



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

**APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR  
CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)**

**David Alberto García Illescas**

Asesorado por el Ing. Víctor Bladimir Castellanos Albeño

Guatemala, abril de 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR  
CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**DAVID ALBERTO GARCÍA ILLESCAS**

ASESORADO POR EL ING. VÍCTOR BLADIMIR CASTELLANOS ALBEÑO

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, ABRIL DE 2024

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

|            |  |
|------------|--|
| DECANO     | Ing. José Francisco Gómez Rivera (a. i.) |
| VOCAL II   | Ing. Mario Retano Escobedo Martínez      |
| VOCAL III  | Ing. José Milton de León Bran            |
| VOCAL IV   | Ing. Kevin Vladimir Armando Cruz Lorente |
| VOCAL V    | Ing. Fernando José Paz González          |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez          |

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

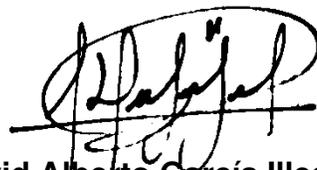
|            |  |
|------------|--|
| DECANO     | Ing. José Francisco Gómez Rivera (a. i.) |
| EXAMINADOR | Ing. Álvaro Giovanni Longo Morales       |
| EXAMINADOR | Ing. Luis Fernando Espino Barrios        |
| EXAMINADOR | Ing. Manuel Haroldo Castillo Reyna       |
| SECRETARIO | Ing. Hugo Humberto Rivera Pérez          |

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 25 de agosto del 2021.

A handwritten signature in black ink, appearing to read 'David Alberto García Illescas', written over a horizontal line.

**David Alberto García Illescas**

Guatemala, 14 de noviembre de 2023

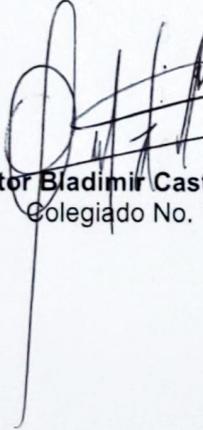
Ingeniero  
**Carlos Alfredo Azurdia**  
Coordinador de Privados y Trabajos de Tesis  
Escuela de Ingeniería en Ciencias y Sistemas  
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **DAVID ALBERTO GARCÍA ILLESCAS** con carné 201021158 y CUI 2316 93362 0101 titulado "APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)", luego de corroborar que el mismo se encuentra finalizado, lo he revisado y doy fe de que el mismo cumple con los objetivos propuestos en el respectivo protocolo, por consiguiente, procedo a la aprobación correspondiente.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
Ing. Victor Bladimir Castellanos Albeño  
Colegiado No. 16411

Victor Bladimir Castellanos Albeño  
INGENIERO EN CIENCIAS Y SISTEMAS  
COLEGIADO 16411



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 1 de febrero de 2024

Ingeniero  
**Carlos Gustavo Alonzo**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **DAVID ALBERTO GARCÍA ILLESCAS** con carné **201021158** y CUI **2316 93362 0101** titulado **“APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

SIST.LNG.DIRECTOR.14.EICCSS.2024

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)**, presentado por: **David Alberto Garcia Illescas**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”



Ingeniero Carlos Gustavo Alonzo  
DIRECTOR  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, abril de 2024

Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, -Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS), Maestría en Sistemas Mención construcción y Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas, Licenciatura en Matemática, Licenciatura en Física. Centros: de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad Universitaria, Zona 12, Guatemala, Centroamérica



**USAC**  
TRICENTENARIA  
Universidad de San Carlos de Guatemala

Decanato  
Facultad e Ingeniería

24189101- 24189102

LNG.DECANATO.OIE.174.2024

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **APLICACIÓN CON LECCIONES INTERACTIVAS PARA REFORZAR CONCEPTOS DEL LENGUAJE DE HOJAS DE ESTILO EN CASCADA (CSS)**, presentado por: **David Alberto Garcia Illescas** después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. José Francisco Gómez Rivera  
Decano a.i.



Guatemala, abril de 2024

Para verificar validez de documento ingrese a <https://www.ingenieria.usac.edu.gt/firma-electronica/consultar-documento>

Tipo de documento: Correlativo para orden de impresión Año: 2024 Correlativo: 174 CUI: 2316933620101

Escuelas: Ingeniería Civil, Ingeniería Mecánica Industrial, Ingeniería Química, Ingeniería Mecánica Eléctrica, - Escuela de Ciencias, Regional de Ingeniería Sanitaria y Recursos Hidráulicos (ERIS). Postgrado Maestría en Sistemas Mención Ingeniería Vial. Carreras: Ingeniería Mecánica, Ingeniería Electrónica, Ingeniería en Ciencias y Sistemas. Licenciatura en Matemática. Licenciatura en Física. Centro de Estudios Superiores de Energía y Minas (CESEM). Guatemala, Ciudad

## **ACTO QUE DEDICO A:**

- Dios** Por la vida que me dio y las oportunidades que me ha brindado para seguir creciendo como persona.
- Mi abuela** Odilia, por su amor y por haberme educado y guiado por el camino del bien, para que llegue a ser la persona que soy ahora.
- Mi hermana** Azucena, por todo el apoyo incondicional que me dio durante el tiempo de estudio de la carrera.
- Mis padres** Por haberme brindado la oportunidad de estudiar y llegar a superarme hasta este punto.
- Mi familia cercana** Hermanos, tíos y primas, quienes siempre han sido una compañía para superar los malos momentos y celebrar los buenos.
- Mis amigos** Rodrigo, Wilson y Jorge, por ser las personas que la vida me dio para recorrer este camino con su invaluable compañía.



## **AGRADECIMIENTOS A:**

|   |   |
|---|---|
| <b>Universidad de San Carlos de Guatemala</b> | Por ser la casa de estudio que me permitió completar este ciclo profesional.                                |
| <b>Facultad de Ingeniería</b>                 | Por proporcionar los conocimientos necesarios para que me convirtiera en un profesional.                    |
| <b>Ingeniero Bladimir</b>                     | Por brindarme su asesoría en esta última etapa y así lograr la culminación de esta meta.                    |
| <b>Daniel Mendoza</b>                         | Quien fue el amigo que me brindó su ayuda y apoyo incondicional durante la etapa más difícil de la carrera. |



## ÍNDICE GENERAL

|  |      |
|--|------|
| ÍNDICE DE ILUSTRACIONES.....   | V    |
| LISTA DE SÍMBOLOS .....  | VII  |
| GLOSARIO .....   | IX   |
| RESUMEN.....   | XI   |
| OBJETIVOS.....   | XIII |
| INTRODUCCIÓN .....   | XV   |
| <br>   |      |
| 1. MARCO TEÓRICO.....  | 1    |
| 1.1. Importancia del diseño en sitios web.....                         | 1    |
| 1.1.1. Historia del diseño para sitios web .....                       | 1    |
| 1.1.2. Diseño para sitios web.....                                     | 3    |
| 1.1.3. Beneficios de un buen diseño web .....                          | 3    |
| 1.2. Lenguajes utilizados para la creación de páginas web.....         | 5    |
| 1.2.1. HTML.....   | 5    |
| 1.2.2. CSS .....   | 6    |
| 1.2.3. Javascript.....   | 6    |
| 1.3. Conocimientos mínimos para un desarrollador <i>frontend</i> ..... | 7    |
| 1.3.1. Habilidades de un desarrollador <i>frontend</i> .....           | 8    |
| <br>   |      |
| 2. ESTRATEGIA DE LA SOLUCIÓN.....                                      | 11   |
| 2.1. Antecedentes.....   | 11   |
| 2.2. Solución propuesta.....   | 12   |
| 2.2.1. <i>Benchmarking</i> de la aplicación .....                      | 13   |
| 2.3. Objetivos de la aplicación .....                                  | 14   |
| 2.3.1. Temas para las lecciones .....                                  | 14   |

|        |   |    |
|--------|---|----|
| 2.3.2. | Ejercicios en las lecciones.....  | 15 |
| 2.4.   | Definición de módulos .....   | 16 |
| 2.4.1. | Módulo general.....   | 17 |
| 2.4.2. | Módulo de lecciones.....  | 17 |
| 2.4.3. | Módulo de interacción .....   | 17 |
| 2.4.4. | Módulo gráfico.....   | 18 |
| 2.5.   | Plan de desarrollo para la construcción y despliegue de la aplicación ..... | 18 |
| 2.5.1. | Análisis .....  | 18 |
| 2.5.2. | Diseño .....  | 18 |
| 2.5.3. | Desarrollo .....  | 18 |
| 2.5.4. | Pruebas.....  | 19 |
| 2.5.5. | Implementación .....  | 19 |
| 3.     | ANÁLISIS.....   | 21 |
| 3.1.   | Tecnologías y herramientas utilizadas en el desarrollo.....                 | 21 |
| 3.2.   | Funcionalidad.....  | 22 |
| 3.2.1. | Requerimientos funcionales .....  | 22 |
| 3.2.2. | Requerimientos no funcionales .....   | 27 |
| 3.3.   | Identificación y definición de procesos .....                               | 27 |
| 3.3.1. | Flujo principal .....   | 28 |
| 3.3.2. | Abandono de juego .....   | 28 |
| 3.3.3. | Reanudación de juego .....  | 29 |
| 3.3.4. | Reinicio de juego.....  | 30 |
| 3.4.   | Diseño de prototipos .....  | 31 |
| 4.     | DISEÑO .....  | 35 |
| 4.1.   | Definición de estándares de desarrollo .....                                | 35 |
| 4.1.1. | Herramientas.....   | 35 |

|          |  |    |
|----------|--|----|
| 4.1.2.   | Organización de archivos .....                     | 35 |
| 4.1.3.   | Nomenclatura de archivos .....                     | 36 |
| 4.1.4.   | Nomenclatura de variables y funciones .....        | 36 |
| 4.2.     | Lecciones en la aplicación .....                   | 37 |
| 4.2.1.   | Temas principales.....                             | 37 |
| 4.2.2.   | Lecciones.....                                     | 38 |
| 4.2.2.1. | Lección 1. Selectores basados en relaciones .....  | 38 |
| 4.2.2.2. | Lección 2. <i>Display outside</i> .....            | 39 |
| 4.2.2.3. | Lección 3. <i>Display inside - flex</i> .....      | 39 |
| 4.2.2.4. | Lección 4. <i>Display inside – grid</i> .....      | 40 |
| 4.2.2.5. | Lección 5. Posición ajustada al contenedor.....    | 40 |
| 4.2.2.6. | Lección 6. Posición ajustada a la pantalla.....    | 41 |
| 4.2.2.7. | Lección 7. Selectores basados en propiedades ..... | 41 |
| 4.2.2.8. | Lección 8. Selectores basados en acciones .....    | 42 |
| 4.3.     | Arquitectura de software .....                     | 42 |
| 5.       | IMPLEMENTACIÓN .....                               | 45 |
| 5.1.     | Despliegue de la aplicación .....                  | 45 |
| 5.2.     | Requerimientos .....                               | 45 |
| 6.       | MANUAL DE UTILIZACIÓN .....                        | 47 |
| 6.1.     | Inicio del juego.....                              | 47 |
| 6.2.     | Dinámicas en los niveles .....                     | 48 |
| 6.3.     | Finalización del juego .....                       | 49 |

CONCLUSIONES.....51  
RECOMENDACIONES .....53  
REFERENCIAS .....55  
APÉNDICES.....57

## ÍNDICE DE ILUSTRACIONES

### FIGURAS

|                   |  |    |
|-------------------|--|----|
| <b>Figura 1.</b>  | Diagrama del flujo principal.....                | 28 |
| <b>Figura 2.</b>  | Diagrama de flujo para abandono de juego .....   | 29 |
| <b>Figura 3.</b>  | Diagrama de flujo para reinicio de un juego..... | 30 |
| <b>Figura 4.</b>  | Diagrama de flujo para reinicio de un juego..... | 31 |
| <b>Figura 5.</b>  | Prototipo para la pantalla de inicio .....       | 32 |
| <b>Figura 6.</b>  | Prototipo para pantalla de lección.....          | 33 |
| <b>Figura 7.</b>  | Prototipo para pantalla de finalización .....    | 34 |
| <b>Figura 8.</b>  | Componentes de la arquitectura.....              | 43 |
| <b>Figura 9.</b>  | Pantalla de inicio del juego .....               | 47 |
| <b>Figura 10.</b> | Pantalla del nivel uno del juego .....           | 48 |

### TABLAS

|                  |                                    |    |
|------------------|------------------------------------|----|
| <b>Tabla 1.</b>  | Historia de usuario 1 .....        | 22 |
| <b>Tabla 2.</b>  | Historia de usuario 2 .....        | 22 |
| <b>Tabla 3.</b>  | Historia de usuario 3 .....        | 23 |
| <b>Tabla 4.</b>  | Historia de usuario 4 .....        | 24 |
| <b>Tabla 5.</b>  | Historia de usuario 5 .....        | 24 |
| <b>Tabla 6.</b>  | Historia de usuario 6 .....        | 25 |
| <b>Tabla 7.</b>  | Historia de usuario 7 .....        | 25 |
| <b>Tabla 8.</b>  | Historia de usuario 8 .....        | 26 |
| <b>Tabla 9.</b>  | Historia de usuario 9 .....        | 26 |
| <b>Tabla 10.</b> | Requerimientos no funcionales..... | 27 |



## LISTA DE SÍMBOLOS

| <b>Símbolo</b> | <b>Significado</b>            |
|----------------|-------------------------------|
| <b>px</b>      | Abreviatura del término pixel |
| <b>W3C</b>     | Consortio World Wide Web      |



## GLOSARIO

|                            |   |
|----------------------------|---|
| <b>Backend</b>             | Capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. |
| <b><i>Benchmarking</i></b> | Término de origen inglés que hace referencia al proceso de recopilar, comparar y analizar determinado producto con otros del mismo tipo o mercado.                                    |
| <b><i>Camelcase</i></b>    | Tipo de escritura en la se escriben las palabras sin dejar espacio entre ellas. Cada palabra inicia con letra mayúscula para diferenciarlas.  |
| <b>CSS</b>                 | <i>Cascade style sheet</i> , por sus siglas en inglés. Es un lenguaje aplicado a sitios web para la estilización visual de elementos HTML.  |
| <b><i>Frontend</i></b>     | Parte de un programa o dispositivo a la que un usuario puede acceder directamente.  |
| <b>HTML</b>                | <i>Hyper text markup language</i> , por sus siglas en inglés. Lenguaje utilizado para la maquetación de páginas web.  |

|                      |  |
|----------------------|--|
| <b>HTTP</b>          | <i>Hipertext transfer protocol</i> , por sus siglas en inglés. Significa en español: protocolo de transferencia de hipertexto.               |
| <b>IDE</b>           | Siglas que significan entorno de desarrollo integrado. Software utilizado que combina herramientas de desarrollo en una interfaz de usuario. |
| <b><i>Layout</i></b> | Es un esquema de la distribución de los elementos dentro de una página web.  |
| <b>URL</b>           | Ruta textual utilizada por los navegadores para acceder a un sitio web.  |
| <b>WWW</b>           | World Wide Web.  |

## RESUMEN

Los sitios web son las aplicaciones creadas por desarrolladores más utilizadas actualmente por su fácil acceso gracias al Internet. Las características importantes de un sitio web son la funcionalidad confiable y la seguridad, sin embargo, también la apariencia gráfica de un sitio es importante, porque ayuda a que sea más comprensible y fácil su uso por parte de los usuarios.

El presente trabajo de graduación se adentra en la importancia del diseño web, abordando desde los fundamentos teóricos hasta la implementación práctica. En el marco teórico, se destaca la relevancia del diseño y se explora el uso de lenguajes, especialmente CSS, para construir páginas web efectivas. Se establecen conocimientos mínimos para desarrolladores *front-end* y se examina la preparación de estudiantes en desarrollo web.

La estrategia de solución presenta antecedentes, una solución propuesta con objetivos claros, un *benchmarking* de la aplicación y la definición de módulos, respaldados por un plan de desarrollo. El análisis detalla las tecnologías, funcionalidades y procesos, respaldado por prototipos. El diseño se centra en estándares, pantallas clave y arquitectura de software. La implementación cubre el despliegue, requisitos y desarrollo. El manual de utilización ofrece pautas para iniciar un juego, dinámicas en niveles y reiniciar el juego.



# OBJETIVOS

## General

Desarrollar una aplicación que permita al usuario aprender conceptos del lenguaje de estilizado CSS por medio de lecciones prácticas e interactivas.

## Específicos

1. Crear una o más lecciones por cada tema de CSS seleccionado para enseñar.
2. Explicar de una manera fácil y breve cada lección.
3. Comprobar el aprendizaje de cada lección por medio de la resolución de un reto.
4. Diseñar retos orientados al objetivo de la lección, presentando problemas fáciles de entender, los cuales debe resolver el usuario de forma práctica.
5. Mostrar al usuario, de una forma gráfica, agradable y estilizada, los elementos que describen el reto a resolver.



## INTRODUCCIÓN

En la era digital actual, el diseño efectivo de sitios web y la comprensión de las herramientas utilizadas para su construcción son elementos cruciales en la formación de profesionales en el ámbito del desarrollo web. Esta es la razón por la que los estudiantes de la carrera de ingeniería en sistemas reciben una introducción esencial a la maquetación de páginas web en el curso inicial de programación.

Aunque el énfasis primordial se centra en la programación orientada a objetos utilizando Java, la transición hacia el desarrollo web no siempre abarca de manera equitativa la comprensión de la estilización con CSS. La relevancia de este lenguaje de hojas de estilo es a menudo subestimada, ya que muchos estudiantes, al enfrentarse a proyectos con complejidades lógicas y técnicas, tienden a relegar la importancia de la parte visual y estética de un sitio web. En este contexto, surge la necesidad de una herramienta que enriquezca el aprendizaje de CSS, proporcionando lecciones interactivas y prácticas para consolidar los conceptos fundamentales en diseño web.

Es común que los programadores se vean desafiados al asumir responsabilidades de diseño sin la experiencia necesaria en la estilización de páginas mediante CSS. El presente trabajo de graduación ofrece una solución a este desafío mediante el desarrollo de una aplicación con lecciones interactivas, diseñadas para facilitar el proceso de maquetación de páginas web. Esta aplicación no solo beneficia a los estudiantes de ingeniería en sistemas en sus primeros pasos en el mundo del diseño web, sino que también proporciona una

herramienta valiosa para cualquier persona que desee consolidar sus conocimientos en CSS.

# 1. MARCO TEÓRICO

## 1.1. Importancia del diseño en sitios web

Una página web es un documento que es mostrado al usuario utilizando un navegador web. Estas páginas web son utilizadas para mostrar información u obtener datos solicitados al usuario que realiza la consulta de la página.

En la actualidad, las compañías o instituciones desean obtener y mostrar una gran cantidad de información a los usuarios por medio de un navegador web y que este sea fácil de consultar. Con la finalidad de no utilizar solamente una página web que contenga los bloques de información, ya que esto puede hacer más difícil la ubicación de una sección deseada, se crean varias páginas con porciones de información o solicitudes de datos, a las cuales se puede tener acceso utilizando un menú de navegación. A este conjunto de páginas web enlazadas se les conoce como sitio web.

### 1.1.1. Historia del diseño para sitios web

Siegel (1997), en su libro *Técnicas avanzadas para el diseño de páginas web*, distinguía 3 generaciones para el diseño en páginas web. Para la actualidad se puede agregar una generación más, “hoy podemos hablar de una cuarta generación de webs en las que prima tanto el diseño gráfico como la tecnología [...] el hipertexto ya no es una colección de textos enlazados, sino un espacio de intercambio de servicios” (Colocia, 2015, p. 3). Con base en la definición de los dos autores, entre las características de cada generación para el diseño web se pueden mencionar:

- Primera generación: estructura lineal, eventual inserción de fotografías y líneas de separación, baja definición (proyectados para terminales ASCII en de un solo color).
- Segunda generación: adición de íconos para los enlaces, menú de opciones, fondos coloreados o con imágenes, bordes azules alrededor de las figuras interactivas.
- Tercera generación: hegemonía del diseño sobre la tecnología, utilización de metáforas, uso de *layout* tipográfico y visual para la descripción de una página bidimensional, estructura entrada-área central-salida.
- Cuarta generación: dinamismo de los elementos gráficos según la información de los servicios *backend*, reestructuración de los componentes de acuerdo con el tamaño de la pantalla, *single page application* (sitios web que cuentan con una sección cambiante, las demás secciones se mantienen estáticas).

En la actualidad, con la ayuda de los avances en la tecnología de los dispositivos electrónicos y las mejoras en los lenguajes de programación utilizados para la maquetación de sitios web, los sitios web comerciales cuentan con una buena estética y se ha ido definiendo estándares en la ubicación de los elementos o íconos utilizados para indicar una funcionalidad en específico.

### **1.1.2. Diseño para sitios web**

Comúnmente, es aplicado un diseño a los sitios web con la finalidad de mejorar dos aspectos:

- Interfaz de usuario: esto comprende la forma en la que son mostrados a los usuarios todos los elementos de la página web, por ejemplo: tamaño de textos, color de los botones, estructura de los menús, diferenciación de las secciones, entre otros aspectos.
- Experiencia de usuario: esto se refiere a la disposición de los elementos en la pantalla y las opciones habilitadas en las páginas web que facilitan la interacción con el sitio y la navegación entre las distintas páginas.

Un sitio web con un apartado visual estético y una usabilidad fácil o conocida generará en el usuario una experiencia agradable, abriendo la oportunidad de que el usuario desee visitar nuevamente el sitio web, lo cual es beneficioso para los sitios comerciales, y aumentará también la probabilidad de que un usuario finalice una transacción en línea, debido a que no tuvo problemas en entender los pasos a realizar (Digital Talent Agency, 2022).

### **1.1.3. Beneficios de un buen diseño web**

Se ha generado en las compañías una obligación no escrita: poseer un sitio web. Los usuarios actuales ya están acostumbrados a obtener información de forma rápida a través del Internet, y tener un sitio web que brinde no solo la información que buscan, sino además información que la compañía desea que los usuarios sepan, ofrece una oportunidad para atraer más clientes. Para lograr lo descrito anteriormente son necesarias dos cosas: facilitar la búsqueda de la información que el usuario desea e incentivar al usuario a quedarse en el sitio

web para explorarlo. Esos objetivos pueden ser alcanzados con la implementación de un buen diseño web. A continuación se listan algunos beneficios de un buen diseño web:

- Acercamiento de calidad a los usuarios. No existe una segunda oportunidad para una primera impresión, según la famosa frase del dramaturgo y novelista irlandés Oscar Wilde. La frase aplica no solamente para las presentaciones físicas, sino virtuales, e impactar a un usuario con un buen diseño web en su primera visita al sitio influye indudablemente en su deseo de visitarlo nuevamente y recomendarlo.
- Proyección de la entidad corporativa. El diseño de un sitio web es una extensión de la imagen de la compañía, es otra manera de presentarse ante el mundo.
- Buena experiencia de usuario. Generar un espacio donde los usuarios se sientan cómodos, con elementos que sean fáciles de manejar y con una usabilidad que les permita navegar con facilidad, ya que esto aumenta la posibilidad de visitar nuevamente el sitio y el deseo de visualizar las opciones que se encuentran en el mismo.

SEO (*search engine optimization*, por sus siglas en inglés). Ninguna compañía, sea grande o pequeña, quiere estar muy debajo en las opciones de los motores de búsqueda de sitios web. La optimización orgánica en los motores de búsqueda puede incrementar según la experiencia de usuario. Google, el motor de búsqueda más utilizado, toma en cuenta la adaptabilidad de un sitio web hacia dispositivos móviles, algo que forma parte del diseño.

## **1.2. Lenguajes utilizados para la creación de páginas web**

Berners-Lee (1989) describió su propuesta para permitir la transferencia de información a través de una red, incluyendo la forma en la que se podría visualizar esta información, en el documento titulado *Information Managment: A Proposal*. Este documento contenía la descripción del protocolo de transferencia de hipertextos (HTTP, por sus siglas en inglés), el cual sería utilizado para transferir archivos que contenían texto en el lenguaje de maquetación HTML, que sirve para la construcción de la interfaz gráfica que representaría la información recibida.

Un año después fue implementada la solución y, en el año 1990, nació lo que se conoce como WWW (*world wide web*). Al hacerse pública esta solución, su creador Berners-Lee decidió fundar el Consorcio Mundial de la Red (abreviado como W3C), el cual sería la institución encargada de regular los estándares a utilizar en el WWW. De esta forma el lenguaje de maquetación HTML sigue siendo, hasta la fecha, el único lenguaje reconocido para la modelación de sitios web (BBC News Mundo, 2019).

Con el objetivo de mejorar la estilización visual de las páginas web, en 1995 el W3C desarrolló el lenguaje CSS que, hasta la fecha, sigue siendo el único lenguaje reconocido para la estilización de sitios web.

### **1.2.1. HTML**

Este lenguaje tiene la característica de definir sus elementos por medio de etiquetas, las cuales pueden estar definidas de tres formas: la primera es con una palabra encerrada dentro de los signos mayor que y menor que, y es conocida como etiqueta de apertura, por ejemplo: <html>; la segunda forma es llamada una etiqueta de cierre y posee una estructura parecida a la etiqueta de apertura,

pero tiene el signo diagonal antes de la palabra de la etiqueta, por ejemplo: `</html>`; y la tercera forma es colocando el signo diagonal al final de la palabra de la etiqueta, esto se utiliza para distinguir etiquetas que no necesitan una etiqueta de cierre, por ejemplo: `<input/>`.

### **1.2.2. CSS**

Cuando es creado un conjunto de atributos que estilizarán elementos, es necesario indicar los selectores a los cuales pertenecen estos atributos. Los selectores permiten delimitar las etiquetas que serán afectadas por los atributos descritos (W3 Schools, s.f.).

### **1.2.3. Javascript**

HTML y CSS son considerados lenguajes de maquetación y estilización, respectivamente. En estos lenguajes no es posible desarrollar procedimientos programados para que sean ejecutados en las páginas web. Ante la necesidad de añadir acciones programadas surgió el lenguaje de programación Javascript en 1995. En 1997 fue estandarizado su uso en los navegadores web, dándole el nombre de ECMAScript al lenguaje, sin embargo, hasta la fecha sigue siendo llamado con su nombre original Javascript.

Agregar un lenguaje de programación al ambiente de sitios web mejoró la experiencia de usuario y también la cantidad de funciones de los sitios web, ya que permitía a los desarrolladores agregar un comportamiento programado. Actualmente sigue siendo el único lenguaje de programación que reconocen los navegadores para las páginas web.

## **1.3. Conocimientos mínimos para un desarrollador *frontend***

El modelo cliente-servidor, utilizado desde la primera implementación de páginas web, consiste en dos partes que permiten la navegación web: el servidor que proporciona los recursos informáticos necesarios para que una página web sea visualizada y el cliente que realiza la solicitud de estos recursos al servidor para que se puedan ejecutar localmente en una máquina.

Pero la interacción del cliente con el servidor no finaliza cuando la página web se encuentra cargada, las páginas pueden realizar solicitudes hacia el servidor como parte de una acción del usuario o como parte de la misma lógica de la página. Las acciones programadas en la página web son desarrolladas en el lenguaje mencionado como estándar para la programación: Javascript, pero del lado del servidor no se cuenta con un lenguaje estandarizado y, a lo largo de los años, han surgido varios lenguajes con la capacidad de procesar las peticiones que los sitios web realizan.

Surgen entonces dos enfoques en el desarrollo web: uno centrado en la parte con la que el usuario interactúa directamente, con componentes visuales y estilizados, además de tener una composición orientada a una buena experiencia de usuario, y otro enfocado al procesamiento de datos que el usuario solicita, formateando y verificando información, realizando consultas hacia una base de datos u otros servidores. A estos dos enfoques se les llama *frontend* y *backend*.

Se conoce con el término *frontend* a todo lo relacionado con el diseño y desarrollo de las páginas web que forman un sitio web. Esta parte es la que se puede observar desde el navegador, el usuario interactúa directamente con sus elementos y es la que tiene sus tecnologías de desarrollo reguladas por la W3C. Los lenguajes utilizados para su desarrollo son los 3 mencionados anteriormente: HTML, CSS y Javascript.

Se conoce con el término *backend* a los servicios programados para que el *frontend* pueda realizar peticiones, ya sea solicitando información o enviando información para que sea procesada. Esto es lo que se conoce como «el lado del servidor» y este no se encuentra regulado más que en el protocolo utilizado para recibir y enviar información al *frontend*, debido a esto existen muchas tecnologías y lenguajes de programación utilizados para desarrollar esta parte.

Al existir diferencias significativas en la construcción del *frontend* y *backend*, muchos desarrolladores se especializan en algunas de estas dos ramas, por lo que términos como desarrollador *frontend* o desarrollador *backend* son muy utilizados en el ámbito de la programación web (Barba, 2015).

### **1.3.1. Habilidades de un desarrollador *frontend***

Al tener que desarrollar la parte interactiva con el usuario, adicional al conocimiento sobre los lenguajes de programación utilizados, son necesarias otras habilidades relacionadas con el diseño. Con base en Acosta (2020) y Digital Talent Agency (2022), a continuación se listan los conocimientos y habilidades más importantes:

- HTML y CSS: como se mencionó anteriormente, son los lenguajes utilizados para la maquetación y estilización de los componentes en una página web.
- Javascript: único lenguaje utilizado para la programación de procedimientos y funciones en la página web. Este lenguaje es el que permite realizar y recibir peticiones desde el servidor y, además, permite

realizar verificación de datos, así como cambiar los elementos de la página web en tiempo real para un mejor dinamismo.

- *Frameworks*: un *framework*, por su traducción del inglés, se puede considerar como un marco definido para que el desarrollador realice su trabajo. Los *frameworks* ponen a disposición del desarrollador funciones útiles para utilizar al momento de la codificación. Se puede decir que un *framework* es solo una herramienta que brinda ayuda al desarrollador para escribir el código de mejor forma y más rápido. Actualmente no son muy utilizados los lenguajes HTML, CSS y Javascript de forma pura al momento de desarrollar un sitio web. En cambio, se utilizan *frameworks* que facilitan la maquetación, estilización y, principalmente, ayudan mucho en la programación de procedimientos a ejecutar con Javascript. Un desarrollador *frontend* en la actualidad debe conocer algún *framework*, ya que es la tendencia que se está implementando en los proyectos de desarrollo de software web.
- Sentido del diseño: un desarrollador *frontend* participa en la construcción de una parte muy visual para el usuario. Por lo tanto, es importante poseer cierta habilidad y conocimiento acerca del diseño de forma real. La creatividad llega a ser una característica muy notable en el desarrollador, adicional al conocimiento sobre el manejo del lenguaje utilizado para el diseño (CSS).
- Diseño *responsive*: los usuarios utilizan distintos dispositivos para acceder a un sitio web como computadores, *tablets*, celulares, entre otros. Estos dispositivos tienen distintos tamaños de pantalla, por lo que un sitio web no debería verse de igual forma en una computadora que en un teléfono celular. A raíz de esta necesidad, el diseño *responsive* (también conocido

como diseño adaptativo) permite que los elementos de una página web cambien en su forma y posición dependiendo del tamaño de la pantalla del dispositivo que está mostrando la página web. Esta característica *responsive* es agregada por parte del desarrollador y, en la actualidad, es indispensable que esta persona tenga conocimientos de cómo implementarlo.

Como se puede ver en la lista descrita anteriormente, la combinación entre conocimientos técnicos y conocimientos acerca de diseño es necesaria para que un desarrollador *frontend* pueda desempeñar sus funciones de forma eficiente.

## **2. ESTRATEGIA DE LA SOLUCIÓN**

### **2.1. Antecedentes**

En la Universidad de San Carlos de Guatemala, los estudiantes de la carrera de ingeniería en sistemas reciben una introducción a la maquetación de páginas web en el primer curso de programación. Esta introducción es breve, ya que el objetivo del curso es enseñar al estudiante la programación orientada a objetos utilizando el lenguaje Java. En los siguientes cursos, al estudiante le es solicitado que elabore páginas web centrándose en la funcionalidad y no en la parte visual-estética del sitio web.

Muchos estudiantes se centran en la búsqueda de la solución lógica y técnica para realizar los proyectos que en los distintos cursos de la carrera se le solicitan y no tanto en el apartado visual, esto es debido normalmente a que las funcionalidades definidas en estos proyectos provocan que el estudiante invierta mucho tiempo en investigación y desarrollo, restando importancia al esfuerzo por aprender más acerca del lenguaje de estilización CSS, esto apoyado también con la idea de que, al no ser un lenguaje de programación, es más fácil su implementación.

En Guatemala, se dan casos en los cuales el equipo de desarrollo no cuenta con una persona especializada para la maquetación de un sitio web, y, si cuentan con un diseñador, crea solamente la plantilla, pero no genera el maquetado con los lenguajes HTML y CSS. Este trabajo es delegado entonces al programador, que muchas veces no tiene experiencia o amplio conocimiento con el manejo de la estilización de páginas utilizando CSS. Esto puede llegar a

causar frustración en el desarrollador al no avanzar a la velocidad deseada o no lograr el resultado esperado, debido a que no puede implementar algo tan sencillo como CSS.

## **2.2. Solución propuesta**

El lenguaje CSS, como se mencionó anteriormente, consiste en un conjunto de atributos aplicados a los elementos HTML. No es un lenguaje complejo de entender ni de definir, pero cuenta con una gran variedad de atributos, los cuales, a su vez, admiten la asignación de distintos tipos de valores. Esto puede agregar complejidad al lenguaje, porque el desarrollador o desarrolladora debe seleccionar los atributos y valores adecuados para lograr el efecto visual deseado que muchas veces puede variar por el comportamiento de la página web y, al ser tantas las opciones, puede invertir una significativa cantidad de tiempo en investigar y probar el efecto que tienen los atributos hasta obtener el resultado deseado.

La estilización visual, al no ser un tema primordial dentro del pensum de estudios de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos, los estudiantes deben aprenderla por su cuenta y, al no exigirse su implementación en los proyectos que los alumnos desarrollan para la aprobación de cursos, se releva hacia un segundo plano de importancia. Esto tiene consecuencias en el ámbito laboral cuando se ven forzados a utilizarlo, ya que necesitan realizar investigaciones para aplicarlo de forma adecuada y muchas veces el tiempo indicado para el desarrollo no toma en cuenta este tiempo de investigación, por lo que puede resultar en un trabajo apresurado.

Con el objetivo de brindar un apoyo al estudiante en su aprendizaje del lenguaje CSS, fue ideada la aplicación con lecciones interactivas para reforzar

conceptos del lenguaje de hojas de estilo en cascada. Esta es una aplicación web a la cual no puede accederse desde cualquier navegador utilizando una dirección URL pública. La aplicación provee lecciones cortas acerca del lenguaje CSS, cada lección tiene un ejercicio interactivo que debe completarse para acceder a la siguiente lección. Los ejercicios interactivos fueron diseñados a manera de minijuegos, con el objetivo de mantener la atención del usuario e incentivar a completar la lección a través de un reto.

### **2.2.1. Benchmarking de la aplicación**

Actualmente, existen otras aplicaciones web gratuitas con el objetivo de dar lecciones acerca de CSS con una forma interactiva, ya sea a través de minijuegos o animaciones. Entre estas aplicaciones se puede mencionar:

- Flexbox Froggy: es un juego en el cual se desea ayudar a un personaje (Froggy) a llegar a un lugar en específico de la pantalla utilizando la propiedad de CSS *display: flex*, agregando también otros atributos que modifican el posicionamiento del personaje en la pantalla. El juego está orientado a enseñar el manejo de la propiedad *flex*.
- CSS Diner: es un juego con el objetivo de enseñar los selectores utilizados en CSS y los tipos que existen. La dinámica del juego consta de escribir los selectores necesarios para cumplir con la instrucción que le es dada al usuario. Gráficamente muestra objetos y alimentos sobre una mesa, los cuales deben ser seleccionados según su selector. La desventaja de este juego es que se encuentra solamente en idioma inglés.
- Flexbox Defense: en este juego, al igual que en Flexbox Froggy, se utiliza el atributo de CSS *flex* para posicionar unas torres de ataque de tal forma

que eliminen enemigos al momento de que se proceda a iniciar la animación. Este juego se centra solamente en el posicionamiento de elementos utilizando *flexbox* y tiene la desventaja de que se encuentra solamente en idioma inglés.

- Grid Garden: este último juego tiene el objetivo de crear ejercicios para aprender la propiedad de CSS *grid*. La dinámica del juego consiste en una matriz que representa un jardín en el cual se deberán ir posicionando elementos en secciones específicas para hacer crecer distintas plantas.

Se puede observar que las aplicaciones mencionadas anteriormente se centran específicamente en una propiedad o característica de CSS, mostrando las distintas formas en las que se puede manipular y, desafortunadamente, solamente dos de estas se encuentran en idioma español, abarcando los temas acerca de las propiedades *flexbox* y *grid*.

## **2.3. Objetivos de la aplicación**

El objetivo principal de la aplicación es brindar lecciones acerca de CSS al usuario, proveyendo ejercicios interactivos para practicar lo aprendido. Para cumplir con este objetivo, es necesario definir y delimitar dos aspectos: los temas a enseñar y los ejercicios a implementar.

### **2.3.1. Temas para las lecciones**

Esta aplicación es un apoyo para el estudiante en su aprendizaje de CSS, por lo que no se pretende enseñar el lenguaje desde cero ni abarcar la totalidad de temas, ya que este cuenta con más de 170 propiedades que pueden ser aplicadas a los elementos HTML. Intentar abarcar la mayoría o totalidad de estas propiedades resultaría en una cantidad muy grande de lecciones, las cuales, a su vez, necesitarían una gran cantidad de tiempo para su desarrollo. Se delimitaron los temas a tratar considerando cuáles propiedades son las más utilizadas e importantes durante la maquetación de un sitio web.

Fue considerado que los temas relacionados a los selectores para aplicar atributos, posicionamiento y despliegue de elementos y cambios dinámicos, son los conocimientos mínimos necesarios para un buen uso de CSS. Los temas seleccionados para incluir en las lecciones son:

- Selectores y relación entre selectores
- Despliegue de elementos (*display*)
- Posicionamiento de elementos (*position*)
- Pseudoclases
- Pseudoelementos
- Disposición (*layout*)
- Cambios dinámicos

Estos temas tampoco serán abarcados en su totalidad, ya que pueden ser implementados utilizando y combinando diferentes atributos de CSS, las lecciones estarán orientadas a enseñar las instrucciones más comúnmente aplicadas.

### **2.3.2. Ejercicios en las lecciones**

Además del contenido teórico, se decidió agregar ejercicios prácticos en cada lección. El avance es gradual y para pasar a la siguiente lección se tiene que completar el ejercicio de la lección actual. Con la finalidad de presentar lecciones llamativas, y de esta forma incentivar al usuario a completar las lecciones, los ejercicios son presentados como prácticas interactivas, en las cuales el usuario puede agregar y modificar los atributos CSS para completar el reto presentado.

Los ejercicios son llamados entonces retos. Estos retos son visualizados de forma gráfica, utilizando elementos HTML, el usuario debe modificar las propiedades de los elementos para completar la instrucción dada por el reto. Si lo hace se tomará el reto como aprobado y pasará a la siguiente lección, de lo contrario puede seguir intentando hasta encontrar la solución.

Debido al contenido interactivo, fue decidido presentar la aplicación como un juego para aprender el lenguaje CSS. Cada lección representa un nivel, en cada nivel se tendrán que completar los ejercicios presentados como minijuegos, los cuales contendrían el reto a cumplir. Una vez completado el minijuego, el usuario avanzaría al siguiente nivel. La pantalla para cada lección sigue una estructura con tres secciones: una con el contenido teórico de la lección, otra con opciones en las cuales el usuario puede hacer modificaciones, ya sea escritas o con el *mouse*, y otra sección donde es posible visualizar los elementos del minijuego.

## **2.4. Definición de módulos**

Para la definición de módulos se agruparon los elementos de la aplicación según su funcionalidad. Esto dio como resultado la definición de cuatro módulos, los cuales se describen a continuación:

#### **2.4.1. Módulo general**

Este es el módulo principal y contiene el resto de los módulos que permiten la ejecución del juego. Este módulo maneja también las funciones que permiten el inicio y la finalización del juego.

#### **2.4.2. Módulo de lecciones**

Este módulo está encargado de mostrar el texto con el contenido teórico de la lección. Además, debe aplicar el formato establecido según el contenido textual.

#### **2.4.3. Módulo de interacción**

La sección donde el usuario seleccionará las instrucciones que serán ejecutadas por CSS para cambiar los elementos gráficos será manejada por este módulo. Las instrucciones pueden ser indicadas por el usuario por medio de texto escrito o selección de elementos ya existentes, por lo que este módulo incluirá un analizador de texto.

#### **2.4.4. Módulo gráfico**

Toda la parte visual del minijuego que le es presentada al usuario es manejada por este módulo. Está encargado de mostrar los elementos que serán afectados por las instrucciones indicadas en la sección de interacción, así como de crear animaciones según sea necesario.

## **2.5. Plan de desarrollo para la construcción y despliegue de la aplicación**

Dentro del plan de desarrollo del proyecto, el proceso de construcción para la aplicación fue dividido en cinco etapas: análisis, diseño, desarrollo, pruebas e implementación.

### **2.5.1. Análisis**

En la etapa actual fueron definidos y especificados los requerimientos de la aplicación, utilizados para delimitar el alcance en cuanto a funcionalidad. También fueron definidas las herramientas a utilizar para cumplir con los requerimientos.

### **2.5.2. Diseño**

En la etapa de diseño fueron creados los bosquejos de los distintos tipos de pantallas, indicando los elementos en cada una de estas. Fueron definidas las lecciones y los tipos de retos que acompañan a cada una de estas, esto incluye el diseño visual y funcionalidad de cada uno de los retos. También fue especificada la organización a utilizar para los elementos del código y el manejo de la información.

### **2.5.3. Desarrollo**

Tomando como base los requerimientos y las especificaciones del diseño, en esta etapa fue construida la aplicación por medio de código. Adicional al código también se crearon los elementos gráficos necesarios para la visualización de los minijuegos.

#### **2.5.4. Pruebas**

Esta etapa se mezcla con la etapa de desarrollo, ya que, al completar una funcionalidad, se pasaba a la etapa de pruebas para comprobar que la ejecución fuera correcta, detectar errores y oportunidades de mejora. Si era detectado algo por corregir o mejorar, se regresaba a la etapa de desarrollo para realizar la modificación. Si no eran detectados cambios, se regresaba a la etapa de desarrollo para continuar con la codificación.

#### **2.5.5. Implementación**

Una vez culminado el desarrollo de la aplicación, esta fue desplegada en un sitio web público para permitir el acceso a los diferentes usuarios. Una URL pública es utilizada para su fácil acceso.



## 3. ANÁLISIS

### 3.1. Tecnologías y herramientas utilizadas en el desarrollo

Durante cada una de las etapas del desarrollo fueron utilizadas distintas herramientas y tecnologías. Para el proyecto en general fueron utilizadas prácticas de la metodología ágil SCRUM, la cual fue seleccionada por su flexibilidad, por lo tanto, los requerimientos se encuentran definidos por medio de historias de usuario.

El código fuente de la aplicación no se encuentra en Javascript y CSS directamente, fue tomada la decisión de utilizar un *framework* de Javascript para facilitar la modularización del código y también el desarrollo. La lista de tecnologías a utilizar para el desarrollo y herramientas para el despliegue es la siguiente:

- HTML: lenguaje común para la maquetación de páginas web.
- Sass: preprocesador de CSS para el manejo de estilos.
- Angular 12: *framework* de Javascript utilizado para la codificación de la lógica y los procesos de la aplicación.
- Servidor en la nube para el despliegue de la aplicación.

Dentro del código de la aplicación se encuentran implementadas algunas librerías, estas serán detalladas en el capítulo 4 del documento.

### 3.2. Funcionalidad

La funcionalidad principal de la aplicación es permitir al usuario avanzar por los distintos niveles hasta llegar al último que, al ser completado, indicaría que el juego ha terminado. Esta funcionalidad se encuentra en otras pequeñas funcionalidades que mejoran la experiencia de usuario, además, se toma en cuenta también dentro de la funcionalidad las características generales y restricciones.

#### 3.2.1. Requerimientos funcionales

Toda la funcionalidad que define al juego se encuentra en esta sección, expresada por medio de historias de usuario. Cada historia de usuario describe el usuario que puede realizar la acción, número de historia, nombre y descripción.

**Tabla 1.**

*Historia de usuario 1*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 1</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Iniciar juego nuevo   |                         |
| <b>Descripción:</b><br>Iniciar un juego nuevo indicando el nombre que utilizará el jugador en todos los niveles. |                         |

*Nota.* Se describe la historia de usuario. Elaboración propia, realizado con Microsoft Excel.

**Tabla 2.**

*Historia de usuario 2*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 2</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Visualizar texto de lección   |                         |
| <b>Descripción:</b><br>El texto de la lección ocupará un área de la pantalla. En caso el texto sea extenso y el espacio asignado no sea suficiente, deberá utilizarse una función de desplazamiento que permita la lectura completa del texto. |                         |

*Nota.* Se describe la historia de usuario 2. Elaboración propia, realizado con Microsoft Excel.

### **Tabla 3.**

#### *Historia de usuario 3*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 3</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Reto para cada lección  |                         |
| <b>Descripción:</b><br>Cada lección debe tener un reto en forma de minijuego, el cual debe completarse utilizando la característica de CSS descrita en la lección. |                         |

*Nota.* Se describe la historia de usuario 3. Elaboración propia, realizado con Microsoft Excel.

**Tabla 4.**

*Historia de usuario 4*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 4</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Retos en forma de minijuegos  |                         |
| <b>Descripción:</b><br>El reto incluido en cada lección debe presentarse al usuario en forma de minijuego. |                         |

*Nota.* Se describe la historia de usuario 4. Elaboración propia, realizado con Microsoft Excel.

**Tabla 5.**

*Historia de usuario 5*

| Historia de usuario   |                         |
|---|-------------------------|
| <b>Número: 5</b>  | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Instrucciones para completar el minijuego  |                         |
| <b>Descripción:</b><br>El usuario debe utilizar instrucciones CSS que modifiquen la parte gráfica del minijuego para poder completarlo. Estas instrucciones pueden ser escritas o seleccionadas por el jugador. |                         |

*Nota.* Se describe la historia de usuario 5. Elaboración propia, realizado con Microsoft Excel.

**Tabla 6.**

*Historia de usuario 6*

| Historia de usuario   |                         |
|---|-------------------------|
| <b>Número: 6</b>  | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Cambio automático de nivel   |                         |
| <b>Descripción:</b><br>Al momento que un reto sea completado, al jugador le es mostrada la pantalla correspondiente del siguiente nivel o lección. En caso el reto completado pertenezca a la última lección, el juego será finalizado. |                         |

*Nota.* Se describe la historia de usuario 6. Elaboración propia, realizado con Microsoft Excel.

**Tabla 7.**

*Historia de usuario 7*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 7</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Guardado del estado del juego   |                         |
| <b>Descripción:</b><br>Si el jugador sale de la aplicación en un juego ya iniciado, al entrar nuevamente a la aplicación aparecerá en el nivel donde se quedó. |                         |

*Nota.* Se describe la historia de usuario 7. Elaboración propia, realizado con Microsoft Excel.

**Tabla 8.**

*Historia de usuario 8*

| Historia de usuario  |                         |
|--|-------------------------|
| <b>Número: 8</b>   | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Reinicio de juego   |                         |
| <b>Descripción:</b><br>El jugador, desde cualquier nivel, será capaz de reiniciar el juego. Al realizar esta acción el juego iniciará nuevamente en el primer nivel. |                         |

*Nota.* Se describe la historia de usuario 8. Elaboración propia, realizado con Microsoft Excel.

**Tabla 9.**

*Historia de usuario 9*

| Historia de usuario   |                         |
|---|-------------------------|
| <b>Número: 9</b>  | <b>Usuario: Jugador</b> |
| <b>Nombre:</b> Finalización del juego   |                         |
| <b>Descripción:</b><br>Cuando el jugador complete el reto del último nivel, se debe indicar que el juego fue completado y proporcionar la opción de reiniciarlo nuevamente. |                         |

*Nota.* Se describe la historia de usuario 9. Elaboración propia, realizado con Microsoft Excel.

### 3.2.2. Requerimientos no funcionales

Para los requerimientos no funcionales se consideraron las características que pueden afectar o inhabilitar la experiencia de usuario al momento de utilizar la aplicación. Los requerimientos y el comportamiento deseado según sus características se describen en la siguiente tabla:

**Tabla 10.**

*Requerimientos no funcionales*

| <b>Requerimiento</b>                        | <b>Descripción</b>  |
|---|---|
| Acceso público                              | Puede accederse a la aplicación por medio de un navegador desde cualquier conexión a Internet.  |
| Permanencia de datos de la partida iniciada | Cuando un juego es iniciado, los datos deben guardarse de forma local en el dispositivo utilizado   |
| Pantalla adaptativa                         | Los elementos en la pantalla deben adaptarse al tamaño de la ventana en el navegador al momento que este cambie de tamaño, sin embargo, el juego no estará diseñado para resoluciones menores a 960 px de ancho |
| Disponibilidad para su uso en navegadores   | La aplicación deberá ejecutarse en los navegadores web más utilizados, estos son: Chrome, Firefox, Opera y Safari.  |
| Disponibilidad                              | El servicio que provee la aplicación web puede ser consultado a cualquier hora del día  |

*Nota.* Requerimientos no funcionales descritos. Elaboración propia, realizado con Microsoft Excel.

Estos requerimientos, en conjunto con los requerimientos funcionales, definen todas las características de la aplicación. Con base en los requerimientos serán creados los diseños de las pantallas y la lógica de la aplicación, así como la arquitectura y tecnología a utilizar.

### 3.3. Identificación y definición de procesos

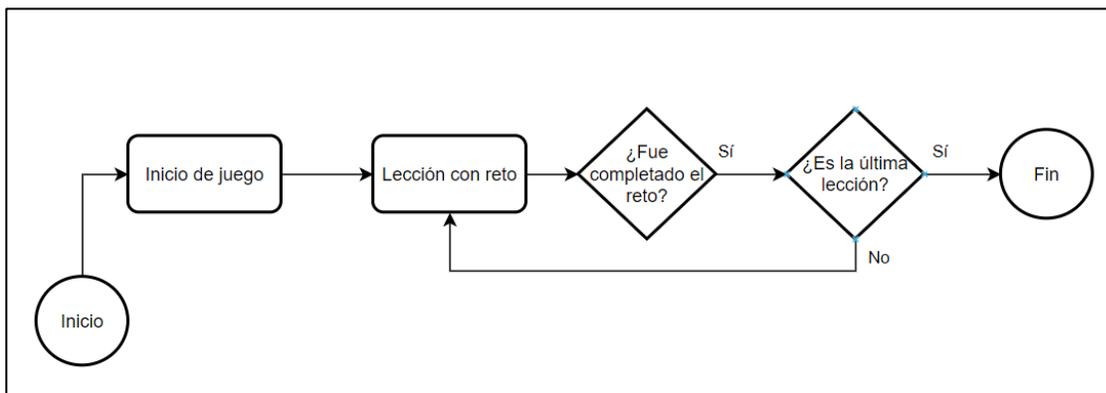
Los procesos, en este contexto, hacen referencia a los flujos que la funcionalidad de la aplicación puede seguir con base en las acciones del jugador. El flujo principal inicia al momento de comenzar una partida y termina cuando el juego es completado. Las acciones que el usuario realice dentro de estas dos referencias crean otros escenarios que son definidos por otros flujos.

### 3.3.1. Flujo principal

Como fue mencionado anteriormente, este inicia al momento de comenzar un juego y, si el usuario completa todos los niveles en este flujo, termina cuando el juego es completado.

#### Figura 1.

*Diagrama del flujo principal*



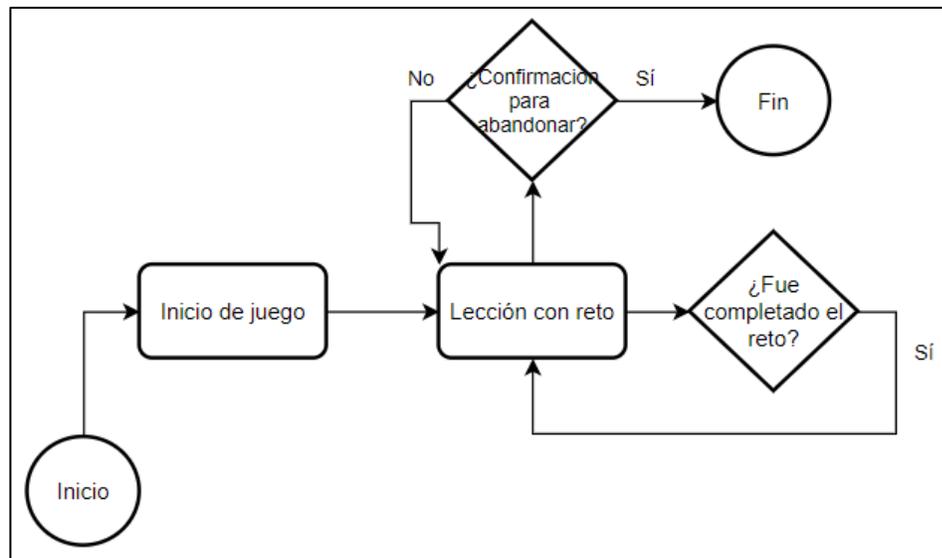
*Nota.* Flujo principal. Elaboración propia, realizado con Visio.

### 3.3.2. Abandono de juego

En el momento de que el usuario decida no continuar con el juego, podrá seleccionar una opción que le permitirá realizar esta acción siempre con la confirmación necesaria.

**Figura 2.**

*Diagrama de flujo para abandono de juego*



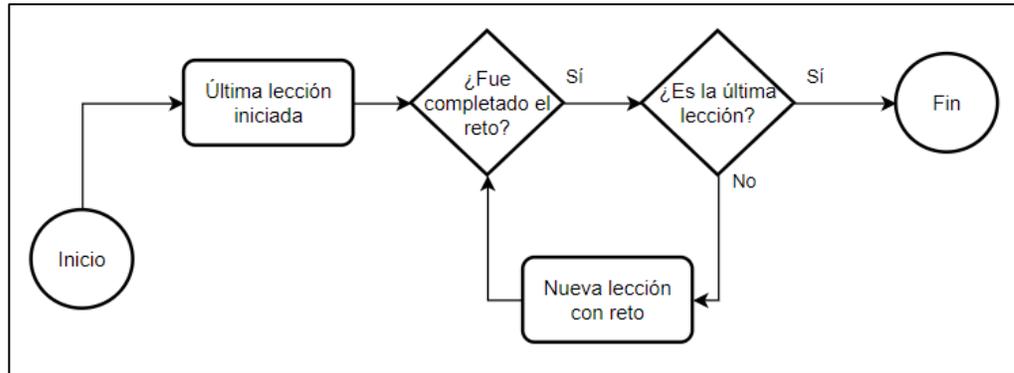
*Nota.* Diagrama para abandono del juego. Elaboración propia, realizado con Visio.

### 3.3.3. Reanudación de juego

Si el usuario abandonó el juego antes de concluir todos los niveles, puede iniciar nuevamente desde el nivel que quedó registrado. A continuación se muestra el diagrama correspondiente:

**Figura 3.**

*Diagrama de flujo para reinicio de un juego*



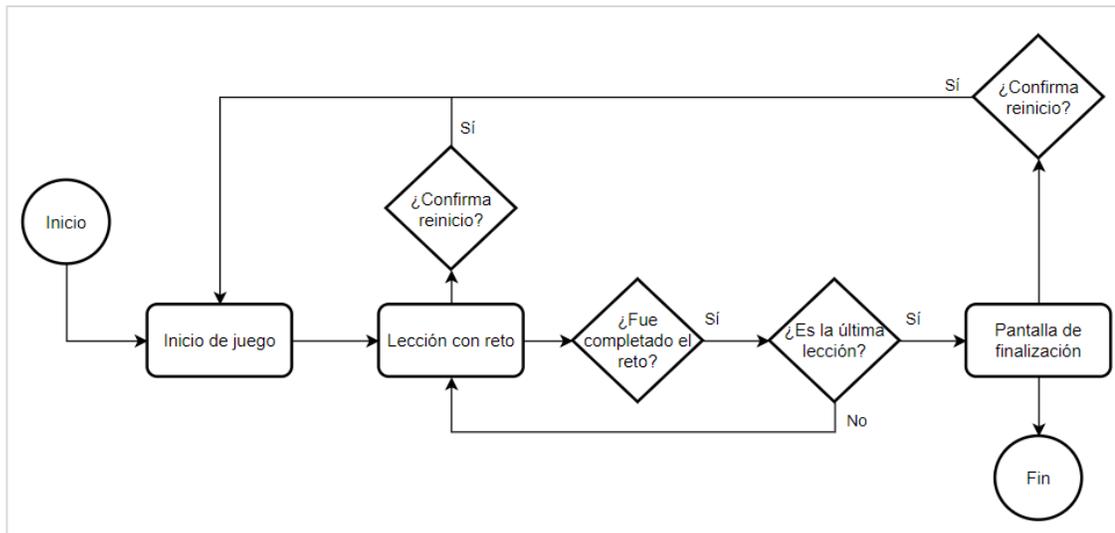
*Nota.* Diagrama para reinicio. Elaboración propia, realizado con Visio.

### **3.3.4. Reinicio de juego**

Al momento de concluir todos los niveles del juego, el usuario puede iniciar nuevamente desde el nivel uno, si así lo desea. A continuación se muestra el diagrama correspondiente:

**Figura 4.**

*Diagrama de flujo para el reinicio de un juego*



*Nota.* Diagrama de reinicio 2. Elaboración propia, realizado con Visio.

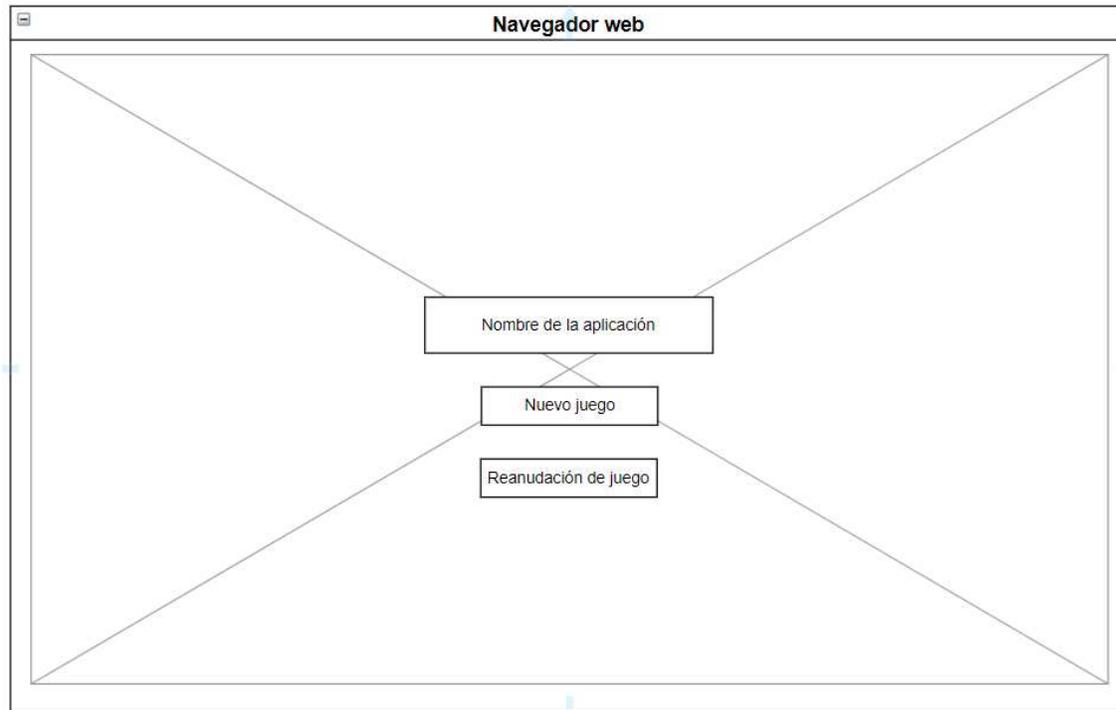
### 3.4. Diseño de prototipos

Los prototipos muestran un bosquejo de las pantallas de la aplicación. Esta aplicación cuenta con 3 tipos de pantallas: la pantalla de inicio, la pantalla donde las lecciones son mostradas con su reto y la pantalla de finalización del juego. Los prototipos muestran solamente los elementos que conforman cada tipo de pantalla y la ubicación de estos.

La pantalla de inicio contiene en su fondo una imagen que representa la aplicación, el nombre del juego y las opciones para iniciar un juego se encuentran en el centro.

## Figura 5.

*Prototipo para la pantalla de inicio*

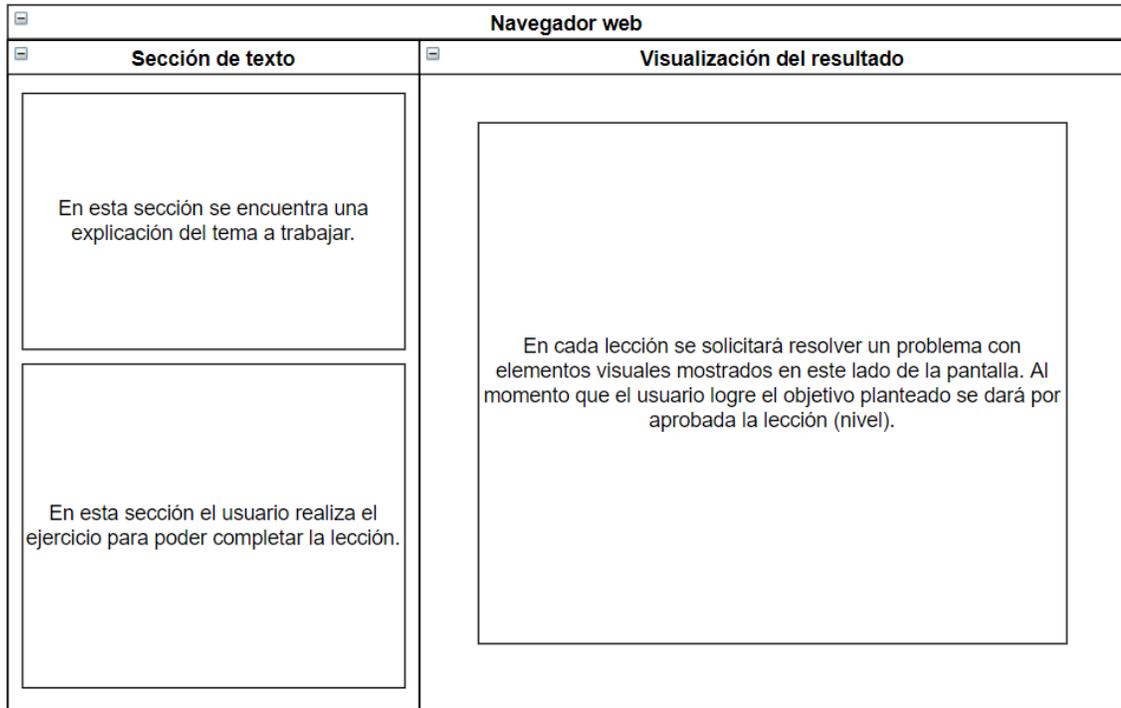


*Nota.* Pantalla de inicio (prototipo). Elaboración propia, realizado con Visio.

La pantalla que muestra la lección contiene tres secciones: una sección donde se encuentra el texto describiendo la característica de CSS, debajo de esta sección se encuentra el espacio donde el usuario realiza el ejercicio para completar el reto y la sección que ocupa mayor espacio es la destinada a mostrar el contenido visual para el reto.

## Figura 6.

### Prototipo para pantalla de lección

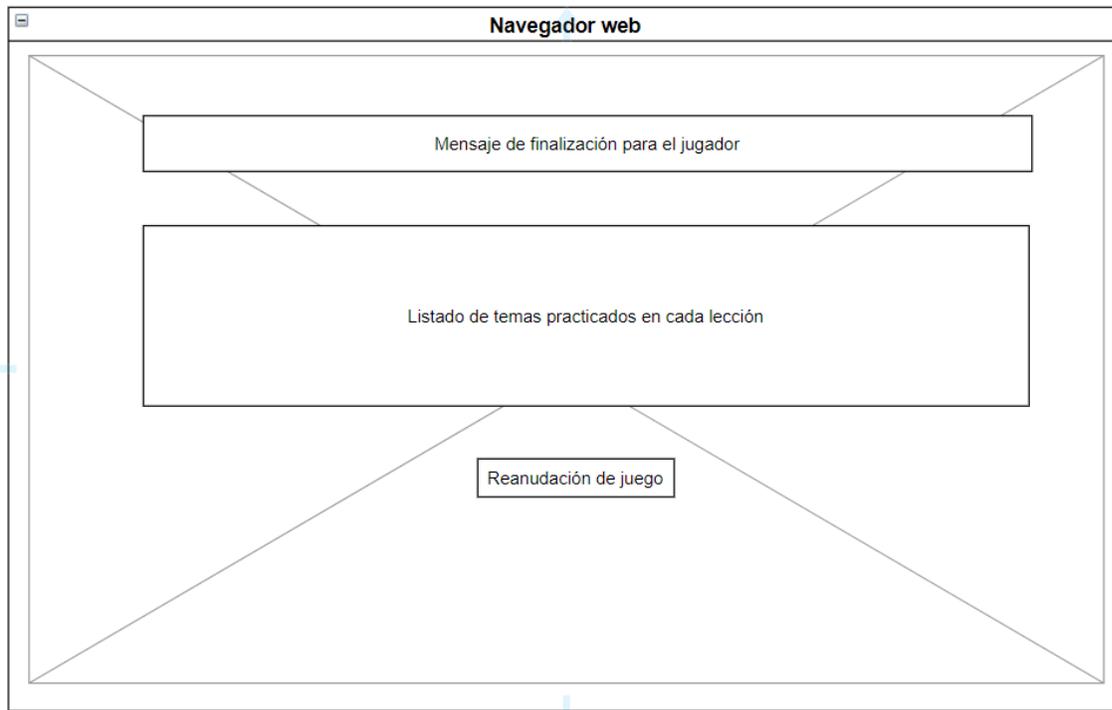


*Nota.* Pantalla de lección (prototipo). Elaboración propia, realizado con Visio.

La pantalla de finalización contiene una imagen de fondo relacionada al juego. En su contenido muestra un mensaje para el jugador y lista los temas aprendidos en cada lección.

**Figura 7.**

*Prototipo para pantalla de finalización*



*Nota.* Pantalla de finalización (prototipo). Elaboración propia, realizado con Visio.

## 4. DISEÑO

### 4.1. Definición de estándares de desarrollo

Todo el contenido que sea generado durante el desarrollo seguirá estándares para facilitar la búsqueda y modificación de sus elementos. En estos estándares se definen las herramientas a utilizar, nomenclatura para los archivos y elementos de codificación, estructura para los archivos y formatos.

#### 4.1.1. Herramientas

Las herramientas que se utilizan incluyen los *frameworks*, IDEs y herramientas de versionamiento:

- Git como controlador de versiones
- Angular en su versión 11 como *framework* para el código Javascript
- SCSS para los archivos de estilo
- Visual Studio Code como IDE de desarrollo

#### 4.1.2. Organización de archivos

Los archivos del código fuente de la aplicación estarán organizados dentro de carpetas en el proyecto según su tipo o funcionalidad:

- El código que pertenece a la lógica operacional (instrucciones escritas en *typescript*) se encuentran ubicados dentro de la carpeta *modules*.

- Los archivos de estilos SCSS se encuentran ubicados dentro de la carpeta *assets* dentro de la subcarpeta *styles*. Todos los archivos de estilos serán importados desde el archivo *styles.css* que se encuentra en el directorio raíz.
- Las imágenes se encuentran dentro de la carpeta *assets* en la subcarpeta *images*. Dentro de esta carpeta están agrupadas en subcarpetas dependiendo del uso que tengan.

#### **4.1.3. Nomenclatura de archivos**

Los archivos del código fuente tienen su nombre escrito en inglés y adicionalmente pueden tener texto que hace referencia al tipo de funcionamiento que tienen. Los archivos que conforman los componentes y módulos lógicos tienen la palabra *component* y *module*, respectivamente. El resto de los archivos solamente tienen un nombre acorde a su contenido.

#### **4.1.4. Nomenclatura de variables y funciones**

Las variables, funciones y clases siguen un estándar en su nomenclatura dependiendo del tipo de archivo en el que se encuentren. Se manejan tres tipos de archivos: *html*, *scss* y *typescript*.

- En los archivos *typescript* se tienen 4 tipos de elementos, los cuales siguen un estándar en su nomenclatura. Las variables se escriben en minúscula separando las palabras con el uso del guion bajo; las constantes tienen su nombre en mayúscula separando las palabras con guion bajo; las clases y las funciones utilizan la escritura *camelcase*, iniciando con la primera letra en minúscula para las funciones y en mayúscula para las clases.

- En los archivos SCSS las clases que necesiten separar palabras en su nombre utilizarán el guion para indicar esta diferencia.
- En los archivos HTML los identificadores utilizan el tipo de escritura *camelcase* iniciando con minúscula.

## **4.2. Lecciones en la aplicación**

Como fue mencionado anteriormente, la aplicación contiene lecciones sobre 7 temas de CSS y estos temas se encuentran distribuidos en 15 lecciones, cada lección contiene un reto o minijuego para completar y así practicar el contenido de la lección.

### **4.2.1. Temas principales**

Para la selección de los temas principales fue considerada la importancia de la utilidad de ciertos atributos al momento de implementar CSS en un documento HTML. Estos temas tratan sobre la selección de elementos para modificar sus propiedades, posicionamiento de elementos en la pantalla y cambios en los atributos, dependiendo del comportamiento del elemento o del usuario. Los temas principales son:

- **Selectores:** son los que se encargan de indicar cuáles son los elementos que se verán afectados por un conjunto de propiedades.
- *Display:* indica la forma en la cual los elementos serán ubicados en la sección donde se encuentran desplegados.

- *Position*: indica la posición que tendrá un elemento con relación a su contenedor o a la pantalla.
- Pseudoclases: estos ayudan a los selectores para aumentar la especificidad respecto a los elementos que serán afectados por un grupo de atributos. Hacen referencia a atributos o comportamientos en los elementos.
- Pseudoelementos: al igual que las pseudoclases, los pseudoelementos ayudan a los selectores, pero estos se enfocan en los subelementos.
- Cambios dinámicos: se refiere a cambios que pueden tener los atributos respecto al tamaño y tipo de pantalla donde son mostrados los elementos. Esta cualidad es más conocida como *responsive*.

#### **4.2.2. Lecciones**

Se escogieron subtemas a partir de los temas principales seleccionados. Estos subtemas se explican en las diferentes lecciones que tiene la aplicación. Una lección puede contener uno o más subtemas y fueron agrupados según la relación que tenían respecto a su definición o comportamiento. En total se cuenta con 2 lecciones para el tema de selectores, 3 lecciones para el tema de *display*, 2 lecciones para el tema de *position*, 3 lecciones para el tema de pseudoclases, 2 lecciones para el tema de pseudoelementos y 3 lecciones para el tema de cambios dinámicos.

##### **4.2.2.1. Lección 1. Selectores basados en relaciones**

- Tema principal: selectores.
- Contenido de la lección: es posible indicar una relación entre selectores para definir los elementos que se verán afectados, por ejemplo, cambiar los atributos de todos los hijos de un elemento. Esta lección indica las posibles relaciones que se pueden indicar entre selectores.
- Reto: el usuario debe utilizar la relación entre selectores para cambiar el color de los grupos de elementos siguiendo las instrucciones indicadas.

#### **4.2.2.2. Lección 2. *Display outside***

- Tema principal: *display*.
- Contenido de la lección: un elemento puede indicar la forma que utilizará para posicionarse en su contenedor a través de atributos denominados *display outside*. Esta lección habla de los dos tipos más comunes: *block* e *inline*.
- Reto: el usuario debe ordenar los elementos aplicando el valor *display* adecuado para cumplir con las instrucciones.

#### **4.2.2.3. Lección 3. *Display inside - flex***

- Tema principal: *display*.

- Contenido de la lección: es posible indicar la forma en la cual será posicionado el contenido de un elemento, a este tipo de propiedades se les conoce como *display inside*. Flex es una de las propiedades de *display* más utilizadas por su variedad de opciones al momento de indicar la forma de un despliegue, estas propiedades serán explicadas en la lección.
- Reto: el usuario debe posicionar los elementos siguiendo las instrucciones indicadas, además debe ordenar sus subelementos utilizando la propiedad *flex* para cumplir con las condiciones dadas por el reto.

#### **4.2.2.4. Lección 4. *Display inside – grid***

- Tema principal: *display*.
- Contenido de la lección: otra propiedad dentro del grupo *display inside* es la propiedad *grid*. Esta propiedad permite posicionar los elementos como en una tabla utilizando filas y columnas.
- Reto: el usuario debe posicionar los elementos indicados sobre otro grupo de componentes utilizando la propiedad *grid* para ordenarlos por filas y columnas.

#### **4.2.2.5. Lección 5. Posición ajustada al contenedor**

- Tema principal: *position*.

- Contenido de la lección: la posición de un elemento puede cambiar tomando como referencia el contenedor indicado como principal. Tres propiedades utilizadas para realizar estos cambios son *static*, *relative* y *absolute*.
- Reto: el usuario debe cambiar las propiedades relacionadas con la posición de los elementos para impedir que estos se sobrepongan uno con otro.

#### **4.2.2.6. Lección 6. Posición ajustada a la pantalla**

- Tema principal: *position*.
- Contenido de la lección: es posible indicar que la posición de un elemento depende de la pantalla, por lo que todo el tiempo será visible. Los valores de *position* utilizados para esto son *fixed* y *sticky*.
- Reto: el usuario debe utilizar los valores *fixed* y *sticky* sobre los elementos indicados como barreras para permitir que otro elemento se mueva libremente al mover el *scroll* hacia abajo y llegar al punto indicado como meta.

#### **4.2.2.7. Lección 7. Selectores basados en propiedades**

- Tema principal: pseudoclasses.

- Contenido de la lección: es posible seleccionar algunos elementos indicando las propiedades de etiqueta que tienen aplicadas, estas propiedades pueden ser: *enabled*, *read-only*, *required*, *disabled*, *checked*.
- Reto: el usuario debe cambiar las propiedades de los elementos en el texto siguiendo las instrucciones indicadas.

#### **4.2.2.8. Lección 8. Selectores basados en acciones**

- Tema principal: pseudoclasas.
- Contenido de la lección: es posible indicar cambios en los atributos de los elementos según el comportamiento o las acciones que reciban. Estas características incluyen poseer el enfoque principal, la indicación de que un elemento ya fue enfocado, poseer el *mouse* sobre el elemento, entre otras.
- Reto: el usuario debe indicar la pseudoclase apropiada para el selector de los elementos, permitiendo que cambie sus propiedades al momento de que el usuario interactúe con el elemento para cumplir con las instrucciones del reto.

### **4.3. Arquitectura de software**

La aplicación web resultante es una *Single Page Application* (SPA), en la cual los datos se almacenarán localmente en el navegador donde se esté ejecutando la aplicación. En consecuencia, la arquitectura adoptada sigue un

modelo cliente-servidor simple. En este modelo, el navegador del usuario solicitará los archivos necesarios para ejecutar la aplicación, pero no se producirá un intercambio constante de datos con el servidor. Esto se debe a que toda la información requerida será almacenada localmente, optimizando así la eficiencia y la experiencia del usuario.

**Figura 8.**

*Componentes de la arquitectura*



*Nota.* Componentes de la arquitectura. Elaboración propia, realizado con Visio.



## 5. IMPLEMENTACIÓN

### 5.1. Despliegue de la aplicación

La aplicación ha sido implementada en un servidor web de acceso público, eligiendo Netlify como el servicio de alojamiento. Esta elección se basó en las opciones de accesibilidad y seguridad que Netlify proporciona.

Es posible acceder a la aplicación web utilizando el siguiente hipervínculo:

<https://csspirit-academy.netlify.app/>

### 5.2. Requerimientos

Para consumir la aplicación web, es necesario contar con los siguientes requisitos:

- Un navegador web actualizado capaz de ejecutar HTML5 y Javascript ECMAScript 5.
- Un mínimo de 4 GB de memoria RAM.
- Al menos 500 MB de espacio de almacenamiento disponible.
- Conexión a Internet de banda ancha.

Cumplir con estos requisitos asegura un rendimiento óptimo de la aplicación, brindando así una experiencia agradable al usuario.



## 6. MANUAL DE UTILIZACIÓN

### 6.1. Inicio del juego

Para comenzar un nuevo juego, se debe hacer clic en el botón «Iniciar» que se encuentra en la pantalla principal. En caso de ser un juego recién iniciado, el jugador será dirigido al primer nivel. En cambio, si se trata de retomar un juego previamente comenzado, el jugador iniciará en el último nivel no completado.

#### Figura 9.

*Pantalla de inicio del juego*



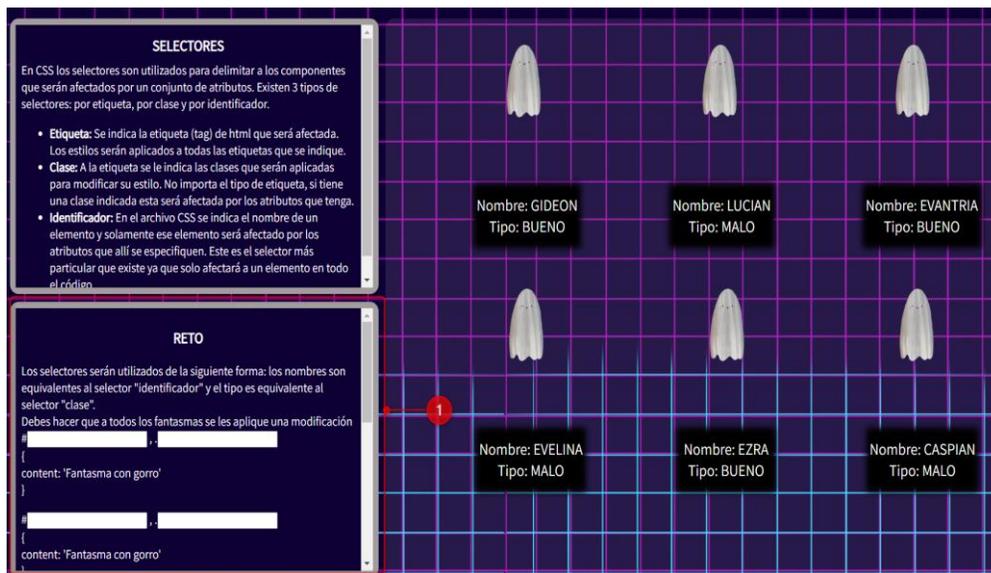
*Nota.* Pantalla de inicio. Elaboración propia, realizado con JavaScript.

## 6.2. Dinámicas en los niveles

En cada nivel es necesario superar un desafío vinculado a la lección correspondiente. La descripción detallada de este desafío se ubica en la esquina inferior izquierda de la pantalla. Para superar los retos, es preciso ingresar la información requerida en los cuadros de texto, siguiendo las instrucciones proporcionadas. Los datos ingresados afectarán directamente lo que se visualiza en la parte derecha de la pantalla.

**Figura 10.**

*Pantalla del nivel uno del juego*



*Nota.* Pantalla de nivel uno. Elaboración propia, realizado con JavaScript.

Una vez superado el desafío, es posible avanzar al siguiente nivel haciendo clic en el botón «siguiente». En este momento se verificará automáticamente si el desafío ha sido completado. Si ha sido superado, el juego mostrará la pantalla del siguiente nivel; de lo contrario, se presentará un mensaje

indicando que no es posible avanzar debido a que el desafío aún no ha sido completado.

### **6.3. Finalización del juego**

Al culminar el último nivel del juego, se presentará una pantalla notificando que el juego ha concluido y ofrecerá la opción de reiniciar. Optar por reiniciar redirigirá al jugador a la pantalla del primer nivel del juego.



## CONCLUSIONES

1. La estética y el estilo han adquirido una importancia crucial en las aplicaciones web actuales, subrayando la necesidad imperativa para los desarrolladores de adquirir habilidades en el uso de CSS como un lenguaje esencial de estilización.
2. La integración de lecciones interactivas en una aplicación se revela como una alternativa excepcionalmente efectiva para comprender los conceptos de CSS. Este método permite a los aprendices visualizar en tiempo real cómo los atributos afectan a diversos componentes HTML, ofreciendo una experiencia de aprendizaje más inmersiva y práctica.
3. CSS presenta una amplia gama de atributos que definen la disposición de los elementos HTML. Familiarizarse con todas las opciones disponibles capacita a los desarrolladores para seleccionar la más apropiada, según las necesidades específicas que buscan satisfacer, potenciando así su capacidad de diseño y maquetación.



## RECOMENDACIONES

1. Recordar que para mantener una destreza constante en CSS, al igual que con otros lenguajes empleados en el desarrollo web, es crucial practicar de manera regular. Esto garantizará que los desarrolladores no olviden los valores asociados a los diversos atributos, contribuyendo así a una implementación más fluida y precisa.
2. Tomar en cuenta que, aunque la aplicación interactiva se enfoca en propiedades específicas relacionadas con la disposición de componentes, se recomienda que los usuarios exploren y experimenten con la amplia variedad de atributos que CSS proporciona para modificar las propiedades de los elementos HTML. Esta iniciativa de investigación y prueba ampliará la comprensión del lenguaje de estilización, permitiendo a los desarrolladores abordar diversos aspectos de diseño con mayor versatilidad y precisión.



## REFERENCIAS

- Acosta, J. (4 de diciembre de 2020). *Las 7 skills que debe tener un desarrollador Front End*. OpenWebinars Blog. <https://openwebinars.net/blog/las-7-skills-que-debe-tener-un-desarrollador-front-end/>
- Barba, J. (2015). *Diseño y desarrollo web. Análisis de casos*. [Proyecto de fin de carrera, Universitat Politècnica de València]. Archivo digital. <http://hdl.handle.net/10251/49757>
- BBC News Mundo. (2019). *30 años de la World Wide Web: ¿cuál fue la primera página web de la historia y para qué servía?* <https://www.bbc.com/mundo/noticias-47524843>
- Berners-Lee, T. (1989). *Information Managment: A Proposal*. CERN.
- Colocía, C. (2015). *Importancia del diseño web adaptativo*. Unimex. <https://unimex.edu.mx/investigacion>
- Digital Talent Agency. (8 de junio de 2022). *10 habilidades y conocimientos clave de un desarrollador Front-End*. Digital Talent Agency Blog. <https://dtagency.tech/10-habilidades-y-conocimientos-clave-de-un-desarrollador-front-end/>
- Siegel, D. (1997). *Técnicas avanzadas para el diseño de páginas web*. Anaya.

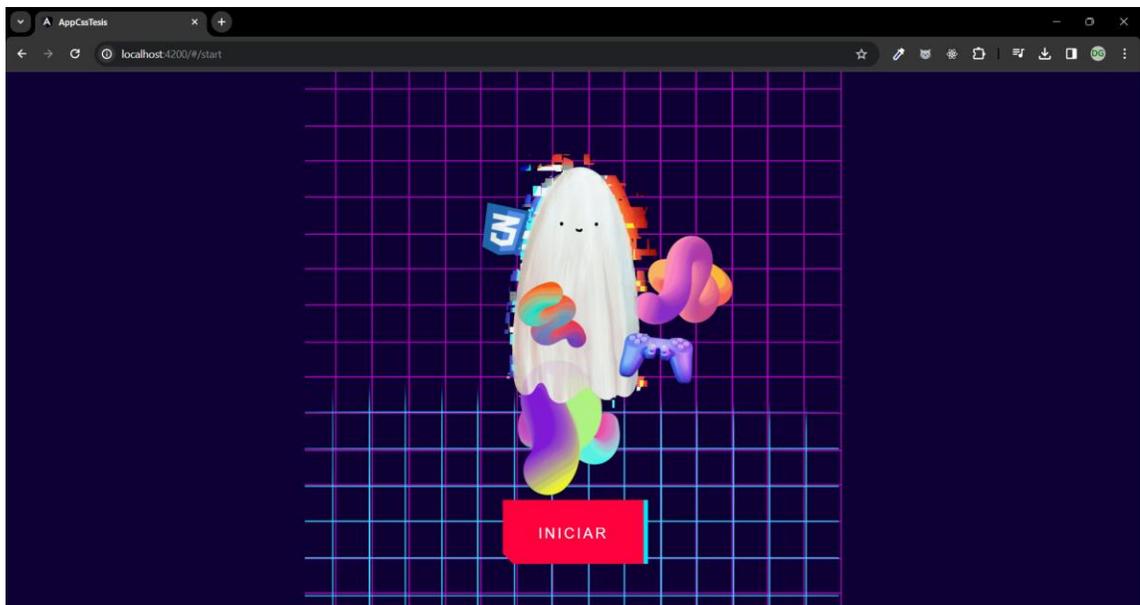
Utel. (9 de febrero de 2017). *Historia del lenguaje Javascript. Utel Blog.*  
<https://utel.mx/blog/menu-profesional/historia-del-lenguaje-javascript/>

W3 Schools. (s.f.). *Conceptos sobre atributos.*  
<https://www.w3schools.com/cssref/index.php>

# APÉNDICES

## Apéndice 1.

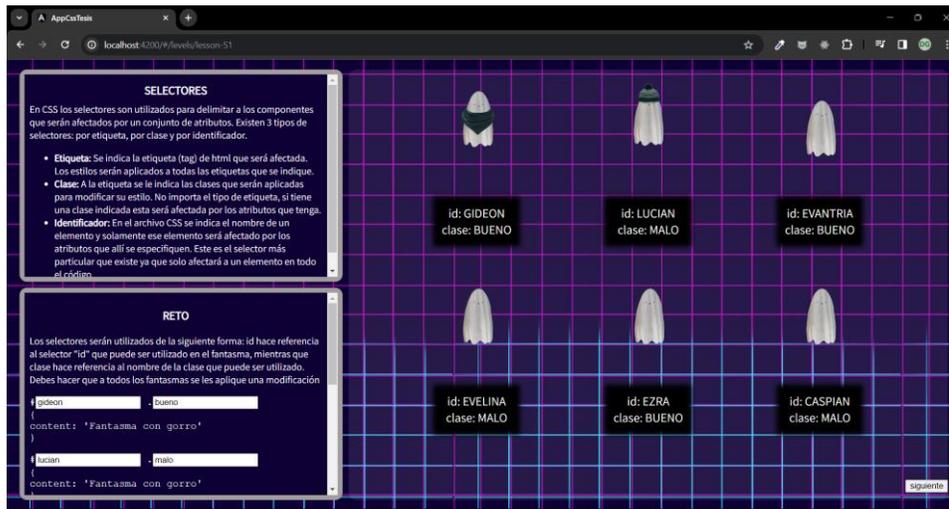
### *Pantalla inicial*



*Nota.* Pantalla inicial. Elaboración propia, realizado con JavaScript.

## Apéndice 2.

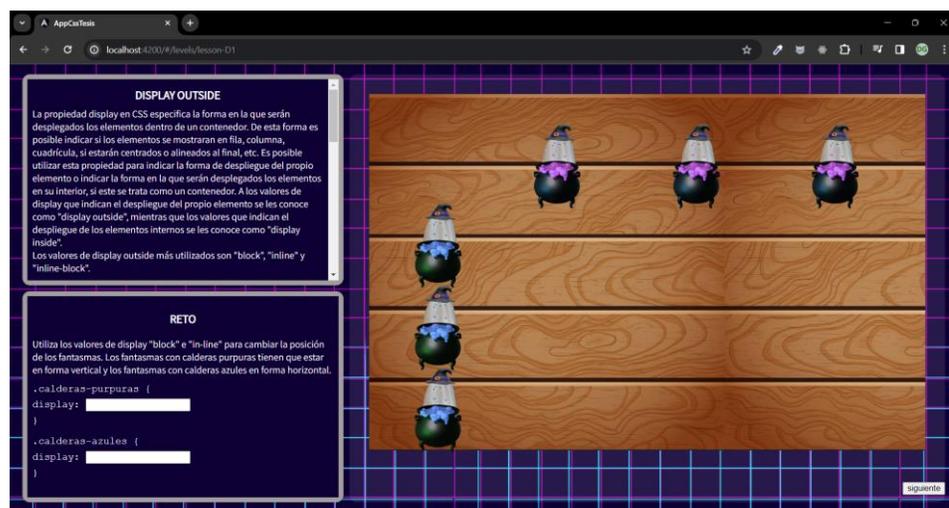
### Nivel uno del juego



Nota. Pantalla de nivel 1. Elaboración propia, realizado con JavaScript.

## Apéndice 3.

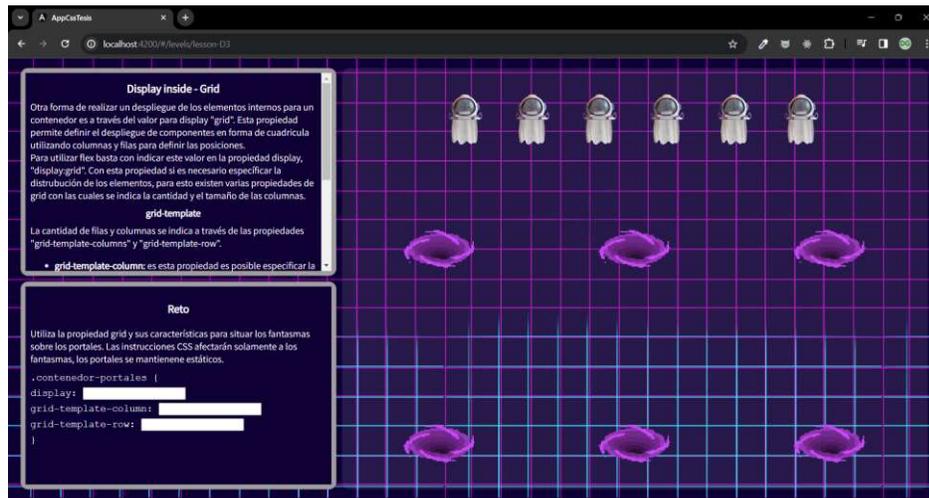
### Nivel dos del juego



Nota. Pantalla de nivel 2. Elaboración propia, realizado con JavaScript.

## Apéndice 4.

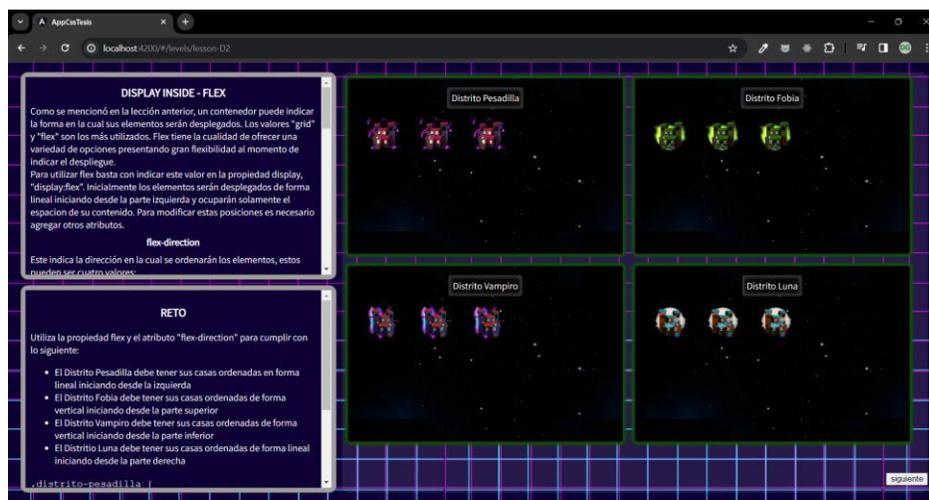
### Nivel tres del juego



Nota. Pantalla de nivel 3. Elaboración propia, realizado con JavaScript.

## Apéndice 5.

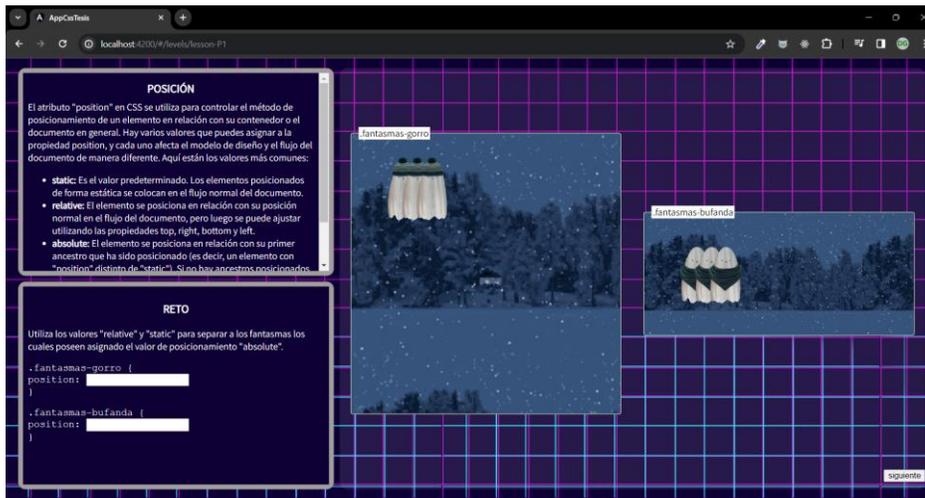
### Nivel cuatro del juego



Nota. Pantalla de nivel 4. Elaboración propia, realizado con JavaScript.

## Apéndice 6.

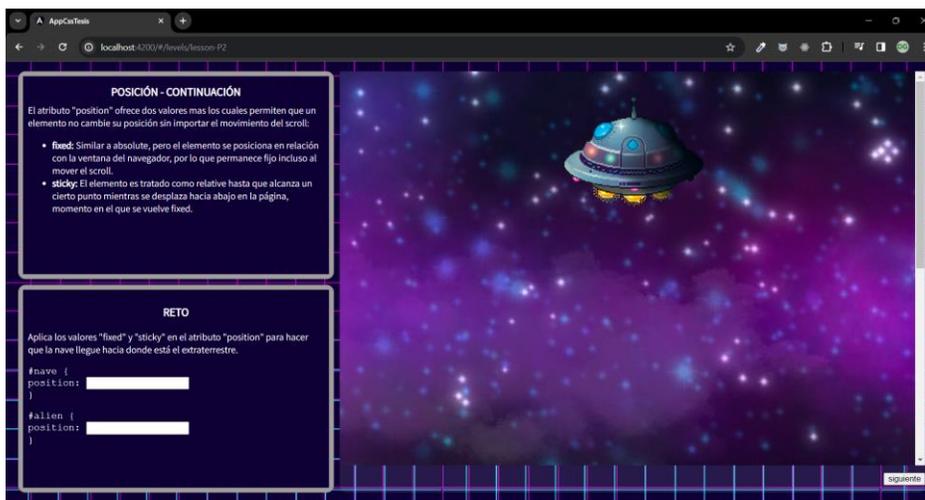
### Nivel cinco del juego



Nota. Pantalla de nivel 5. Elaboración propia, realizado con JavaScript.

## Apéndice 7.

### Nivel seis del juego



Nota. Pantalla de nivel 6. Elaboración propia, realizado con JavaScript.

