

Reporte Semana 4 – Taller de Estandarización y Depuración de Código

(*Week 4 Report – Code Standardization and Debugging Workshop*)

Nombre: Melanny Guate

Asignatura: Arquitectura y Diseño del Software

Fecha: 04 de noviembre de 2025

1. Depuración de Enlaces / Link Debugging

Pregunta / Question:

¿Qué errores se encontraron en el archivo index.html y cómo se solucionaron?

(*What errors were found in the index.html file and how were they fixed?*)

Respuesta / Answer:

Durante la verificación inicial del proyecto, se detectaron dos errores en los enlaces del archivo index.html:

1. En la etiqueta <link>, el archivo CSS estaba mal referenciado como styles.css (con una “s” adicional).
El nombre correcto del archivo es style.css.
2. En la etiqueta <script>, el archivo JavaScript estaba escrito incorrectamente como app-legcy.js, omitiendo la letra “a” de *legacy*.

Estos errores impedían que la aplicación cargaría correctamente los estilos y la funcionalidad.

Para solucionarlo, se aplicó la Opción B (Corregir HTML), ajustando los nombres para que coincidieran con los archivos reales del proyecto.

```
<!-- Antes / Before -->
<link rel="stylesheet" href="styles.css">
<script src="app-legcy.js"></script>

<!-- Después / After -->
<link rel="stylesheet" href="style.css">
<script src="app-legacy.js"></script>
```

Tras la corrección, la aplicación heredada cargó tanto los estilos visuales como la funcionalidad JavaScript sin modificar la estructura del frontend.

English Version:

During the initial verification of the project, two link errors were found in the index.html file:

1. In the <link> tag, the CSS file was incorrectly referenced as styles.css (with an extra “s”).
The correct name of the file is style.css.
2. In the <script> tag, the JavaScript file was written incorrectly as app-legcy.js, missing the letter “a” from *legacy*.

These errors prevented the application from loading styles and functionality properly. To fix them, Option B (Fix HTML) was applied by correcting the filenames to match the actual project files.

```
<!-- Antes / Before -->
<link rel="stylesheet" href="styles.css">
<script src="app-legcy.js"></script>

<!-- Después / After -->
<link rel="stylesheet" href="style.css">
<script src="app-legacy.js"></script>
```

After the correction, the legacy application successfully loaded both its styles and JavaScript functionality without altering the frontend structure.

2. Guías de Estilo / Style Guides

Pregunta / Question:

¿Por qué son importantes las guías de estilo en el desarrollo de software?
(*Why are style guides important in software development?*)

Respuesta / Answer:

Las guías de estilo son fundamentales porque garantizan que el código sea claro, consistente y fácil de mantener. Permiten que varios desarrolladores trabajen en un mismo proyecto sin generar confusión, reducen errores comunes y aumentan la calidad general del software.

English Version:

Style guides are fundamental because they ensure that the code is clear, consistent, and easy to maintain.

They allow multiple developers to work on the same project without confusion, reduce common mistakes, and improve the overall quality of the software.

3. Proceso de Estandarización / Standardization Process

Pregunta / Question:

¿Qué reglas de estilo se aplicaron en el archivo app-legacy.js y cómo mejoran el código?

(*Which style rules were applied to the app-legacy.js file and how do they improve the code?*)

Respuesta / Answer:

En el archivo app-legacy.js se realizó una auditoría manual completa para corregir todas las violaciones de estilo, sin alterar la funcionalidad original.

Las reglas aplicadas fueron las siguientes:

1. no-var: Se reemplazaron todas las declaraciones var por let o const, evitando problemas de alcance y reasignación.

(*All var declarations were replaced by let or const to prevent scope and reassignment issues.*)

2. camelCase: Se renombraron variables de snake_case a camelCase (por ejemplo: ultimo_operador → ultimoOperador, todo_list → todoList).

(*Variables were renamed from snake_case to camelCase, e.g., ultimo_operador → ultimoOperador, todo_list → todoList.*)

3. eqeqeq: Se reemplazaron == y != por === y !==, garantizando comparaciones estrictas entre tipo y valor.

(*Operators == and != were replaced with === and !== for strict type and value comparison.*)

4. quotes: Se estandarizó el uso de comillas simples ' ' para mantener consistencia en todo el archivo.

(*Single quotes ' ' were standardized throughout the file for consistency.*)

```
// Antes / Before  
var ultimo_operador = "+";  
  
// Después / After  
let ultimoOperador = '+';
```

Justificación / Justification:

Estas reglas no alteraron la lógica original, pero mejoraron la legibilidad, mantenibilidad y calidad profesional del código.

(These rules did not change the original logic but improved the code's readability, maintainability, and professional quality.)

4. Conclusión / Conclusion

Pregunta / Question:

¿Cuál fue el resultado final después de la depuración y estandarización?
(What was the final result after debugging and standardization?)

Respuesta / Answer:

Después de aplicar las correcciones en los enlaces y la estandarización del código, la aplicación heredada cumple con las buenas prácticas de programación. El proyecto ahora es más limpio, legible y fácil de mantener, sin modificar el diseño ni la lógica funcional.

English Version:

After applying the link corrections and code standardization, the legacy application follows good programming practices. The project is now cleaner, more readable, and easier to maintain without altering its design or functionality.

5. Archivos Incluidos / Included Files

- *index.html → Corregido (errores en style.css y app-legacy.js ajustados).*
- *app-legacy.js → Estandarizado (sin comentarios ni cambios de lógica).*