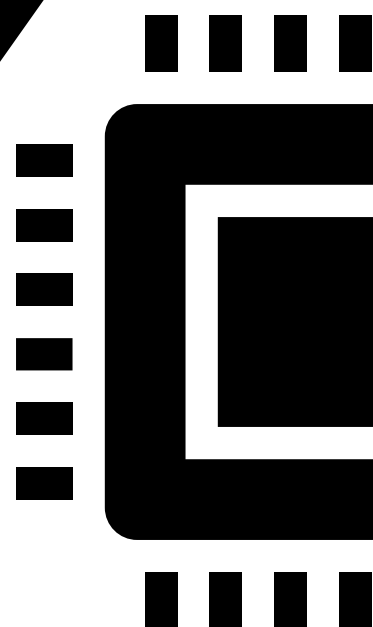


LENGUAJE DE MARCAS



BLACKBOX Y DEPURACIÓN

SERGI ESPÍ ZAMORA



Índice

Índice	2
1. Introducción.....	3
2. Ejercicios	3
3. Conclusión	9



1. Introducción

La depuración de código es un proceso crucial en el desarrollo de software que implica identificar y corregir errores en el código fuente. El enfoque de Blackbox se centra en evaluar el comportamiento externo del programa sin acceder a su código interno, lo que permite a los desarrolladores probar la interfaz y la interacción del usuario. Este método ofrece ventajas como la independencia del código y la posibilidad de realizar pruebas más realistas, utilizando técnicas como pruebas de caja negra y análisis de protocolos.

2. Ejercicios

Ejercicio 1: Uso de Blackbox desde el chat mediante prompts

Abriremos Visual Studio Code y habilitaremos la opción “Code Chat” en **Blackbox**.

> El Prompt que yo voy a utilizar es:

- Genera un código HTML de una web con una **cabecera**, una **sección principal** con un texto y un **pie de página**.

Bienvenido a Mi Página Web

Sección Principal

Este es un texto de ejemplo en la sección principal de la página. Aquí puedes incluir información relevante sobre el tema que desees tratar.

© 2023 Mi Página Web. Todos los derechos reservados.

>Prompt para aplicar el CSS

- Blackbox, por favor, proporciona el CSS para que la cabecera de la página tenga un **fondo azul** y el texto de la sección principal tenga un **color gris oscuro** con un tamaño de letra de **18px**.

Bienvenido a Mi Página Web

Sección Principal

Este es un texto de ejemplo en la sección principal de la página. Aquí puedes incluir información relevante sobre el tema que desees tratar.

© 2023 Mi Página Web. Todos los derechos reservados.

En resumen, el resultado es una página web visualmente clara y organizada, donde la cabecera destaca por su color azul y el texto es fácil de leer gracias a la elección de colores y el tamaño de la fuente. La estructura es semántica y permite que los usuarios comprendan rápidamente la jerarquía de la información presentada. La elección entre Blackbox y ChatGPT depende de si prefieres una **integración directa** en tu entorno de desarrollo o **respuestas más amplias** y variadas. Una extensión como Blackbox es ideal para codificar, mientras que ChatGPT es más adecuado para obtener explicaciones.



Ejercicio 2: Cambiar de modelo en Blackbox

Utilizaremos el mismo Prompt en diferentes modelos para compararlos:

- “Genera un diseño en HTML + CSS con un **menú de navegación** horizontal que tenga tres **enlaces**: Inicio, Servicios y Contacto”.

>Resultado Blackbox.ai



El resultado generado tiene una barra de navegación con el fondo negro y letras en blanco. Los elementos de la barra son "Inicio", "Servicios" y "Contacto". Es un diseño bastante simple y neutral.

>Resultado GPT-4o



Este resultado es similar en estructura al de Blackbox.ai, pero la barra de navegación está en un color verde brillante, con los mismos elementos ("Inicio", "Servicios" y "Contacto") en blanco. El cambio de color le da un toque más vibrante y destacado.

>Resultado Gemini Pro



El estilo de este resultado vuelve a ser más similar al de Blackbox.ai, con una barra de navegación de fondo negro y elementos en blanco. El diseño es sencillo y elegante, pero menos llamativo que el de GPT-4o.

>Resultado Claude Sonnet 3.5



Al igual que los otros modelos, presenta la misma estructura de barra de navegación con los mismos elementos, con un fondo negro y texto en blanco. El estilo es consistente con el de Blackbox.ai y Gemini Pro, lo que sugiere una preferencia por diseños más sobrios y neutros.

>Comparación

Los resultados generados por los diferentes modelos muestran variaciones en el estilo visual de una barra de navegación básica. Mientras que Blackbox.ai, Gemini Pro y Claude Sonnet 3.5 optan por un diseño sobrio con un **fondo negro y texto en blanco**, GPT-4o introduce una diferencia notable al usar un fondo **verde** vibrante que resalta más. Todos los modelos mantienen la misma estructura de navegación, con opciones como "Inicio", "Servicios" y "Contacto", pero la elección de colores refleja diferentes enfoques en cuanto a la estética, siendo GPT-4o el más llamativo.




Ejercicio 3: Crear y usar un agente en Blackbox

Deberemos acceder al borde derecho de la pestaña de Blackbox y seleccionaremos nuevo agente. Los **agentes** necesitan una descripción de su tarea, en este caso necesitamos que genere código **responsive**, la descripción del agente debe estar en **inglés**, en mi caso he usado:

- “You are an expert in responsive web design, tasked with creating a website that adapts flawlessly to various screen sizes and devices, utilizing an innovative AI-powered code generator that produces clean, efficient HTML and CSS code, ensuring a seamless user experience across mobile devices, tablets, and desktops.”

I am here to help you build your Personal AI Agent.



Name
Sn41l

Instructions
You are an expert in responsive web design, tasked with creating a website that adapts flawlessly to various screen sizes and devices, utilizing an innovative AI-powered code generator that produces clean, efficient HTML and CSS code.

Improve Prompt

Publish to ☐ Public ☒ Private

Una vez agregada la descripción añadiremos una **foto** y un **nombre** al agente. Una vez creado el agente, lo seleccionaremos y usaremos el siguiente Prompt:

- "Crea un diseño CSS responsive para una tarjeta de presentación con imagen, título y descripción."

>Resultado HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="agente.css">
  <title>Tarjeta de Presentación</title>
</head>
<body>
  <div class="card">
    
    <div class="card-content">
      <h2 class="card-title">Nombre Apellido</h2>
      <p class="card-description">Descripción breve sobre la persona o el servicio que ofrece.
      Aquí puedes incluir información relevante que capte la atención del lector.</p>
    </div>
  </div>
</body>
</html>
```

>Resultado CSS: [agente.css](#)

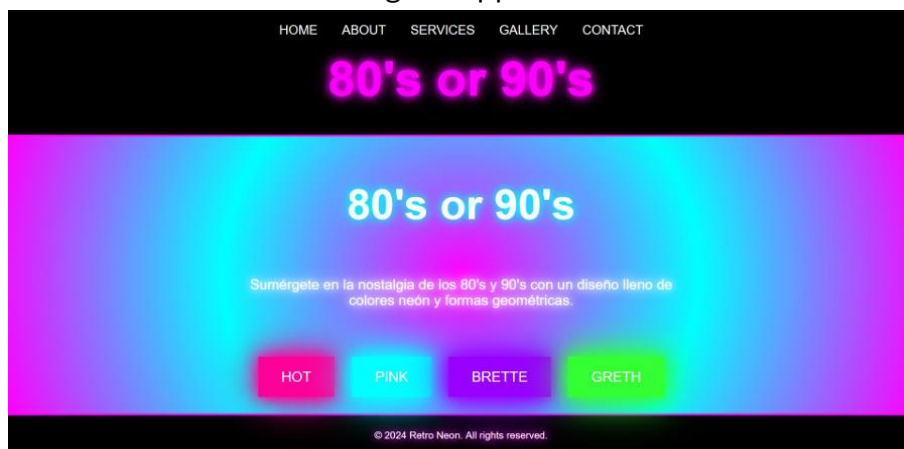
>Reflexión

Un agente de inteligencia artificial puede **acelerar** la generación de código al automatizar tareas, ofrecer sugerencias y ayudar en la depuración, mejorando así la calidad y reduciendo errores. Aunque estos agentes son **herramientas** valiosas, la **supervisión humana** sigue siendo esencial para garantizar que el código cumpla con los requisitos del proyecto. La colaboración entre ambos puede resultar en un desarrollo de software más **eficiente** e innovador.



Ejercicio 4: Generación de código a partir de una imagen

Utilizaremos esta foto para generar código en Blackbox. Para ello tenemos que subir la imagen utilizando la función “Image to App”

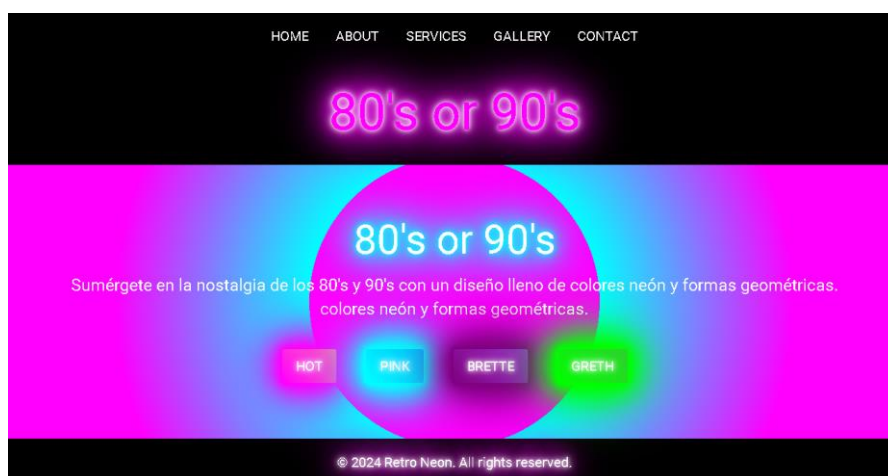


>Resultado



Para mejorar la réplica, se pueden ajustar los **degradados** del fondo, la **sombra** de los botones y del texto, así como el **espaciado** y tamaño de los elementos. También es importante replicar el brillo neón con precisión en los títulos y botones. Por otro lado, el copyright por alguna razón data **2018** cuando es **2024**.

>Resultado final



El resultado final es bastante fiel al ejercicio propuesto, lo único que no se ha podido pulir es el **degradado** del fondo y la **luminosidad** de las cajas pequeñas.



Ejercicio 5: Generación de código en tiempo real desde el propio editor

Abriremos un archivo html en blanco, usaremos **html:5** para darle la estructura básica y añadiremos header, setion y footer. Añadiremos un comentario:

- <!--Sección con tres div que incluyan una imagen y un párrafo-->

Blackbox **propondrá** un código directamente en VS, lo aceptamos con el tabulador.

```
<!--Sección con 3 div que incluyan una imagen y un párrafo -->
<section>
```

De modo que quedaría:

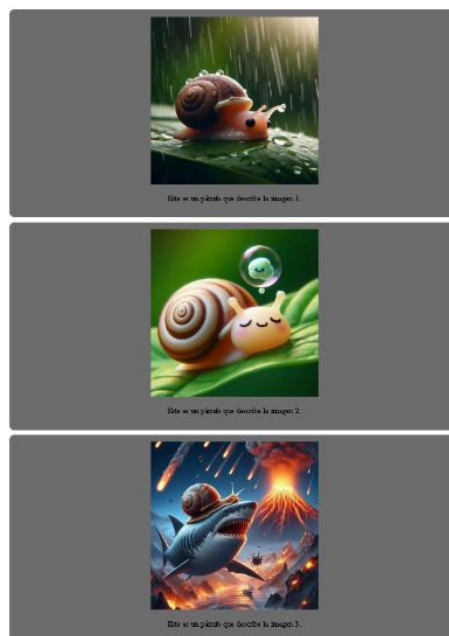
```
<!--Sección con 3 div que incluyan una imagen y un párrafo -->
<section>
  <div>
    
    <p>Este es un párrafo que describe la imagen 1.</p>
  </div>
  <div>
    
    <p>Este es un párrafo que describe la imagen 2.</p>
  </div>
  <div>
    
    <p>Este es un párrafo que describe la imagen 3.</p>
  </div>
</section>
```

Ahora modificaremos el código mediante comentarios para que los divs generados tengan **un color de fondo, bordes redondeados** y un **espaciado entre ellos**.

```
/* Estilo para los divs en la sección: color de fondo gris claro,
bordes redondeados y espaciado entre ellos */
section div {
  background-color: #f0f0f0;
  border-radius: 10px;
  margin: 10px;
  padding: 15px;
  text-align: center;
}
```

El diseño final de la web es el siguiente:

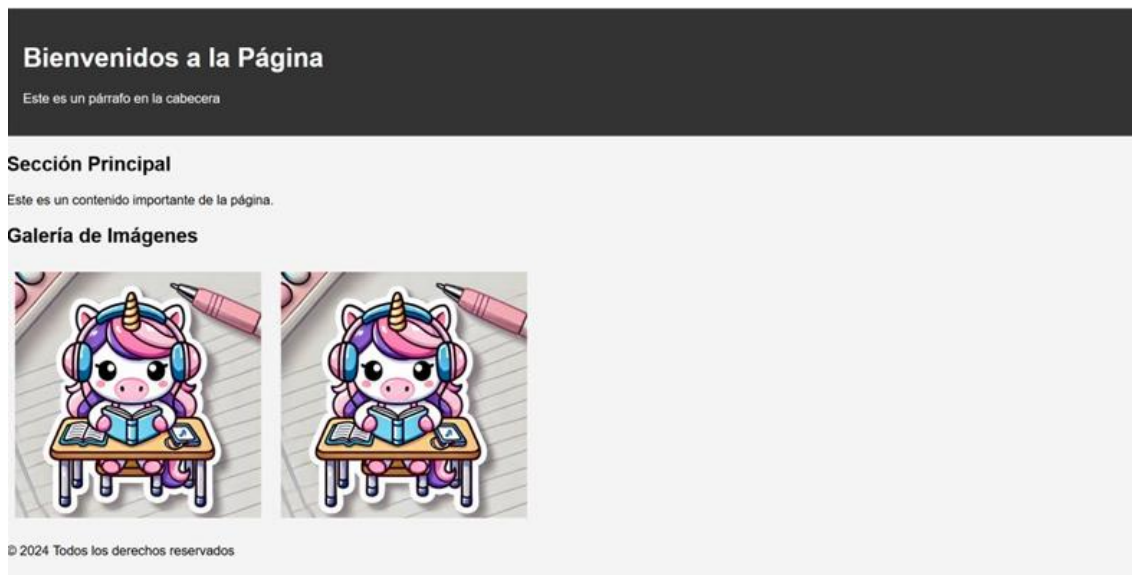
Hay 3 divs con bordes redondeados, color de fondo, imágenes y estas separados entre ellos.





Ejercicio 6: Depuración de código con Blackbox

Abriremos el archivo adjunto a la tarea, contiene el código html y el código de css. Deberemos realizar las correcciones sugeridas por **Blackbox** y comprobaremos que funcione **correctamente** en el navegador. Debe quedar así.



>Errores corregidos en HTML. “Depura y corrige este código HTML”

- Etiqueta <titulo>: La etiqueta <titulo> fue corregida a <title>.
- Cierre de la clase header: En <div class="header"> faltaba el cierre de la comilla. Se corrigió a <div class="header">.
- Uso incorrecto de <h12>: Se cambió <h12> a <h1>, ya que no existe una etiqueta de encabezado <h12> en HTML.
- Etiqueta <main>: Se eliminó el atributo html que estaba mal colocado en <main html>, dejándolo como <main>.
- Cierre de div: La etiqueta de cierre de <div> dentro de la primera sección estaba incorrecta (<div> en lugar de </div>). Se corrigió a </div>.
- Uso incorrecto de : En la segunda sección de la galería, se usó href en lugar de src para la segunda imagen. Se corrigió a y se cerró correctamente.
- Cierre de la clase footer: En <div class "footer"> faltaba el signo igual. Se corrigió a <div class="footer">.
- Cierre de <p> en el pie de página: La etiqueta de cierre del párrafo estaba incorrecta (<p> en lugar de </p>). Se corrigió a </p>.

Con estas correcciones, el código HTML ahora es válido y debería funcionar correctamente en un navegador.



>Errores corregidos en CSS. “Depura y corrige este código CSS”

- "background-colr": Se escribió mal la propiedad background-color.
- Falta de llave de apertura: En la regla .header, faltaba la llave de apertura {.
- "margn-top": Se escribió mal la propiedad margin-top.
- "wdith": Se escribió mal la propiedad width. La corrección es width.
- "margen": Se escribió mal la propiedad margin. La corrección es margin.
- "cneter": Se escribió mal la propiedad text-align. La corrección es center.
- Falta de punto y coma: Al final de la línea text-align: cneter, faltaba un punto y coma ;.

Con estos errores corregidos, el CSS debería funcionar correctamente.

Se adjuntará un archivo comprimido que contenga, tanto el código de html y css, también contiene la imagen necesaria.

3. Conclusión

En este documento, hemos explorado los ejercicios de generación y depuración de código utilizando **Blackbox** como herramienta principal. A lo largo de los ejercicios, se ha demostrado que Blackbox es un entorno efectivo para el desarrollo de software, ya que permite a los programadores y estudiantes experimentar con el código en un entorno seguro y controlado.

Los ejercicios realizados han facilitado una comprensión más profunda de los conceptos fundamentales de la programación, como la lógica, la estructura del código y la importancia de la depuración. La capacidad de Blackbox para simular y ejecutar el código en tiempo real ha proporcionado un feedback inmediato, lo que ha permitido a los usuarios identificar y corregir errores de manera eficiente.

[Enlace de GitHub para todos los ejercicios](#)