21
```

The line 11 of the code is highlighted with a blue background, and the status bar indicates 'Line 11, Column 5'. The bottom right corner shows 'Coverage: n/a'.

The bottom panel contains a 'Scope' tab with 'Paused on breakpoint' checked, and a 'Watch' tab with a dropdown menu showing 'local', 'this', 'Window', 'Script', and 'Global'. The 'variables.js' entry in the dropdown is also highlighted.

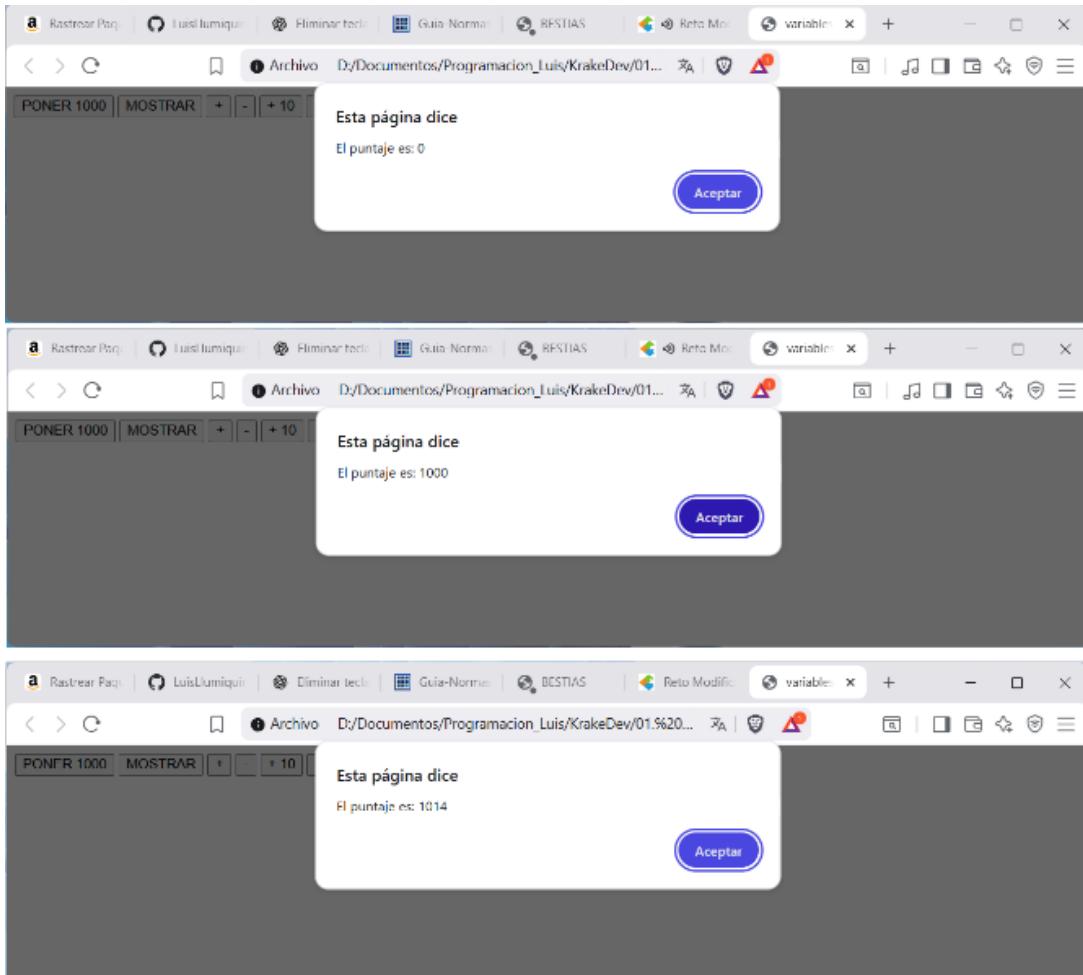
Reto 7: Modificación de una variable

Este reto consistió en declarar, modificar y mostrar el valor de una variable en JavaScript. Se analizaron los tipos de variables (let, var, const) y cómo cambian su comportamiento dentro del código.

El aprendizaje central fue la importancia del manejo correcto de variables para controlar el estado de una aplicación, así como la relevancia de los tipos y el ámbito (scope).

Figura 7

Modificación de variables



Reto 8: Encender una computadora mediante botones

Este reto integró HTML y JavaScript para cambiar imágenes según la interacción del usuario. Se crearon dos botones: uno para mostrar una imagen de una computadora encendida y otro para mostrar la computadora apagada. Las funciones JavaScript cambiaban dinámicamente el atributo src de la imagen.

El aprendizaje más relevante fue comprender cómo manipular elementos del DOM (Document Object Model) usando getElementById() y cómo generar interactividad básica en una página web.

Figura 8

Modificación de imágenes

The figure shows two code editor windows side-by-side.

interaccionPc.html:

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <script src="interaccionPc.js"></script>
5   </head>
6
7   <body>
8     
9     <input type="button" value="ENCENDER" onclick="encender();"/>
10    <input type="button" value="APAGAR" onclick="apagar();"/>
11  </body>
12 </html>

```

interaccionPc.js:

```

1 encender=function(){
2   let cmpOn;
3   cmpOn=document.getElementById("computador");
4   cmpOn.src="compuPrendida.PNG";
5 }
6
7 apagar=function(){
8   let cmpOff;
9   cmpOff=document.getElementById("computador");
10  cmpOff.src="compuApagada.PNG";
11 }

```

Conclusiones

Los retos del curso “*Krakedev: Primeros Pasos*” permitieron adquirir conocimientos fundamentales en desarrollo web, combinando HTML y JavaScript para crear páginas funcionales, ordenadas y dinámicas. Se afianzaron conceptos como estructura del documento, inserción de elementos multimedia, manipulación del DOM, creación de alertas y manejo de errores. Además, la práctica constante fortaleció el pensamiento lógico y la capacidad de identificar, analizar y solucionar errores en el código. En conjunto, estos aprendizajes representan una base sólida para avanzar hacia desarrollos web más complejos.

Referencias

Krakedev. (2025). *Curso: Primeros Pasos*. Course Platform.

Mozilla Developer Network. (2025). *HTML: HyperText Markup Language*.
<https://developer.mozilla.org/>

Mozilla Developer Network. (2025). *JavaScript Documentation*.
<https://developer.mozilla.org/>