Propuesta de enseñanza virtual del Lenguaje de Marcas de Hipertexto HTML mediante un Entrenador desarrollado a la medida de la propuesta

Elfar Didier Morantes Sánchez

Universitat Oberta de Catalunya

Universidad Autónoma de Bucaramanga

Maestría E-learning

Facultad Ciencias Sociales Humanidades y Artes

Bogotá

2023

Propuesta de enseñanza virtual del Lenguaje de Marcas de Hipertexto HTML mediante un Entrenador desarrollado a la medida de la propuesta

Elfar Didier Morantes Sánchez

Trabajo de grado presentado para optar el título de Magíster en E-Learning

Jorge Andrick Parra Valencia

(Director)

Universidad Oberta de Catalunya

Universidad Autónoma de Bucaramanga

Maestría E-learning

Facultad Ciencias Sociales Humanidades y Artes

Bogotá

2023

Nota de Aceptación:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del presidente del jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma del jurado

Bogotá 2023

Dedicatoria:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Agradecimientos:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contenido

[Resumen: 7](#_Toc11189824)

[Abstract: 8](#_Toc11189825)

[Introducción: 9](#_Toc11189826)

[Estado del Arte: 13](#_Toc11189827)

[Nacimiento del HTTP y Estandarización del HTML 13](#_Toc11189828)

[Consolidación del HTML 14](#_Toc11189829)

[Complementos del HTML 16](#_Toc11189830)

[Desarrollo Web: 17](#_Toc11189831)

[Descripción de la Experiencia: 19](#_Toc11189832)

[Investigación de modelos , interfaces, y lenguajes de programación aptos para desarrollar el simulador para el aprendizaje del lenguaje de marcas HTML 20](#_Toc11189833)

[Lenguaje de programación a utilizar 20](#_Toc11189834)

[Modelo 21](#_Toc11189835)

[Interfaces 22](#_Toc11189836)

[Investigación de la forma en que serán presentados los conceptos en el simulador (figuras, formas, colores, animación, ejemplos, situaciones, organización de conceptos) 23](#_Toc11189837)

[Obtener código HTML 23](#_Toc11189838)

[HTML editable en tiempo real 25](#_Toc11189839)

[Prototipo de Sistema Drag and Drop para el Simulador 27](#_Toc11189840)

[Integración de HTML editable en tiempo real y el Prototipo de Sistema Drag and Drop para el Simulador 33](#_Toc11189841)

[Resultados Alcanzados 40](#_Toc11189842)

[Conclusiones y Recomendaciones 42](#_Toc11189843)

[Referencias bibliográficas 43](#_Toc11189844)

# Resumen:

¿Cómo facilitar el proceso de enseñanza-aprendizaje de los conceptos básicos de la programación en lenguaje de marcas HTML y superar el escollo para asimilar dichos conceptos mediante el uso de una base pedagógica en combinación con un entrenador? La propuesta de este trabajo de grado es, **por un lado, desarrollar un entrenador** que le permita al estudiante, en especial al estudiante a distancia virtual, precisar y definir los conceptos abstractos de la programación en lenguaje de marcas HTML, de manera que tenga unas bases sólidas para continuar su aprendizaje de conceptos más avanzados de forma autónoma, utilizando las facilidades, ayudas y características que le ofrece el entrenador mediante el aprendizaje activo y colaborativo. **Por otro lado, proponer la utilización de un modelo pedagógico** **que sirva de soporte y andamiaje para que el entrenador se pueda utilizar efectivamente** en la construcción de conocimientos y habilidades relacionadas con el desarrollo de páginas web. El método consiste en desarrollar un sistema para generar componentes del tipo drag and drop –arrastre y suelte-, de manera que el estudiante inicialmente no tendrá que saber cómo construir algún elemento del lenguaje de marcas HTML, sino que sencillamente lo arrastrará de un menú que contiene todos los elementos esenciales para construir formularios, y de la utilización de un modelo pedagógico que soporte y coadyuve al proceso de enseñanza-aprendizaje de desarrollo de páginas web utilizando como herramienta el entrenador desarrollado. La base del modelo pedagógico será la teoría del aprendizaje Socio Constructivista, utilizando como enfoque o modelo pedagógico el aprendizaje experiencial de la Escuela Activa o Escuela Nueva, e implementando el aprendizaje basado en problemas ABP y el aprendizaje cooperativo como métodos pedagógicos que se centran en el estudiante y promueven el aprendizaje y la socialización entre los estudiantes.

**Palabras Claves:** Entrenador, Programación, Html, Formularios, Aprendizaje a Distancia, Socio Constructivismo, Aprendizaje Experiencial, Aprendizaje Basado en Problemas, Aprendizaje Cooperativo

**Línea de investigación:** Pensamiento sistémico y educación

Abstract:  
How to facilitate the teaching-learning process of the basic concepts of programming in HTML markup language and overcome the obstacle to assimilate these concepts by using a pedagogical base in combination with a trainer? The proposal of this degree work is, on the one hand, to develop a trainer that allows the student, especially the virtual distance student, to specify and define the abstract concepts of programming in HTML markup language, so that they have a solid foundation to continue their learning of more advanced concepts autonomously, using the facilities, aids and features that the trainer offers you through active and collaborative learning. On the other hand, propose the use of a pedagogical model that serves as support and scaffolding so that the trainer can be used effectively in the construction of knowledge and skills related to the development of web pages. The method consists of developing a system to generate components of the drag and drop type, so that the student initially will not have to know how to build any element of the html markup language, but will simply drag it from a menu that It contains all the essential elements to build forms, and the use of a pedagogical model that supports and contributes to the teaching-learning process of developing web pages using the developed trainer as a tool. The basis of the pedagogical model will be the Socio-Constructivist learning theory, using the experiential learning of the Active School or New School as an approach or pedagogical model, and implementing PBL problem-based learning and cooperative learning as pedagogical methods that focus on learning and the student and promote learning and socialization among students.

**Keywords:** Trainer, Programming, Html, Forms, Distance Learning, Socio-Constructivism, experiential learning, problem-based learning PBL, cooperative learning

**Research line:** Systemic thinking and education

# Introducción:

La enseñanza de los lenguajes de programación en las carreras técnicas, tecnológicas y profesionales relacionadas con los Sistemas, la Electrónica y las Computadoras es esencial para desarrollar competencias básicas en los aspirantes a estas titulaciones. Sin embargo, la enseñanza de estos lenguajes no es una actividad que se realice de manera orgánica ni didáctica, y muchas veces se reduce a remitirse a los libros y a practicar con el código. Como lo indican algunos investigadores del campo de la enseñanza de la programación “*Ver la solución de un problema en un libro, o ver al profesor construyéndola en el tablero, no garantiza la adecuada generación de habilidades en el estudiante. Y aunque en un porcentaje significativo de los casos este enfoque funciona adecuadamente (así se ha enseñado programación durante los últimos 20 años), el problema es que su éxito depende de factores que no controlamos. (Villalobos, Casallas, Marcos 2005).”* Posiblemente estas falencias se deban a que existen muy pocas herramientas que sean intuitivas para el usuario y le permitan apropiarse cómodamente de los conceptos asociados a los lenguajes de programación. Indudablemente, si se quiere aprender a hacer pan, se tendrá que interactuar con la masa, untarse las manos, amasar, y meter al horno dicha masa para obtener resultados. Así mismo, si se quiere aprender a programar, se tiene que interactuar con el código, manipularlo, depurarlo, y modificarlo una y otra vez para obtener resultados.

No obstante, surge una pregunta: ¿Habrá una forma diferente y más entretenida de enseñar los lenguajes de programación? ¿Existirá una forma más “amigable” de acercar los conceptos de la programación que sea diferente a tratar de imaginar los conceptos abstractos presentados por un libro o por un profesor? Como lo mencionan Villalobos, Casallas, Marcos 2005 con respecto a presentar conceptos en clase o demostrar la forma de utilizar alguna estructura de programación mediante la solución de un ejercicio,  *“Dicha manera de enseñar a programar se basa en la esperanza de que el estudiante sea capaz de detectar patrones en los problemas planteados y los logre asociar con las técnicas que usa el profesor cuando desarrolla un ejemplo en el tablero y que luego, el estudiante sea capaz de hacer la generalización adecuada en su cabeza, para poder aplicar esa asociación patrón-técnica en la resolución de otros problemas (Villalobos, Casallas, Marcos 2005).”*

Evidentemente existe un vasto acervo científico y documental sobre los lenguajes de programación, y existen tantos lenguajes de programación casi como dialectos en el planeta y, por tanto, es necesario delimitar este dilema a un solo lenguaje de programación, de manera que sea viable analizar la posibilidad de construir una herramienta, simulador o entrenador que facilite la adquisición de los conocimientos relacionados con dicho lenguaje de programación. Para este caso particular, se seleccionará el lenguaje de marcas de hipertexto HTML por su gran utilidad y versatilidad en el desarrollo de páginas de Internet. En el siglo XXI la mayoría de la información se encuentra disponible en la red, y por tanto, es pertinente seleccionar el lenguaje HTML como objeto de estudio para construir un simulador o entrenador que permita facilitar el aprendizaje de las nociones asociadas a este lenguaje, y allanar el proceso de enseñanza-aprendizaje relacionado.

Quien haya tenido experiencia o relación desarrollando software en lenguajes de marcas como HTML habrá notado que este lenguaje está conformado por una gran cantidad de conceptos abstractos tales como etiquetas, botones, cajas de texto, identificadores, eventos, entre otros conceptos fundamentales de este lenguaje. Para algunos estudiantes es difícil comprender estos conceptos que, inicialmente, no tienen muchos referentes palpables, y otros estudiantes tardan un tiempo considerable en comprender la esencia de estas ideas.

El asunto se torna más interesante cuando el desarrollo de estas competencias de programación se realiza a través de la modalidad a distancia virtual. Es el aprendiz quien por sí mismo debe construir sus definiciones y conceptos a partir del material de estudio proporcionado. Evidentemente tendrá la colaboración del tutor a través de foros y otros espacios destinados para la comunicación, pero es claro que el material proporcionado al estudiante debe ser de una calidad tal que le permita al estudiante desarrollar sus facultades intelectuales y construir conocimiento. Además, otro apuro a considerar en la programación en lenguaje de marcas HTML es el proceso de editar el código, y luego visualizarlo. Normalmente este es un proceso que consta de dos partes separadas en el tiempo, pues en primer lugar se debe escribir el código con un editor de texto plano, guardar los cambios con una extensión de archivo .html, y finalmente visualizar los productos con otra herramienta. Es decir, primero se debe escribir el código con una herramienta, y luego visualizar los frutos en otra. Esto implica que el estudiante debe realizar varios pasos, normalmente secuenciales y con diferentes herramientas, antes de poder percibir algún cambio en el proyecto web que esté desarrollando. En este punto, y habiendo identificado los distintos dilemas, se podría formular la siguiente pregunta: ¿Será posible construir una herramienta que le permita al estudiante apropiarse de estos conceptos abstractos, de manera que se vuelvan concretos, precisos y palpables? ¿Será factible desarrollar un instrumento que facilite el proceso de edición y visualización de código html, de manera que el estudiante experimente simultáneamente la escritura y la percepción de la página web?

Si se define someramente un simulador como una herramienta que permite reproducir sensaciones, experiencias o comportamientos, entonces, se podría pensar en la herramienta que se quiere desarrollar para el estudiante como un simulador. En esta investigación se podrá usar indistintamente la palabra simulador y entrenador, pues el simulador permitirá reproducir experiencias similares a las reales (programando una página web), y el entrenador le permitirá seguir aprendiendo y seguir entrenándose en los campos relacionados con el desarrollo de páginas web. Pero, ¿Será posible que un simulador/entrenador le permita al estudiante acercarse más a los conceptos básicos del lenguaje de marcas HTML? ¿Podrá el simulador/entrenador reproducir la experiencia de programar, pero haciéndolo de una manera más sencilla, y quizás, hasta intuitiva y divertida?

Como lo indica Villalobos 2009, “*Como parte importante de las estrategias que se desarrollan frente al problema de enseñar a programar, se encuentran las herramientas de soporte. Principalmente herramientas de visualización y animación de algoritmos. Los instructores y profesores han encontrado gran soporte en estas herramientas cuando se mide el impacto que tienen en las sesiones de clase y laboratorios. Estas herramientas involucran a los estudiantes en diversos niveles de compromiso con el aprendizaje. (Villalobos 2009)”*. Son muchos los casos que ilustran la utilidad de los simuladores y su habilidad para reproducir ciertos comportamientos o experiencias. Por ejemplo, muchos pilotos de carreras entrenan con simuladores de autos para mejorar sus tiempos e incrementar sus habilidades. Los pilotos de aviones y helicópteros entrenan en simuladores de vuelo para reproducir ciertas experiencias críticas y de esa manera, estar preparados para eventuales situaciones. Si los simuladores han tenido éxito en estos campos, entonces se podría pensar que un simulador de programación le permitirá al estudiante reproducir la experiencia de programar en un lenguaje de marcas HTML de una manera más amigable, haciendo los conceptos más relevantes, concretos y aplicables, coadyuvando al desarrollo de las competencias orientadas a la programación, no solo como una herramienta complementaria a la formación impartida por el tutor y la información que reside en los libros, sino como un elemento de aprendizaje autocontenido que le permite al estudiante por sí mismo apropiarse de los conocimientos.

Redondeando todas estas ideas, se puede afirmar que se ha identificado una problemática: ¿Cómo facilitar el proceso de enseñanza-aprendizaje de los conceptos básicos de la programación en lenguaje de marcas HTML y superar el escollo para asimilar dichos conceptos mediante el uso de un simulador?

La propuesta de este proyecto es, por un lado, desarrollar un simulador que le permita al estudiante, en especial al estudiante a distancia virtual, precisar y definir los conceptos abstractos de la programación en lenguaje de marcas HTML, haciendo énfasis en aquellos conceptos y nociones relacionados con los formularios web – herramienta primordial para el intercambio de información entre un usuario y un servicio-, de manera que tenga unas bases sólidas para continuar su aprendizaje de conceptos más avanzados de manera autónoma, utilizando las facilidades, ayudas y características que le ofrece el simulador. La forma en que este simulador le simplificará al estudiante el proceso de apropiarse de los conceptos será porque utilizará un sistema para generar componentes del tipo drag and drop –arrastre y suelte-, de manera que el estudiante inicialmente no tiene que saber cómo construir un botón o una caja de texto en el lenguaje de marcas html, sino que sencillamente lo arrastrará de un menú que contiene todos los elementos esenciales para construir formularios. Además, en tiempo real podrá observar el código html asociado a dicho elemento, y podrá cambiar sus propiedades y verificar su comportamiento simultáneamente, sin necesidad de guardar el código, y refrescar el navegador para percibir los cambios. Inclusive el estudiante podrá ir más allá del sistema de “arrastre y suelte” y podrá seguir escribiendo código html adicional, de manera que pueda añadir nuevas características y nuevas funcionalidades a su proyecto web. Finalmente, el estudiante podrá copiar el código generado a partir de arrastrar y soltar componentes, y podrá utilizarlo en entornos de producción de la vida real.

Por otro lado, se pretende articular la propuesta de enseñanza del lenguaje de marcas de hipertexto HTML con el entrenador/simulador desarrollado, sobre un modelo pedagógico que soporte y coadyuve al proceso de enseñanza-aprendizaje de desarrollo de páginas web. Se utilizará como teoría de aprendizaje subyacente el Socio Constructivismo propuesto por Lev Siminovach Vigostky, en el que el docente toma un papel de guía o facilitador en vez de tener un papel de proveedor de contenido, y donde los estudiantes toman un rol activo en la construcción de sus propias representaciones de la realidad y propio proceso de aprendizaje, interactuando con otros y trabajando en equipo. El socio constructivismo juega un papel catalizador en el uso del entrenador/simulador desarrollado, pues permite que los estudiantes construyan sus propios conceptos entre ellos, con la ayuda de una herramienta –el entrenador/simulador- que les permitirá conocer y comprender los principios detrás de los elementos constitutivos de las páginas web. Se utilizará como enfoque o modelo pedagógico el aprendizaje experiencial de la Escuela Activa o Escuela Nueva, que se centra en los intereses espontáneos del estudiante, y fortalece su actividad, libertad y autonomía. Es así que el aprendizaje experiencial sirve como apoyo al socio constructivismo con cinco características clave: la experiencia es la base o estímulo para el aprendizaje, los alumnos construyen su propia experiencia activamente, el aprendizaje es un proceso holístico, el aprendizaje se construye social y culturalmente, el aprendizaje se ve influenciado por el contexto socio-emocional en el que se realiza. (Boud, Cohen y Walker 1993). Se utilizará el aprendizaje basado en problemas ABP como método pedagógico que se centra en el estudiante, y en el que el estudiante adquiere habilidades, conocimientos y aptitudes a través de situaciones o problemas de la vida real. También se utilizará el aprendizaje cooperativo como un método pedagógico que promueve el aprendizaje y la socialización entre los estudiantes, en el que el alumnado trabaja conjuntamente para alcanzar metas comunes, maximizando su propio aprendizaje y el de los demás miembros.

La intención es que al estudiante se le facilite el proceso enseñanza-aprendizaje de los conceptos relacionados con los formularios HTML, utilizando una única herramienta que le permita arrastrar/escribir código para componentes, verificar su funcionamiento, y realizar cambios en tiempo real, simplificando los procedimientos relacionados con la creación de los mismos, y aplicando a la vez un modelo pedagógico adecuado para la utilización de la herramienta, de manera que se puedan explotar las ventajas socio constructivistas del aprendizaje cooperativo centrado en el estudiante. Como lo indica Villalobos 2009 “*En la búsqueda de propuestas actuales frente al tema de innovar la enseñanza de la programación, muchos autores concuerdan con la necesidad de integrar la generación de habilidades más que la transmisión de conocimientos planos, especialmente en el nivel de educación superior. (Villalobos 2009)”.* Por tanto, se espera que el estudiante pueda desarrollar habilidades de creación de páginas web con el uso del simulador, más que transmitir el concepto de etiqueta, o el concepto de página web.Evidentemente, desarrollar un simulador que abarque todos los conceptos del lenguaje de marcas HTML es pretencioso y podría convertirse en una tarea compleja de realizar. Por tal razón, el simulador solamente incluirá conceptos relacionados con los formularios web, pues son la herramienta por antonomasia para la comunicación entre un ser humano y una página web, de manera que se pueda verificar la apropiación de los conceptos relacionados con la construcción de elementos HTML de una página web. Así mismo, abarcar en un proceso de enseñanza todas las teorías del aprendizaje, modelos pedagógicos, enfoques y métodos pedagógicos no sería conveniente, práctico ni útil, por lo que se propone integrar únicamente como teoría de aprendizaje subyacente el socio constructivismo apoyado en el aprendizaje experiencial como enfoque o modelo pedagógico y los métodos pedagógicos del aprendizaje basado en problemas y el aprendizaje colaborativo para desarrollar conocimiento utilizando como herramienta el simulador desarrollado.

# Estado del Arte:

## Nacimiento del HTTP y Estandarización del HTML

En 1989, Tim Berners-Lee (conocido como el padre de la World Wide Web actualmente) se encontraba trabajando en el departamento de servicios informáticos en el CERN (Conseil Européen pour la Recherche Nucléaire, Organización Europea para la Investigación Nuclear ) cuando tuvo la idea de permitir el acceso a la información de diferentes colaboradores ubicados en distintas partes del mundo, pero no para poder descargar la información, sino para que pudieran enlazar la información en los mismos archivos (Raggett; Lam; Alexander y Kmiec,1998). La idea concebida era que mientras se estaba leyendo un documento, se podría acceder a cualquier parte de otro documento. Es decir, el texto permitiría mucho más que su lectura, y serviría como una estructura organizativa que enlazaría distintos documentos. Las capacidades del texto serían hiper-desarrolladas, y por tal razón empezaría a consolidarse el nombre de hipertexto.

La palabra hipertexto no era nueva para ese entonces, sino que era un concepto académico que existía desde 1940. De hecho, la primera persona en usar esta expresión fue Ted Nelson, en su artículo “No more teacher´s dirty looks” (Nelson,1970), para referirse a escritos no secuenciales que coordinaran la presentación de cualquier tipo de información, texto, imagen o audio. Un entorno en el que el usuario pudiera interactuar con la información de varias maneras. Este concepto de hipertexto se mantiene hasta el día de hoy, como base para todas las páginas web, donde se presenta una información que está enlazada o vinculada con otra.

Después de trabajar en esta idea de documentos vinculados, Berners-Lee creó un protocolo, el Hyper Text Transfer Protocol (HTTP de aquí en adelante), que era un protocolo muy simple con el cual se podía implementar un formato de texto con marcas hiperdesarrollado (HTML de aquí en adelante), en cualquier máquina, independientemente del sistema operativo que utilizase (Davison y Chen, 1995). Este protocolo HTTP permitía la independencia de la máquina en la que se estuviese trabajando (PC,Macintosh, Unix, entre otros), y la independencia del formato de texto (SGML, Interleaf, LaTex, Word, Troff, entre otros) para implementar el formato de texto HTML y además, serviría como estándar para transferir hipertexto en las máquinas existentes de la época. Desde entonces, HTTP se consolidó como el protocolo estándar utilizado por las transacciones en internet y como el sistema mediante el cual se envían las peticiones para acceder a una página Web y mediante el cual dicha página responde. Este protocolo envía el hipertexto HTML desde el servidor donde esté alojada la página web, para que el cliente (navegador web que solicita la información) lo disponga en la pantalla del cibernauta. Habiendo solucionado el inconveniente de transmitir hipertexto a distintas máquinas sin importar su arquitectura o su tecnología de red mediante el protocolo HTTP, el lenguaje de marcas de hipertexto HTML se consolidó como el estándar para crear contenido vinculado que pudiese ser compartido, visualizado y consumido en la World Wide Web. Este lenguaje de texto con marcas HTML se basa principalmente en un sistema de etiquetas que indica al navegador (software utilizado para visualizar/consumir contenido de internet) dónde está el inicio de un documento, dónde está el cuerpo de un documento, dónde y cuándo colocar una imagen, dónde y cuándo colorear un texto, dónde está el final del documento, entre otras instrucciones.

## Consolidación del HTML

El lenguaje HTML posee distintas características que hicieron que fuese fácilmente aceptado por la comunidad y la industria para estructurar las páginas web, sin embargo, un factor trascendental para facilitar dicha aceptación fue sin duda que su base y cimientos estaban en otro lenguaje de hipertexto, como era “Standard Generalized Mark-up Language” (de aquí en adelante SGML). SGML había sido inventado en 1979 por Charles Goldfarb y era un lenguaje de hipertexto que funcionaba en cualquier máquina reconocido internacionalmente. La idea central detrás de SGM era que el lenguaje fuese independiente del “formateador” (el navegador u otro software) que dispusiera la información en la pantalla. Este lenguaje introducía el concepto de separar la estructura del contenido, de su presentación (Cailliau, 1995). De esa forma, HTML se volvió popular al estar basado en un lenguaje versátil ya existente. También incidió en la popularización del lenguaje HTML la acción de su creador de divulgar su producto y fomentar la discusión abiertamente alrededor de toda la World Wide Web. Para 1992, unos cuantos académicos e informáticos habían mostrado interés por HTML y por su mejora, entre ellos Dave Raggett, de los laboratorios de Hewlett-Packard. Después de unas cuantas discusiones electrónicas Raggett visitó a Berners-Lee en 1992 en el CERN y discutieron sobre cómo convertir HTML en un “producto de masas”, procurando definir las características que un usuario pudiera ver como útiles e importantes. En el transcurso de 1992 muchos entusiastas del novedoso lenguaje de hipertexto se reunirían en torno al grupo de discusión recientemente creado, con el nombre de WWW. Entre ellos estarían Dave Raggett, Tim Berners-Lee , Dan Connolly entre otros muchos más, y con el tiempo, este grupo de discusión daría origen al Consorcio WWW, en inglés World Wide Web Consortium (W3C), que es un consorcio internacional que genera recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo. Gracias al esfuerzo de investigadores como Dan Connolly, en 1993 se recopilan la mayoría de las etiquetas que eran usadas por los navegadores que surgen entre 1993 y 1994, de manera que se pudiese estandarizar aún más el lenguaje HTML que estaba teniendo auge por esos días. Connolly recoge la mayoría de las sugerencias de entusiastas de HTML y las integra en un documento que hace circular por la red, preparando un borrador de lo que más tarde se conocería como HTML2.

Durante 1995, nuevos tipos de etiquetas de HTML surgen y crean bastantes discrepancias sobre su utilización, y así, en el transcurso de ese año, surge la primera versión de HTML3, publicada como un borrador por Dave Raggett, quien había estado trabajando en mejorar las características y etiquetas de HTML. También en 1995 aparece un nuevo elemento en el campo de los navegadores web. Microsoft lanza la primera versión de Internet Explorer, con la intención de competir con los navegadores populares de la época (Netscape) y desarrollar sus propias características de HTML. La entrada de Microsoft populariza aún más el uso del lenguaje HTML, y se convertiría más tarde en una de las organizaciones que haría parte del grupo de estandarización del lenguaje, junto con IBM, Novell y Netscape.

A finales de 1996 el HTML Working Group empezará a trabajar en la nueva versión de HTML, llamada “Cougar”, que será la base del HTML 4. Esta versión incluirá innovaciones para personas con discapacidades, soporte internacional para diferentes lenguajes (no solo aquellos con caracteres latinos), hojas de estilo en cascada, extensiones, scripting y otros.

En 2004, se crea un grupo denominado “WHATWG” con miembros de Apple, la fundación Mozilla, y Opera Software para empezar a desarrollar el estándar HTML5. En 2006 el W3C decide detener su trabajo de XHTML y empieza a trabajar con el “WHATWG” para evolucionar HTML como una tecnología. En 2008 sale a la luz el primer borrador de HTML5, escrito por Ian Hickson, indicando que HTML5 seguiría evolucionando como tecnología continuamente y que nunca estaría completamente terminada

## Complementos del HTML

En 1995 Bert Bos, Hakon Lie, Dave Raggett, Chris Lilley y otros miembros del W3C, se reunían en Versalles, para discutir el desarrollo de un lenguaje especial para escribir estilos en los documentos HTML: las hojas en cascada o CSS (“Cascading Style Sheets”, de aquí en adelante CSS). Este lenguaje CSS describiría el diseño visual de los documentos web, y cómo los elementos HTML deberían ser mostrados y desplegados por el navegador. CSS se diseñó principalmente para marcar la separación del contenido del documento y la forma de presentación de este, es decir, para separar características tales como las capas o layouts, los colores y las fuentes. Esta separación de la “forma” y del “contenido” genera mejor accesibilidad, más flexibilidad y control, y además permite que múltiples páginas (dentro de un mismo sitio Web) compartan el mismo formato y estilo, reduciendo así la complejidad y reutilizando la estructura del contenido. Otra de las principales características de las hojas de estilo en cascada CSS, es que permite especificar una prioridad para determinar qué regla se aplica y en qué secuencia, en caso de que haya algún conflicto de órdenes (de ahí proviene el término “cascada” para explicar la jerarquía de las instrucciones a ejecutar).

En 1996 Dave Raggett propone otro borrador de trabajo, basándose en un borrador inicial de Charlie Kindel y el uso de las extensiones de Netscape para JavaScript,en el cual se aborda el tema del “Scripting”. Este documento se convertiría en parte del estándar HTML y abriría las puertas para los lenguajes de scripting en HTML. Un lenguaje de scripting o script es un pequeño lenguaje de programación cuyo código se inserta dentro del documento HTML y que se ejecuta en el navegador del usuario cuando ocurre un evento, como por ejemplo al cargar la página, o cuando se pulsa sobre un enlace. Estos lenguajes de scripting permiten variar dinámicamente el contenido del documento, modificar el comportamiento normal del navegador, y realizar distintas operaciones como validar formularios, y cambiar aspectos visuales, entre otros, es decir, le proporciona más dinamismo y funcionalidad al lenguaje HTML. El primer lenguaje de script fue el JavaScript de Netscape. Nació con la versión 2.0 de este navegador de la mano de Brendan Eich y basó su sintaxis en Java (un lenguaje de programación para aplicaciones de escritorio). Su popularidad y uso extenso se debió al casi absoluto monopolio que entonces ejercía Netscape en el mercado de navegadores web de aquella época. La idea de Netscape era que este lenguaje atrajera a programadores “amateurs” y que completase al lenguaje Java original.

## Desarrollo Web:

Actualmente existe una gran cantidad de lenguajes de programación para realizar desarrollo web, que define la creación de sitios web para Internet o una intranet. Este desarrollo hace uso de tecnologías de software del lado del servidor (Server) y del cliente (Browser) que involucran una combinación de procesos y sistemas de gestión de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información. Además, han surgido distintas ocupaciones relacionadas con la construcción de sitios web , como por ejemplo el desarrollador web (web developer), que es quien se preocupa por el funcionamiento del software, el diseñador web (web designer), que es quien se preocupa del aspecto final(layout) de la página y el webmaster, quien es el encargado de integrar ambas partes. En ocasiones el webmaster también se encarga de actualizar los contenidos de la página. De la mano con estas ocupaciones también han aparecido distintos Frameworks, que son un conjunto estandarizado de conceptos, prácticas, criterios, herramientas y entornos de trabajo que facilitan o simplifican el proceso de desarrollo de software y que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. Por si fuera poco, el desarrollo de un sitio web puede separar la capa de presentación de la información o capa de interacción con el usuario (Front End), y la capa de acceso a los datos (Back End), que es la capa que procesa los datos ingresados desde el Front End. Esta separación ayuda a mantener las partes del sistema separadas, simplificando así el mantenimiento, depuración, y ampliación del sitio web. Así las cosas, es claro que construir un sitio web no es una cosa tan simple, pues existen distintas tecnologías, prácticas, y frameworks involucrados, y todos ellos tiene que ver en mayor o menor medida con el lenguaje de marcas HTML.

*Socio Constructivismo*

El socio constructivismo y la teoría socio cultural (TS) formulada por el psicólogo ruso Lev Siminovach Vigostky en la década del 40 señala que el individuo necesita de un contexto socio cultural para poder alcanzar un desarrollo adecuado. Indica que todo aprendizaje tiene un origen social y que los individuos necesitan de un ambiente colaborativo que les permita adquirir las capacidades para adaptarse a su entorno. Como lo indica Anama 2020 “*El Socioconstructivismo es una teoría del aprendizaje social, y postula que el sujeto se apropia de su realidad y la dota de significado, siempre y cuando su entorno le provea las herramientas necesarias. Se considera que el aprendizaje es algo colectivo, no algo aislado; esto debido a que siempre es necesario contar con otro para alcanzar el desarrollo, ya que éste cumple una función de guía u orientador en el procedimiento (Anama 2020).*” Hay varios conceptos relacionados con el socio constructivismo como *la zona de desarrollo próximo* (ZDP) considerado como un espacio imaginario entre el conocimiento que ya se posee y el conocimiento que se podría llegar a alcanzar, y que es posible atravesar esta zona de desarrollo próximo con ayuda de otro que le provea el *andamiaje* necesario, entendiéndose *andamiaje* como el proceso que se desarrolla entre el docente y el alumno para que este último alcance la *interiorización*. La *interiorización* es el resultado que obtiene el estudiante después de atravesar la zona de desarrollo próximo gracias al andamiaje recibido. Como cita Arcila 2010 las palabras de Vygotski “*el aprendiz se apodera del conocimiento que estaba tratando de alcanzar, y de esta manera puede ponerlo en práctica para cambiar su entorno(Arcila 2010)*”.

*Aprendizaje Experiencial*

El aprendizaje experiencial como enfoque o modelo pedagógico indica que el conocimiento se crea a través de las experiencias. En otras palabras, señala que las personas tienen la capacidad de aprender de su propia experiencia, dentro de un marco conceptual y operativo bien estructurado y concreto. El aprendizaje experiencial se estructura en ciclos que consideran la experiencia como algo de carácter subjetivo y que responden a diversos modelos que permiten entender cómo se desarrolla el proceso de aprendizaje. En la década del 70 el teórico educacional David A. Kolb ayudó a desarrollar la teoría del aprendizaje experiencial, basándose en los trabajos de John Dewey, Kurt Lewin, y Jean Piaget. Por ejemplo, el ciclo de aprendizaje experiencial del Kolb tiene sus bases en la idea de que el aprendizaje experiencial es inductivo, es decir que los participantes llegan a sus propias conclusiones sobre la experiencia, y que tiene cuatro etapas: Experiencia Concreta, Observación Reflexiva, Conceptualización Abstracta y Experimentación Activa. En palabras de Espinar 2020 “*Esta teoría señala las ventajas para percibir y referir el proceso de aprendizaje, fundamentado en la práctica a partir de los sentidos, que son estímulos exteriores que proporcionan experiencias que encaminan el esfuerzo y la experticia de las habilidades (Espinar2020*)”

*Aprendizaje Basado en Problemas ABP*

El aprendizaje basado en problemas es un método pedagógico de enseñanza-aprendizaje centrado en el estudiante que señala que el aprendiz adquiere conocimientos y habilidades a través de situaciones de la vida real. La finalidad es formar individuos con capacidad analítica que puedan enfrentarse a problemas de la misma manera en que lo harían durante su actividad profesional. El ABP utiliza problemas como punto de partida para la adquisición de conocimientos nuevos y es el estudiante el protagonista de su propio proceso de aprendizaje. Se fundamenta en la base constructivista de que conocer y aprender implican una experiencia de construcción anterior. El principal rol del docente es el de facilitador de actividades de reflexión que ayuda a pensar críticamente los temas y un catalizador de la investigación y el descubrimiento. En la década del 70, el médico y educador Howard Barrows Barrows propuso que se les debería permitir a los estudiantes involucrarse con nueva información en el contexto de la resolución de problemas clínicos auténticos. Los postulados de Barrows establecieron las bases de lo que conocemos como el aprendizaje basado en problemas, y tal y como lo indican Barrows y Tamblyn 1980 “*en el caso del ABP se presenta primeramente el problema, se identifican las necesidades de aprendizaje, se busca la información necesaria y finalmente se regresa al problema a fin de resolverlo (Barrows y Tamblyn, 1980)”.* El ABP busca que el estudiante interiorice la solución del problema y que explore distintos campos de manera integral, pues como lo afirma Poot-Delgado 2013 “*El ABP incluye el desarrollo del pensamiento crítico en el mismo proceso de enseñanza-aprendizaje; no lo incorpora como algo adicional, sino que es parte del mismo proceso de interacción para aprender. Busca que el alumno comprenda y profundice adecuadamente en la respuesta a los problemas que se usan para aprender, abordando aspectos de orden filosófico, sociológico, psicológico, histórico, práctico y demás, haciendo todo lo anterior con un enfoque integral.( Poot-Delgado 2013)”*

Aprendizaje Cooperativo

El aprendizaje cooperativo es un método pedagógico que propicia la enseñanza a través de la interacción y socialización de los estudiantes. El proceso consiste en dividir el alumnado en grupos pequeños que trabajen entre si de forma coordinada resolviendo tareas o problemas, poniéndose de acuerdo sobre las tareas que se deben realizar y decidir cómo y quién realizará las diferentes actividades. Como lo señala Domingo 2008 ***“****El Aprendizaje Cooperativo permite que los estudiantes trabajen de forma independiente y asuman responsabilidades en su propio proceso de aprendizaje. También promueve el desarrollo de la capacidad para razonar de forma crítica y facilita el desarrollo de la habilidad para escribir con claridad. (Domingo 2008)” Es así que tanto la teoría de aprendizaje subyacente del socio constructivismo, como el modelo o enfoque pedagógico del aprendizaje experiencial, como el método pedagógico del aprendizaje basado en problemas tienen sus bases en el constructivismo. Y el aprendizaje cooperativo no es la excepción, pues como loa afirma Ovejero 1990 citado por Garcia 2001 “El constructivismo, al estudiar la interacción social y la construcción del conocimiento a través de los procesos interactivos, también puede considerarse como antecedente del aprendizaje cooperativo. Téngase en cuenta que los fundamentos de este van a estar directamente relacionados con la interacción social y la construcción social de la inteligencia. (García 2001)”*

# Descripción de la Experiencia:

El plan de trabajo que se ha venido desarrollando, y que fue presentado en la versión de proyecto definitivo es el siguiente:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Actividades** | **Semanas** | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| Investigación de modelos , interfaces, y lenguajes de programación aptos para desarrollar el simulador para el aprendizaje del lenguaje de marcas HTML | X | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| Investigación de la forma en que serán presentados los conceptos en el simulador (figuras, formas, colores, animación, ejemplos, situaciones, organización de conceptos) |  |  |  |  |  |  | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |
| Formato de presentación del sistema drag and drop para el aprendizaje del lenguaje de marcas HTML |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  | |
| Investigación de las teorías de aprendizaje subyacentes, modelos o enfoques pedagógicos, y métodos pedagógicos a utilizar junto con el simulador | X | X | X | X | X | X | X | X | X | X | X | X | X |  |  |  |  |  |  |  |  |  |  |  |  | |
| Formato de presentación del sistema de edición-visualización simultáneo para el aprendizaje del lenguaje de marcas HTML |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X |  |  |  |  |  |  |  |  |  | |
| Desarrollo del simulador de acuerdo al modelo, lenguaje de programación, interfaces y formato de presentación seleccionados |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X | X | X | X |  |  |  |  | |
| Implementación del Simulador en LMS para probar flexibilidad y portabilidad |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  |  | |
| Diseño de la encuesta que evaluará la eficacia del simulador implementado |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |  | |
| Evaluación del desempeño del simulador implementado como herramienta de aprendizaje al curso seleccionado |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |

## Investigación de modelos , interfaces, y lenguajes de programación aptos para desarrollar el simulador para el aprendizaje del lenguaje de marcas HTML

Como se puede observar, la primera parte que involucra 6 semanas se denomina “Investigación de modelos , interfaces, y lenguajes de programación aptos para desarrollar el simulador para el aprendizaje del lenguaje de marcas HTML”.

### Lenguaje de programación a utilizar

En esta fase se determinó que el lenguaje de programación que se utilizará para realizar el desarrollo será JAVASCRIPT, pues proporcionará dinamismo y una fácil integración con el lenguaje de marcas HTML. Además, al utilizarse JAVASCRIPT nativo, será más simple la integración de la solución con cualquier sitio web o plataforma LMS, sin necesidad de instalar lenguajes de programación adicionales o frameworks. JAVASCRIPT es fácilmente interpretado por casi todos los navegadores web, lo que redunda en que la solución podrá ser visualizada en cualquier sistema, sin importar su plataforma o sistema operativo (Windows, Linux, Macintosh, Unix, Android, etc). Es importante mencionar que casi todos los sitios web utilizan JavaScript por ser un lenguaje que ya está integrado en los navegadores (no hay que instalar nada) y este lenguaje es uno de los más solicitados en los avisos de búsqueda de empleos para programadores. A continuación se muestran algunas de las ventajas más destacadas de JavaScript:

* Es un lenguaje muy sencillo de aprender con respecto a otros lenguajes de programación web (php, python, asp)
* Es rápido (interpretado), por lo tanto tiende a ejecutar el código casi que inmediatamente.
* Es soportado por los navegadores más populares y es compatible con los dispositivos más modernos, incluyendo tablets, móviles y consolas de videojuegos
* Se puede utilizar para la validación de datos en un formulario del lado del cliente.
* Es multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo ya sea de escritorio (Windows, Linux, Macintosh) o móvil (Android , IOS).
* Es el único lenguaje que permite trabajar modo FullStack en cualquier tipo de desarrollo de programación (JS, NodeJS, VueJS).

Como se puede observar, JAVASCRIPT tiene varias ventajas que lo hacen el candidato idóneo para desarrollar el simulador que facilite la enseñanza virtual del lenguaje de marcas HTML, haciendo énfasis en los formularios.

### Modelo

Se determinó que el modelo para desarrollar el software será el patrón de arquitectura MVC, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. El MVC define la construcción de tres componentes distintos que son el Modelo, la Vista y el Controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en la noción de reutilización de código y la separación de conceptos, buscando así facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

El Modelo es la representación de la información con la cual el sistema opera, por lo tanto gestiona los accesos a dicha información tales como consultas, actualizaciones, entre otros. También implementa los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

El Controlador responde a eventos o acciones del usuario y realiza peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos, o mostrar alguna información). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

La Vista presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.

### Interfaces

La interfaz gráfica que se utilizará consiste en una ventana dividida inicialmente en tres columnas. La primera de ellas poseerá los controles y elementos HTML que responderán al sistema drag and drop con respecto a la segunda columna. Es decir, se pretende que en la primera columna se encuentren los componentes arrastrables, y en la segunda columna se puedan soltar, de manera que en la tercera columna se pueda visualizar el código html asociado.

|  |  |  |
| --- | --- | --- |
| COMPONENTES HTML  Estos componentes permitirán ser arrastrados, de manera que sean soltados en el área de trabajo | AREA DE TRABAJO  En el área de trabajo se sueltan los componentes y se organizan visualmente de acuerdo a los requerimientos del usuario | AREA DE CÓDIGO  En el área de código se visualizará el código HTML generado, de acuerdo a la organización y componentes proporcionada en el área de trabajo. |

Tabla1. Interfaz gráfica propuesta

Esta interfaz además se actualizará en tiempo real cada vez que se suelte un componente en el área de trabajo, de manera que el usuario no tenga que refrescar el navegador para visualizar los cambios. Es decir, una de las ventajas del simulador es que el estudiante vaya observando en tiempo real el resultado de manipular los componentes en el área de trabajo. Así mismo, se tiene pensado que si el estudiante tiene experiencia utilizando el lenguaje de programación, utilice el área de código para que modifique los componentes a través instrucciones, y pueda visualizar los resultados de la manipulación en el área de trabajo. En otras palabras, el área de trabajo y el área de código serán interdependientes.

## Investigación de la forma en que serán presentados los conceptos en el simulador (figuras, formas, colores, animación, ejemplos, situaciones, organización de conceptos)

En esta segunda parte que involucra 4 semanas se definió la forma en que el simulador iba a trabajar, haciendo pruebas de código, y validando la elección del lenguaje de programación utilizado.

### Obtener código HTML

En primer lugar, se investigó la forma en que se podría obtener tanto el contenido como el código de un objeto HTML presionando un botón. La idea era que se pudiese comprobar que era posible extraer el contenido HTML de cualquier objeto y que este pudiese ser visualizado en pantalla. El código presentado a continuación utiliza JAVASCRIPT, específicamente su librería JQUERY, para realizar realiza dicha tarea:

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

alert($("#texto").text());

});

$("#btn2").click(function(){

alert($("#texto").html());

});

});

</script>

</head>

<body>

<div id="texto"><input type="text" placeholder="prueba">Este texto <b>esta en negrita</b></div>

<button id="btn1">Solo Texto</button>

<button id="btn2">HTML</button>

</body>

</html>

Código 1. Página web simple con una caja de texto y dos botones

El resultado de ejecutar dicho código en un navegador web cualquiera es el siguiente:

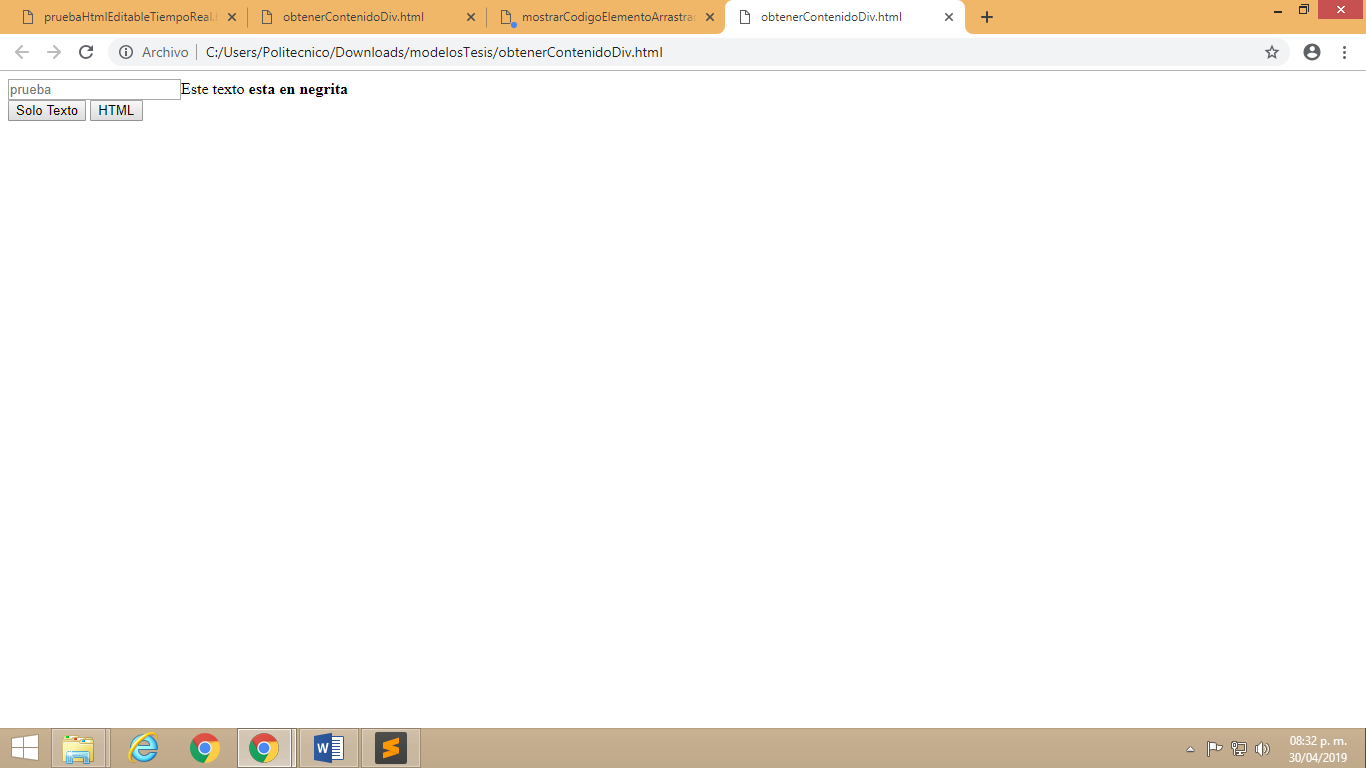


Figura 1. Prueba de código para visualización de HTML

Es decir, se observa una caja de texto con la palabra “prueba” , y seguidamente una frase que dice “Este texto está en negrita”. Debajo de ellos se encuentra un botón que dice “Solo Texto” y otro que dice “HTML”. Lo interesante del asunto es que la caja de texto que dice “prueba” y la frase que dice “Este texto está en negrita” se encuentran dentro de un contenedor HTML denominado “<div>”. Lo que se quiere hacer con el código enmarcado dentro de las etiquetas <script> </script> es que cuando se presiona el botón denominado “Solo Texto”, aparezca en pantalla un mensaje con el contenido del elemento <div>, en este caso la frase “Este texto está en negrita”, y que cuando se presione el botón denominado “HTML”, aparezca en pantalla un mensaje con el código HTML que se encuentre dentro del elemento <div>, en este caso, los siguientes comandos HTML : <input type="text" placeholder="prueba">Este texto <b>esta en negrita</b>.

En la siguiente imagen se puede visualizar el resultado de presiona el botón “Solo Texto”,

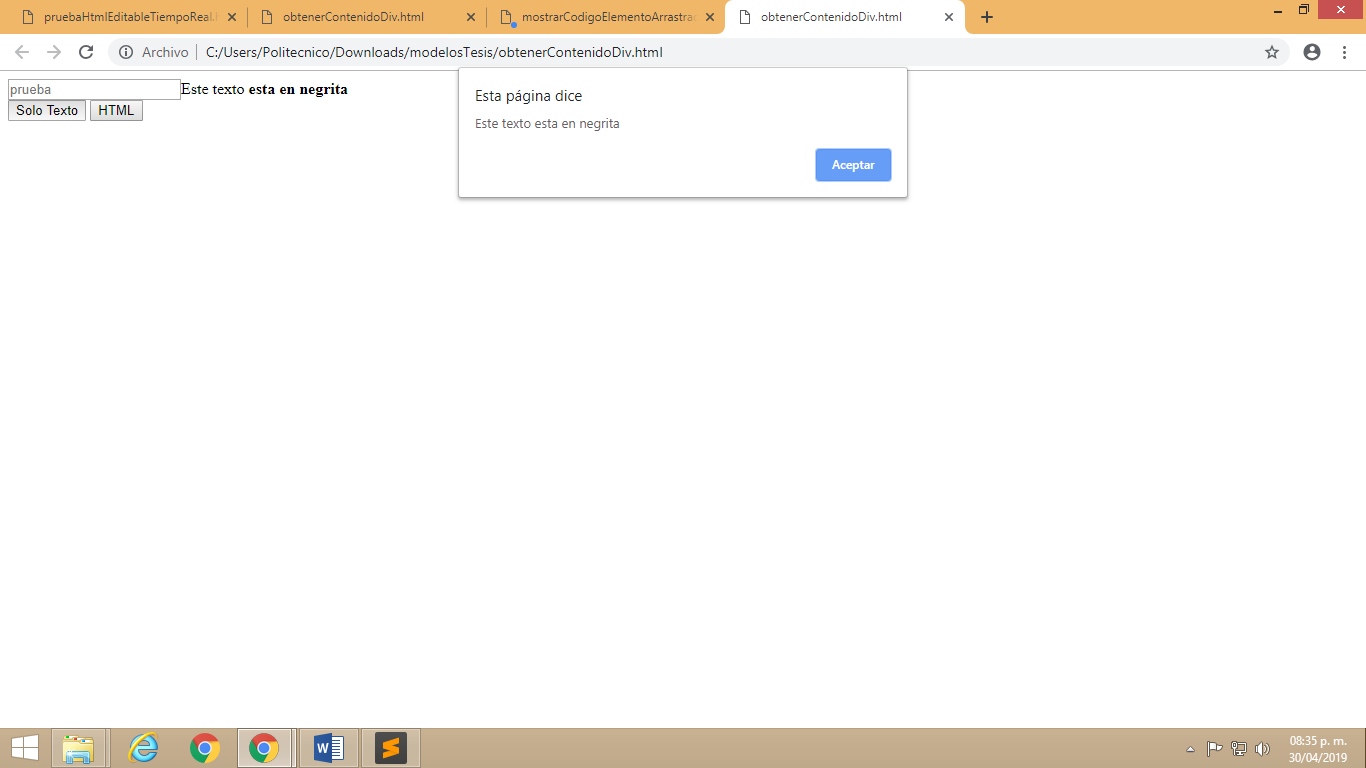


Figura 2. Resultado de presionar el botón “Solo Texto”

En la siguiente imagen se puede visualizar el resultado de presionar el botón “HTML”.

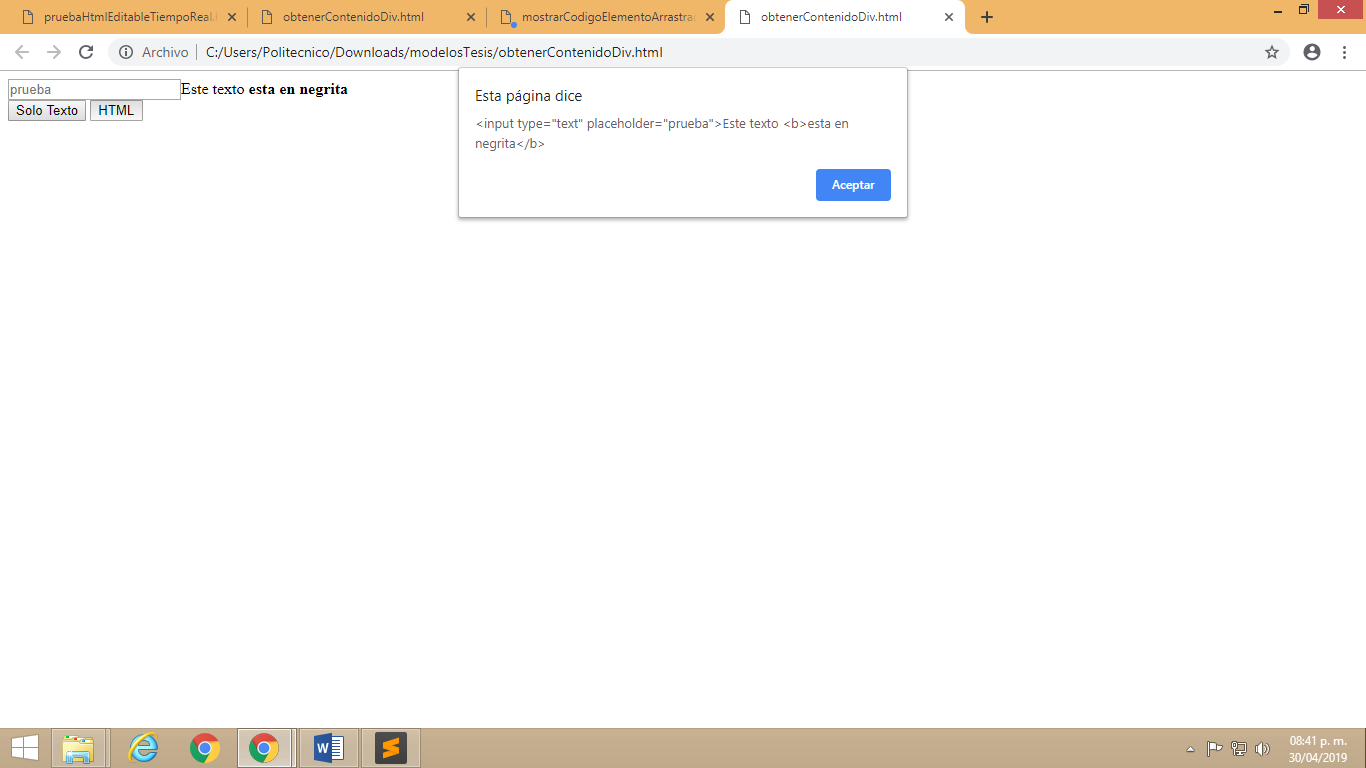


Figura 3. Resultado de presionar el botón “HTML”

Como se puede observar en la Figura 3, es posible utilizar un lenguaje de programación como JAVASCRIPT mediante su librería JQUERY para obtener tanto el código como el contenido de un objeto HTML (en este caso el contenedor <div>), y por tanto, se puede hacer extensible este conocimiento para empezar a desarrollar el simulador, específicamente la sección de “Área de Trabajo” y “Área de Código” descritas en la Tabla 1.

### HTML editable en tiempo real

Una segunda prueba que se realizó fue crear un contenedor que permitiera el ingreso de código HTML y que en tiempo real fuese ilustrando la materialización de dicho código en el navegador, ya sea para crear un botón, una caja de texto, o cualquier componente HTML. La idea era que se pudiese comprobar que era posible escribir código HTML en un contenedor, y poder visualizar en tiempo real el componente o elemento que se estuviese describiendo en dichas instrucciones. El código presentado a continuación utiliza JAVASCRIPT, específicamente su librería JQUERY, para realizar realiza dicha tarea:

<html>

<head></head>

<body>

<table>

<tr>

<td id="real"></td>

</tr>

</table>

<textarea id="txt"></textarea>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$('#txt').keyup(function(){

$('#real').html($(this).val());

});

});

</script>

</body>

</html>

Código 2. Página web simple con contenedor editable de HTML

El resultado de ejecutar dicho código en un navegador web cualquiera es el siguiente:

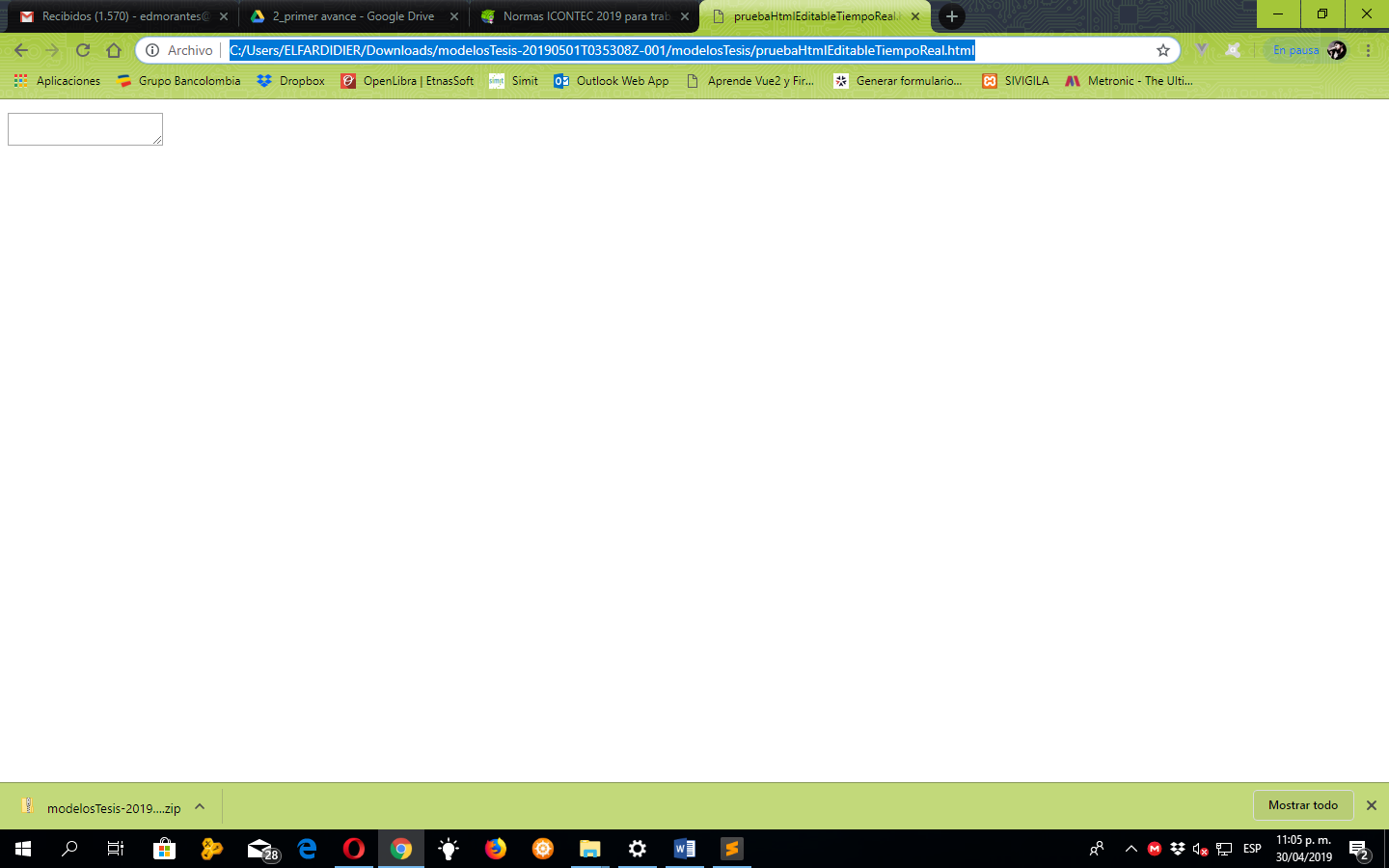


Figura 4. Prueba de código para visualización de HTML en tiempo real

Como se puede observar en la figura 4, solamente se visualiza un contenedor con la propiedad de poder escribirse en él. Evidentemente como no se ha escrito nada, no se puede visualizar su utilidad. Pero si se empiezan a escribir sentencias HTML, dicho contenedor comenzará a responder, y por tanto, se podrán visualizar los componentes HTML que se describan en el contenedor. Por ejemplo, si se escribe el código correspondiente a la creación de un botón en lenguaje HTML dentro de este contenedor, podremos ver dicho botón en nuestra web tras finalizar la escritura del último carácter correspondiente a la sentencia: <input type= “button” value= “enviar”>, como se muestra en la Figura 5.

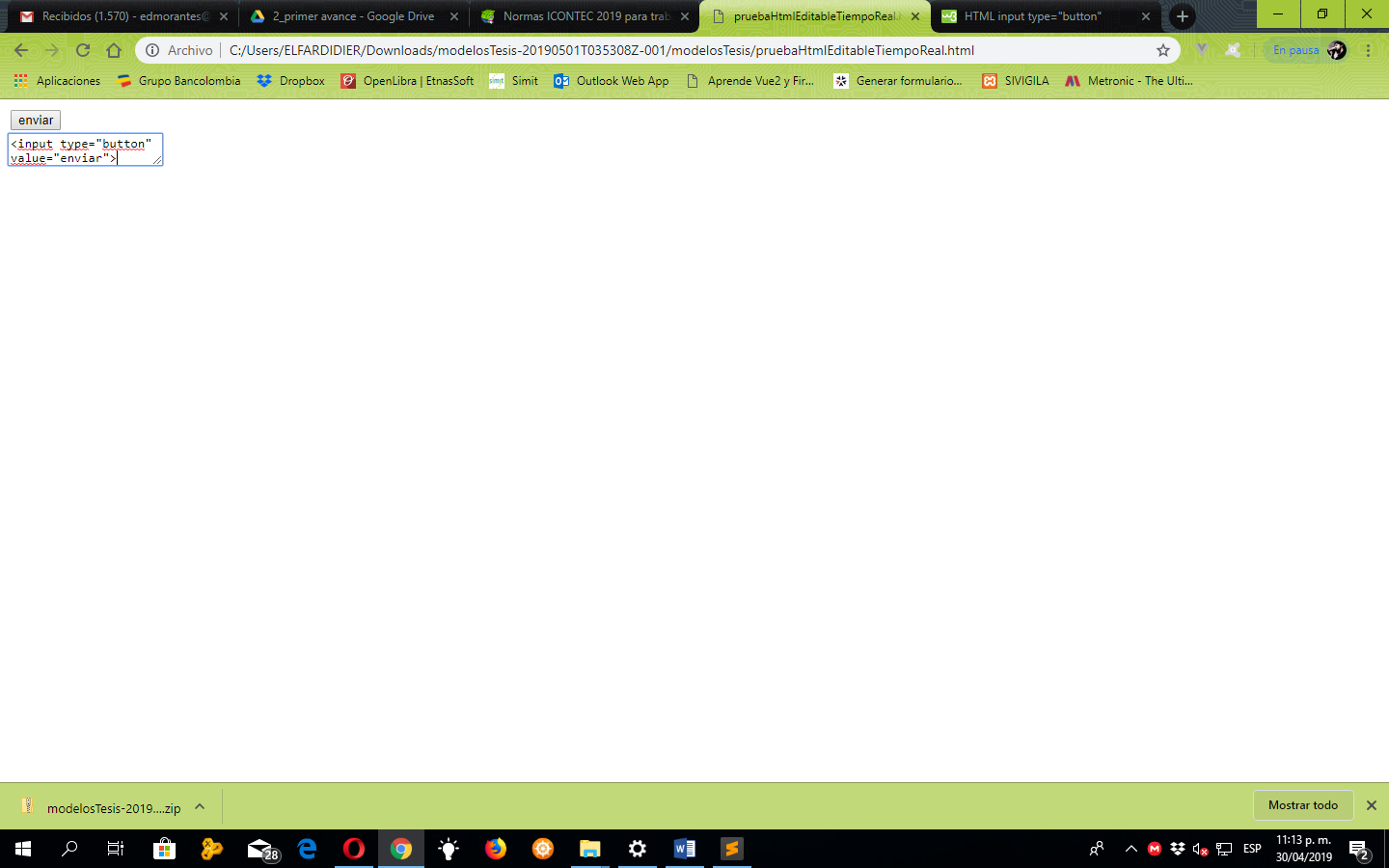


Figura 5. Resultado de escribir código HTML en el contenedor editable

Es de anotar que si no se escriben correctamente las sentencias HTML dentro del contenedor, no se podrá visualizar nada, ya que dicho contenedor se ciñe fuertemente a la sintaxis del lenguaje de marcas HTML. Como se puede observar en la Figura 5, es posible utilizar un lenguaje de programación como JAVASCRIPT mediante su librería JQUERY para generar un contenedor reactivo que presente elementos HTML en la página web en la medida en que se vaya escribiendo correctamente el código, y por tanto, se puede hacer extensible este conocimiento para empezar a desarrollar el simulador, específicamente la sección de “Área de Código” descrita en la Tabla 1.

### Prototipo de Sistema Drag and Drop para el Simulador

Una tercera prueba realizada fue crear un prototipo que plasmara la idea de la Tabla 1, de tener tres secciones diferenciadas, en las cuales se pudiese comprobar de manera minimalista las funcionalidades del simulador propuesto. La idea era que se pudiese comprobar que era posible construir un sistema de secciones interdependientes que respondieran al sistema de arrastrar y soltar (Drag and Drop), de manera que desde la sección 1 se pudiesen tomar componentes y alojarlos en la sección 2, y el código de dicho componente se mostrara en la sección 3. El código presentado a continuación utiliza JAVASCRIPT, específicamente su librería JQUERY, para realizar realiza dicha tarea:

<!DOCTYPE HTML>

<html>

<head>

<style>

#div1, #div2 , #div3{

float: left;

width: 400px;

height: 450px;

margin: 10px;

padding: 10px;

border: 1px solid black;

}

</style>

<script>

function allowDrop(ev) {

ev.preventDefault();

}

function drag(ev) {

ev.dataTransfer.setData("text", ev.target.id);

}

function drop(ev) {

ev.preventDefault();

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

</script>

</head>

<body>

<h2>Drag and Drop</h2>

<p>Drag the image back and forth between the two div elements.</p>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">

<img src="yopGenial.jpg" draggable="true" ondragstart="drag(event)" id="drag1" width="400" height="400">

<input type="button" value="prueba" draggable="true" ondragstart="drag(event)" id="drag2" >

<select draggable="true" ondragstart="drag(event)" id="drag22">

<option value="seiya">Seiya</option>

<option value="shiryu">Shiryu</option>

<option value="hyoga">Hyoga</option>

<option value="shun">Shun</option>

<option value="ikki">Ikki</option>

</select>

<input type="text" placeholder="prueba dos" draggable="true" ondragstart="drag(event)" id="drag3" >

<label draggable="true" ondragstart="drag(event)" id="drag4" >

<select style="pointer-events: none" onclick="console.log(event)" >

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</select>

</label>

<label draggable="true" ondragstart="drag(event)" id="drag5" >

<input type="submit" style="pointer-events: none" onclick="console.log(event)">

</label>

</div>

<div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<div id="div3">

<table>

<tr>

<td id="real"></td>

</tr>

</table>

<textarea id="txt" cols="50" rows="20"></textarea>

<div id="log">

<table>

<tr>

<td id="real"></td>

</tr>

</table>

</div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$('#div2').mouseleave(function(){

var htmlString= ($(this).html());

alert(htmlString);

var htmlParaCajaTexto=document.createTextNode(htmlString);

$( "#txt" ).append(htmlParaCajaTexto) ;

});

});

</script>

</body>

</html>

Código 3. Versión preliminar y minimalista del simulador

El resultado de ejecutar dicho código en un navegador web cualquiera es el siguiente:

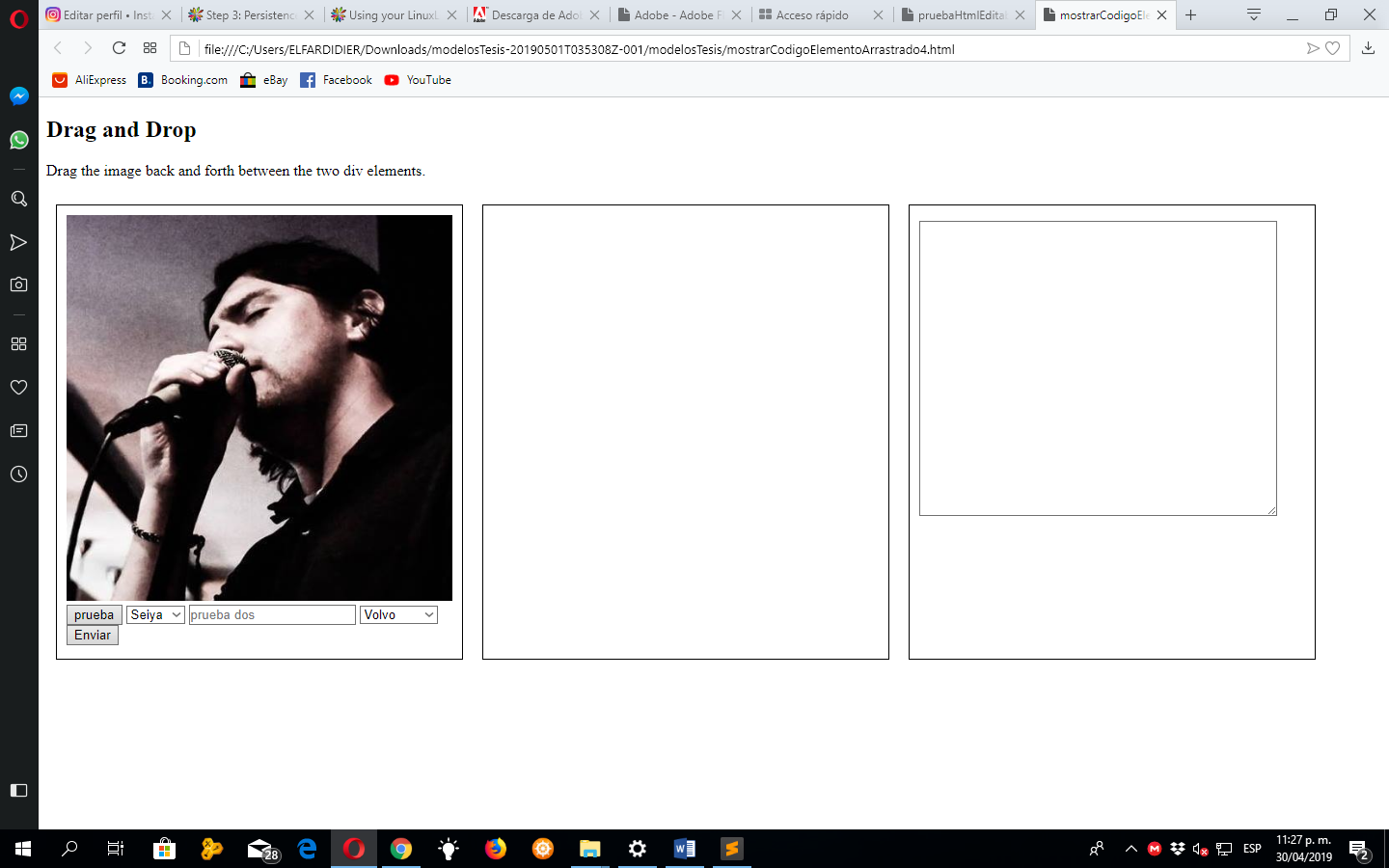


Figura 6. Prueba de código de simulador en su versión minimalista y preliminar

Como se puede observar en la figura 6, encontramos las tres secciones descritas en la Tabla 1, y en el área de componentes HTML encontramos diversos elementos, como una imagen, unos botones, y unas cajas de texto, entre otros. La prueba resulta interesante cuando se toma cualquier elemento del área de componentes HTML y se suelta en el área de trabajo (segunda sección). En la figura 7 se observa el resultado de realizar dicha operación.

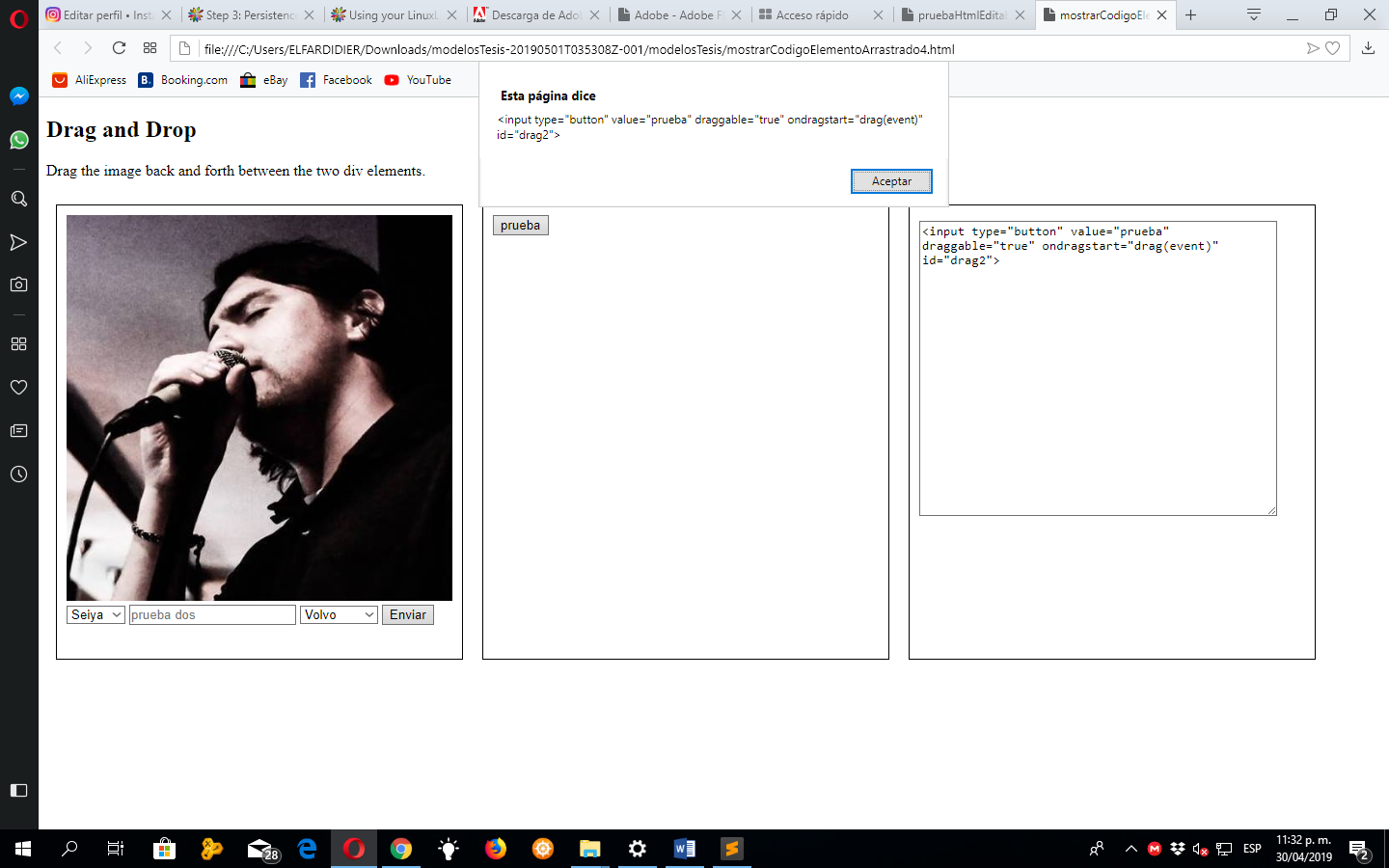


Figura 7. Resultado de probar el prototipo del simulador

Como se puede observar en la figura 7, se ha arrastrado un botón del área de componentes HTML en el área de trabajo, y ha aparecido el código de dicho componente en el área de código. De esta forma se espera que trabaje el simulador, donde el estudiante neófito en el campo del lenguaje HTML empieza a arrastrar componentes en el área de trabajo, y puede visualizar el código correspondiente a dicho componente en el área de código. Evidentemente esta es una versión preliminar del simulador, pero es una aproximación a lo que se quiere lograr. Que el estudiante comience a relacionar en tiempo real el código de cada componente que decida arrastrar al área de trabajo, y si quiere modificar dicho componente a través del código, podrá visualizar sus cambios en el área de trabajo. Se trata de construir una herramienta intuitiva que le permita al estudiante apropiarse de los conceptos HTML asociados con los formularios.

### Integración de HTML editable en tiempo real y el Prototipo de Sistema Drag and Drop para el Simulador

Una cuarta prueba realizada fue tratar de integrar el prototipo de sistema Drag and Drop para el simulador, con el módulo de HTML editable en tiempo real. La idea era que se pudiese comprobar que era posible construir la interdependencia de la sección de código (sección 3) con la sección donde se sueltan los componentes (sección 2). El código presentado a continuación utiliza JAVASCRIPT, específicamente su librería JQUERY, para realizar realiza dicha tarea:

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<style>

#div1, #div2 , #div3{

float: left;

width: 400px;

height: 450px;

margin: 10px;

padding: 10px;

border: 1px solid black;

}

</style>

<script>

function allowDrop(ev) {

ev.preventDefault();

}

function drag(ev) {

ev.dataTransfer.setData("text", ev.target.id);

}

function drop(ev) {

ev.preventDefault();

var data = ev.dataTransfer.getData("text");

ev.target.appendChild(document.getElementById(data));

}

</script>

</head>

<body>

<h2>Arrastre y suelte</h2>

<p>arrastre y suelte la imagen entre el primer y segundo recuadro.</p>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">

<img src="yopGenial.jpg" draggable="true" ondragstart="drag(event)" id="drag1" width="400" height="400">

<input type="button" value="prueba" draggable="true" ondragstart="drag(event)" id="drag2" >

<select draggable="true" ondragstart="drag(event)" id="drag22">

<option value="seiya">Seiya</option>

<option value="shiryu">Shiryu</option>

<option value="hyoga">Hyoga</option>

<option value="shun">Shun</option>

<option value="ikki">Ikki</option>

</select>

<input type="text" placeholder="prueba dos" draggable="true" ondragstart="drag(event)" id="drag3" >

<label draggable="true" ondragstart="drag(event)" id="drag4" >

<select style="pointer-events: none" onclick="console.log(event)" >

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</select>

</label>

<label draggable="true" ondragstart="drag(event)" id="drag5" >

<input type="submit" style="pointer-events: none" onclick="console.log(event)">

</label>

</div>

<div id="div2" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<div id="div3">

<!-- inicio del codigo html -->

<textarea id="txt" cols="45" rows="20"></textarea>

</div>

<script>

$(document).ready(function(){

$('#div2').mouseleave(function(){

var htmlString= ($(this).html());

alert(htmlString);

var htmlParaCajaTexto=document.createTextNode(htmlString);

$( "#txt" ).html(htmlString) ;

});

$('#txt').keyup(function(){

$('#div2').html($(this).val());

});

});

</script>

</body>

</html>

Código 4. Integración del módulo de HTML editable y el prototipo del simulador

El resultado de ejecutar dicho código en un navegador web cualquiera es el siguiente:

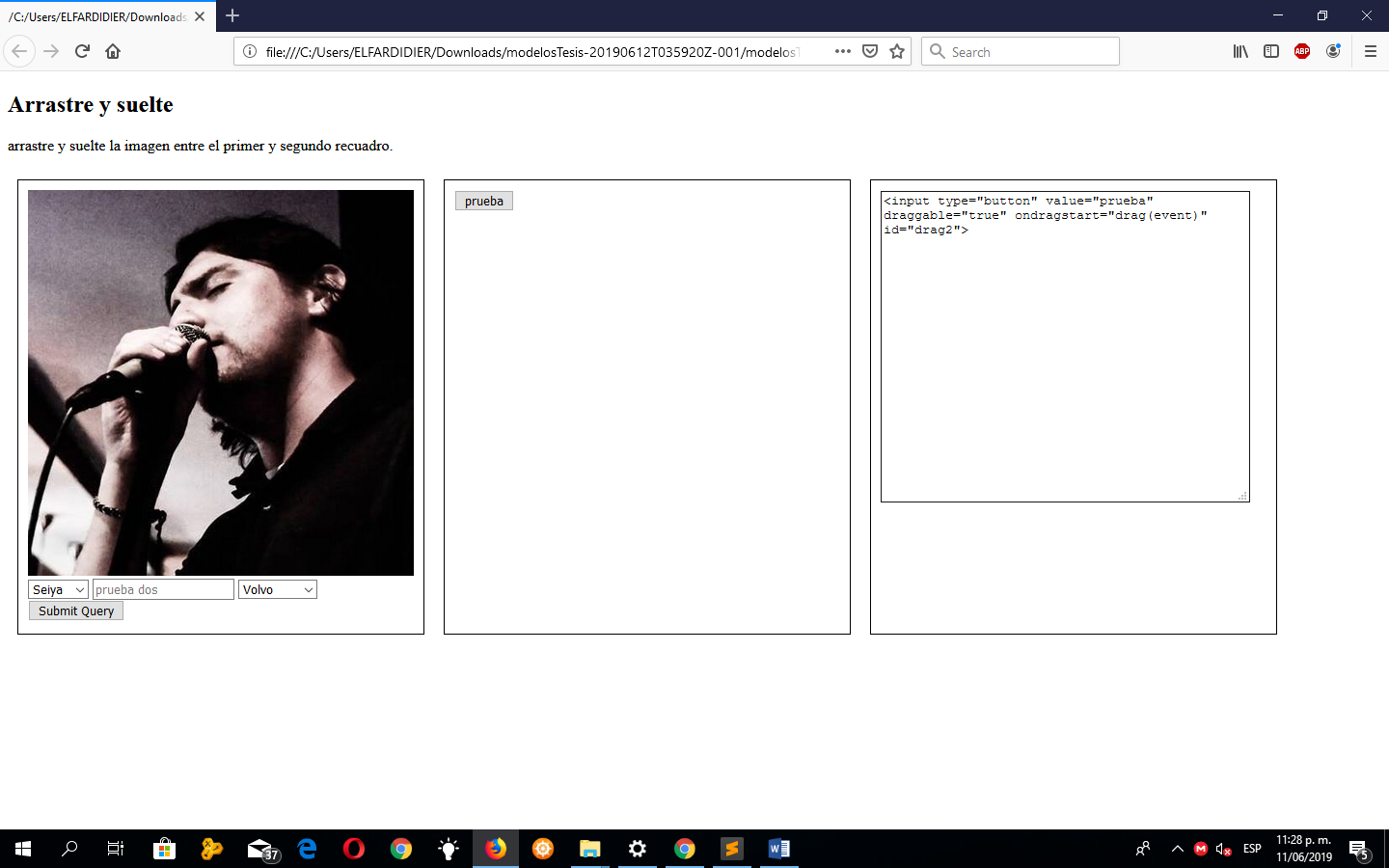


Figura 8. Resultado de probar el prototipo del simulador integrado con el módulo de HTML editable

Como se puede observar en la figura 8, al arrastrar un componente del área de componentes (sección 1) hacia el área de trabajo (sección 2), encontramos el código correspondiente en el área de código (sección 3). Si arrastramos varios componentes del área de componentes (sección1), encontraremos su código correspondiente en el área de código (sección 3), como se muestra en la figura 9.

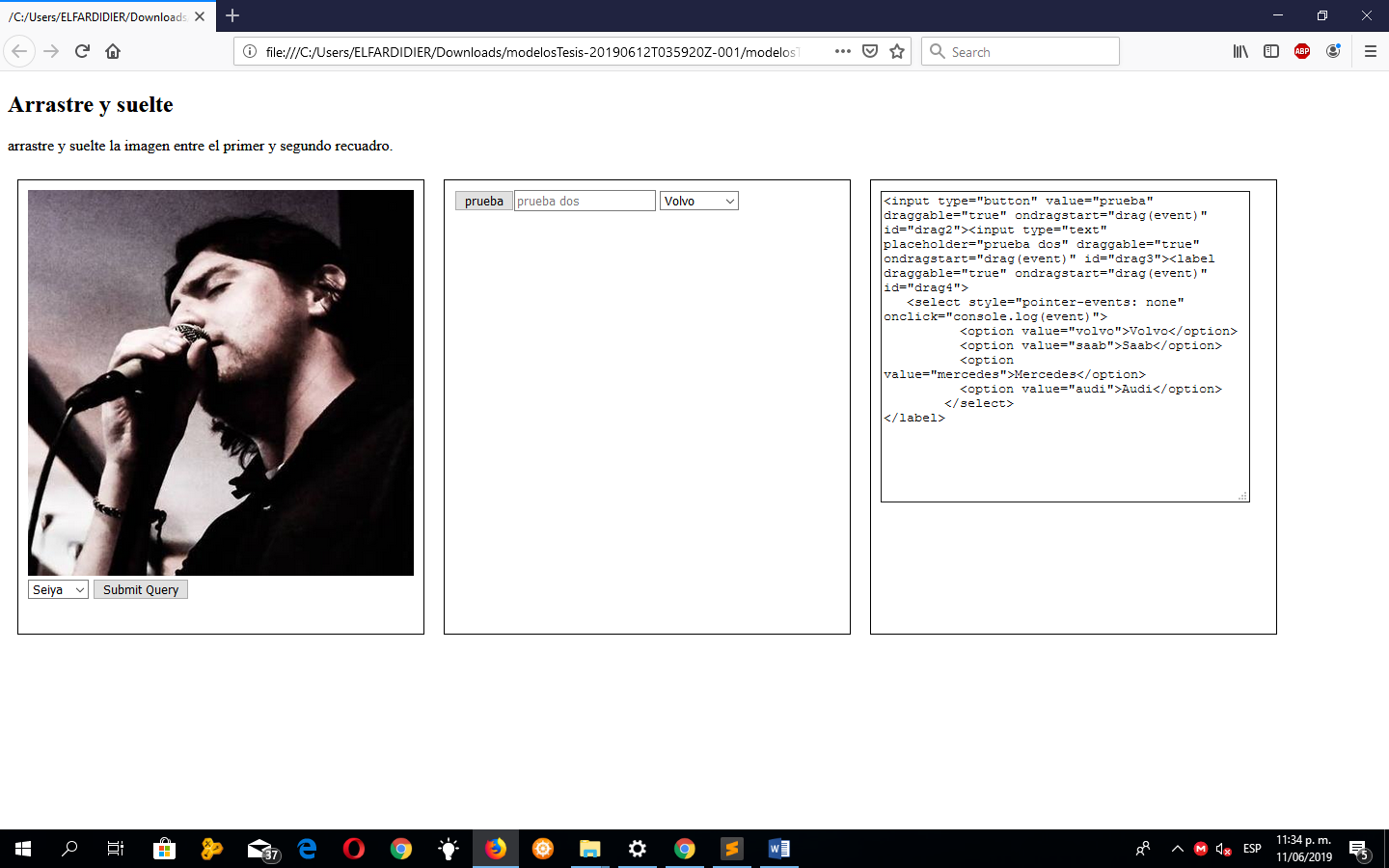


Figura 9. Varios componentes arrastrados al área de trabajo

Si se retiran componentes del área de trabajo (sección 2), inmediatamente se eliminarán las instrucciones del área de código (sección 3), como se observa en la figura 10.

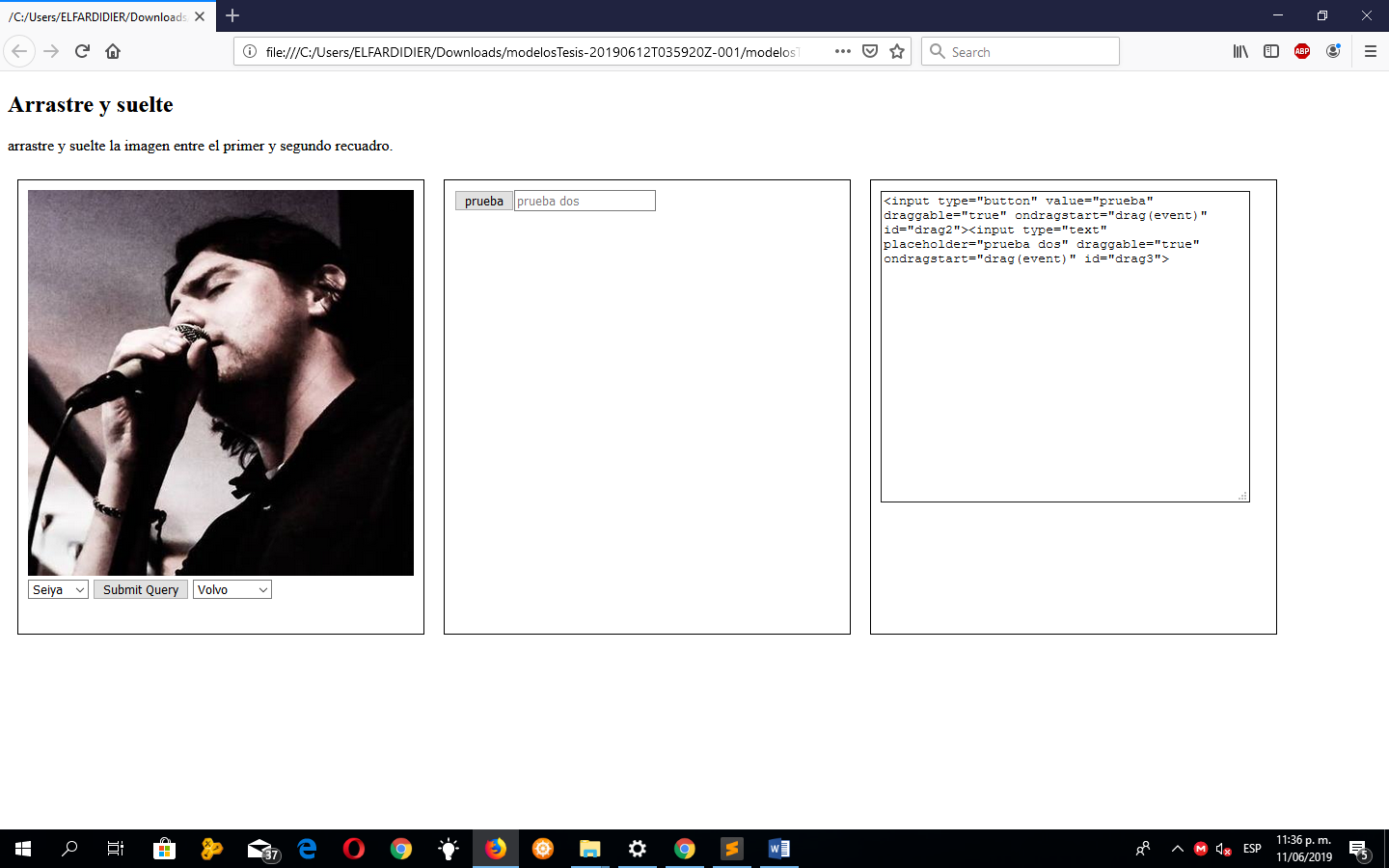


Figura 10. Componentes retirados del área de trabajo.

A continuación, se procede a escribir código adicional en el área de código (sección 3), y visualizar si es posible generar automáticamente contenido en el área de trabajo (sección 2), como se observa en la figura 11.

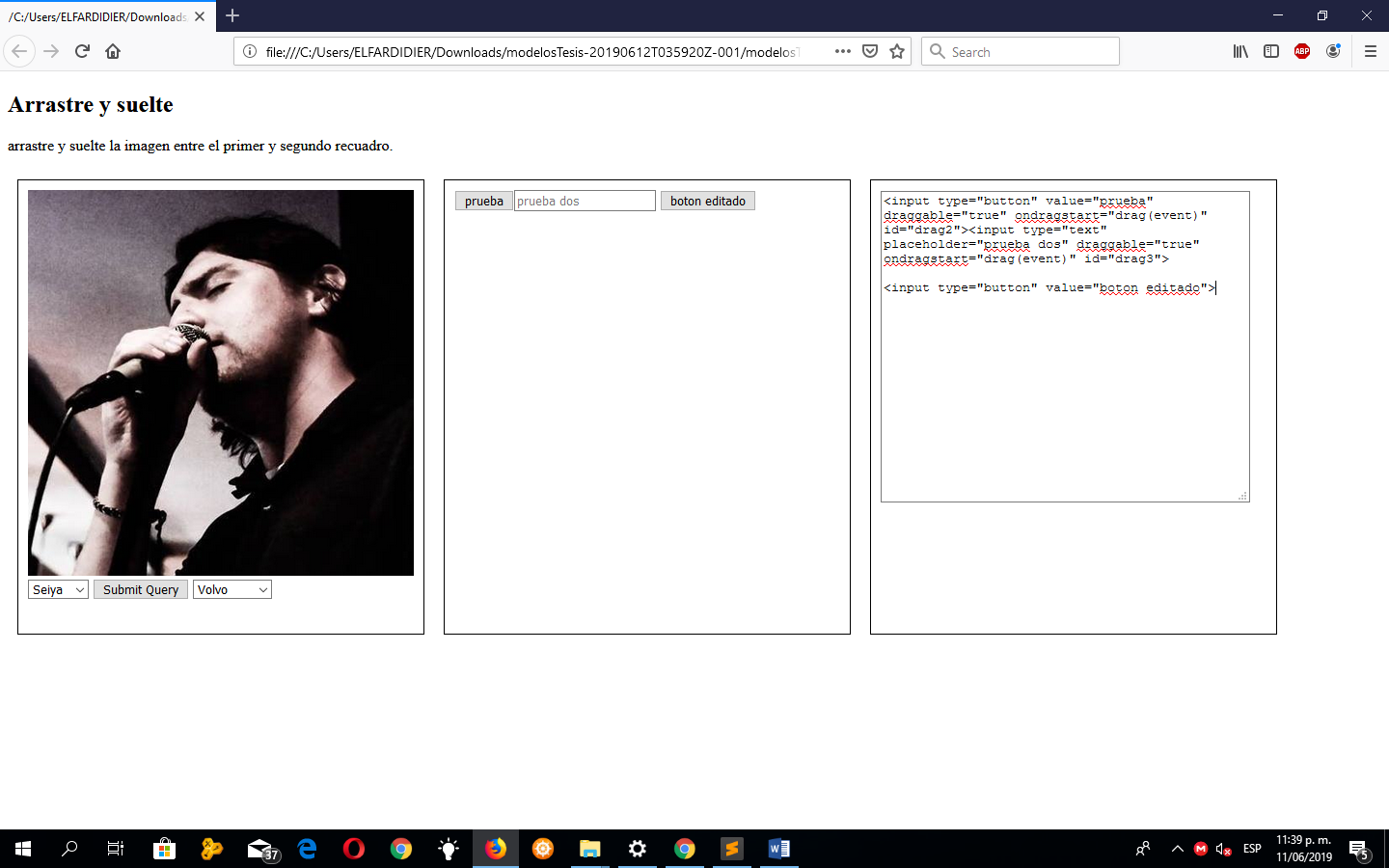


Figura 11. Escritura de instrucciones en el área de código y visualización en el área de trabajo

Como se puede observar en la figura 11, al escribir instrucciones HTML en el área de código (sección 3), inmediatamente aparece el componente en el área de trabajo (sección 2) siempre y cuando la sintaxis del código HTML sea la adecuada. Sin embargo, en este punto se encuentra que cuando se quiere arrastrar un nuevo elemento del área de componentes (sección 1) al área de trabajo (sección 2), no se actualiza la información en el área de código (sección 3), como se muestra en la figura 12.

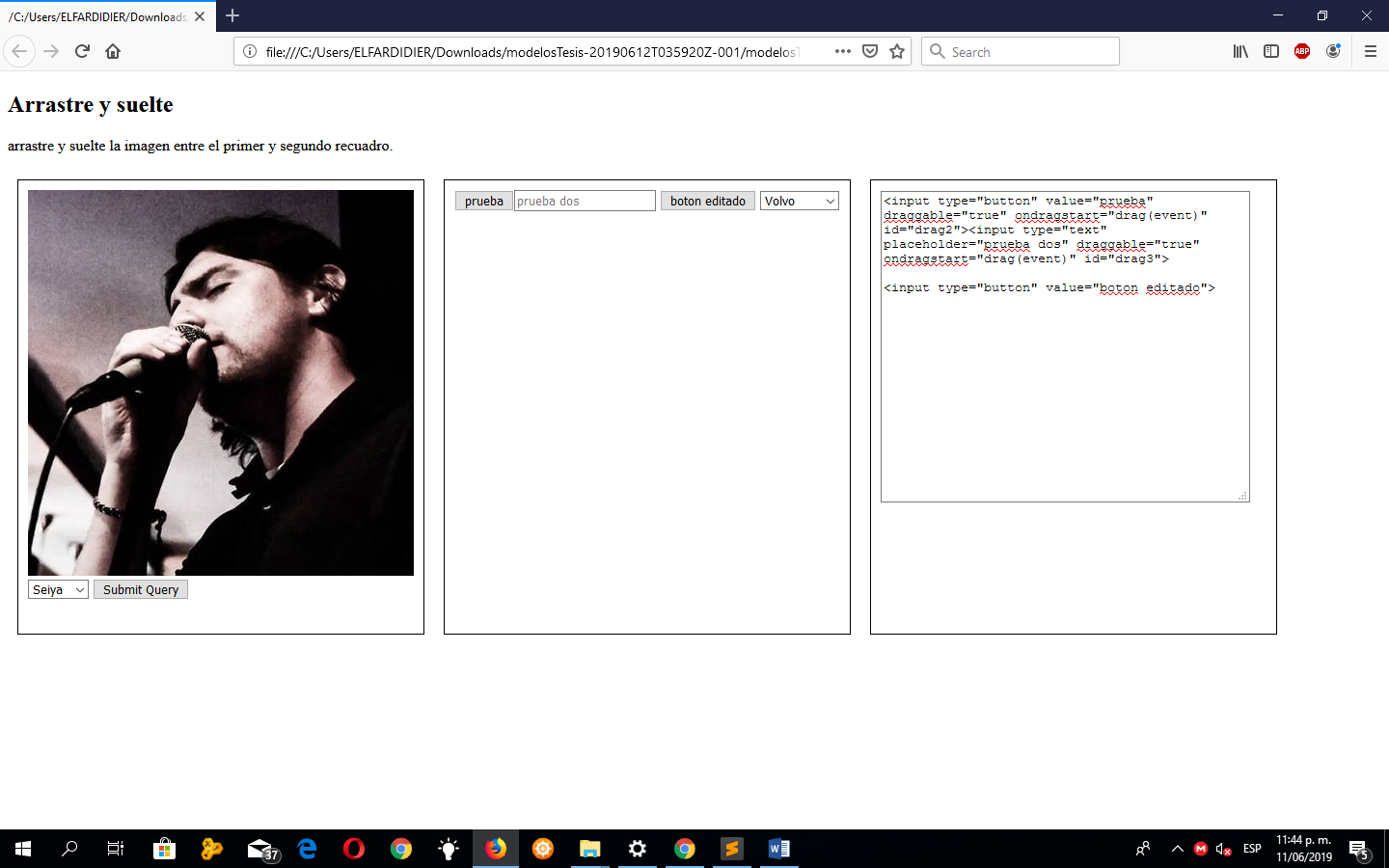


Figura12. Componente no actualizado en el área de código.

Como se puede observar en la figura 12, a pesar de que se arrastró un nuevo componente al área de trabajo (sección 2), sus instrucciones no son presentadas en el área de código. Esto quizás se deba a un conflicto de instrucciones, o que sencillamente no es posible realizar la interdependencia de los objetos pues existirá una jerarquía de acciones y componentes que debe ser respetada. De cualquier forma, es una oportunidad para explorar e investigar más el asunto, de forma que se pueda desarrollar el proyecto tal cual se concibió en su idea original.

# Resultados Alcanzados

Como se pudo observar en la descripción de la experiencia, se obtuvo una versión preliminar del simulador, que ilustra la funcionalidad clave que se espera obtener: Que se puedan arrastrar componentes HTML de una primera sección hacia una segunda sección, y que el código de dicho componente se pueda visualizar en una tercera sección. Es cuestión de ahondar en la investigación, para hacer más funcional el simulador e incluir distintos componentes y arquitecturas que le permitan al estudiante construir formularios completos desde el simulador. De acuerdo al plan de trabajo, se han alcanzado los objetivos pues se ha realizado una Investigación de modelos , interfaces, y lenguajes de programación aptos para desarrollar el simulador para el aprendizaje del lenguaje de marcas HTML y también se ha realizado una Investigación de la forma en que serán presentados los conceptos en el simulador (figuras, formas, colores, animación, ejemplos, situaciones, organización de conceptos). Se espera seguir trabajando de acuerdo al cronograma, de manera que se puedan lograr los objetivos, y poner en marcha este simulador en entorno real de enseñanza-aprendizaje.

Sin embargo, como se pudo observar en la sección denominada Integración de HTML editable en tiempo real y el Prototipo de Sistema Drag and Drop para el Simulador, las pruebas dieron como resultado que cuando se escriben instrucciones en el área de código (sección 3), efectivamente es posible visualizar la materialización de dicho código en el área de trabajo (sección 2), sin embargo, cuando se intenta arrastrar un nuevo elemento del área de componentes (sección 1) al área de trabajo (sección 2), la información de las nuevas instrucciones no es actualizada en el área de código (sección 3). Es posible que exista un conflicto de instrucciones que impida esta actualización, o que la misma arquitectura del lenguaje de programación no permita realizar dicha interdependencia. De cualquier forma, es una oportunidad más para seguir investigando, y tratar de encontrar una solución que permita concluir el simulador de acuerdo a los diseños iniciales, sin modificar tanto ni la estructura, ni el lenguaje de programación elegido.

# Conclusiones y Recomendaciones

Se puede concluir parcialmente que es posible construir el simulador propuesto en este trabajo de grado, utilizando el lenguaje de programación JAVASCRIPT y su librería JQUERY, de manera que puede materializarse la solución planteada en la introducción de este trabajo ¿Será factible desarrollar un instrumento que facilite el proceso de edición y visualización de código html, de manera que el estudiante experimente simultáneamente la escritura y la percepción de la página web? Sin embargo, es necesario sortear algunos obstáculos, pues es posible que no se pueda seguir al pie de la letra el diseño inicial del simulador debido a limitaciones del lenguaje de programación elegido. En ese punto, es necesario replantear el diseño, para no cambiar la arquitectura desarrollada, y continuar con un prototipo que posea todas las funcionalidades, pero cambiando su aspecto visual. Por ejemplo, es posible diseñar una sección superior del simulador que posea las tres secciones trabajadas en el diseño inicial, y una sección inferior que permita realizar edición de código en tiempo real, como se ilustra en la figura 13.

|  |  |  |
| --- | --- | --- |
| COMPONENTES HTML  Estos componentes permitirán ser arrastrados, de manera que sean soltados en el área de trabajo | AREA DE TRABAJO  En el área de trabajo se sueltan los componentes y se organizan visualmente de acuerdo a los requerimientos del usuario | AREA DE CÓDIGO  En el área de código se visualizará el código HTML generado, de acuerdo a la organización y componentes proporcionada en el área de trabajo. |
| AREA DE TRABAJO EDITABLE | AREA DE CÓDIGO EDITABLE |

Tabla 2. Nueva Interfaz gráfica propuesta

# Referencias bibliográficas

Anama, M., Herrera, L., Solarte, A., Torres,M. (2020) Prácticas Educativas Socio constructivistas en Hispanoamérica <https://repository.ucc.edu.co/items/d3ad1099-e9a3-4660-9757-32f8808a15a6> . [Consultado el 04/03/2023 11:10].

Arcila, P., Mendoza, Y., Jaramillo, M., & Cañón, Ó. (2010). Comprensión del significado desde Vygotsky, Bruner y Gergen. Diversitas: Perspectivas en psicología, 6(1), 37-49. <https://www.redalyc.org/pdf/679/67916261004.pdf>

Barrows, H.S. y Tamblyn, R.M. (1980). Problem-based learning: an approach to medical education. New York: Springer Publishing Company.

Berners-Lee, Tim (2000): Weaving the Web, Harper Business.

Boud, D., Cohen, R. y Walker, D. (1993) Understanding learning from experience, en Boud, D., Cohen, R. y Walker, D. (Eds.) Using Experience for Learning, SRHE y Open University Press, Buckingham, págs. 1-17.

Cailliau, Robert (1995): A Short History of the Web. Text of a speech delivered at the launching of the European branch of the W3 Consortium. http://www.netvalley.

com/archives/mirrors/robert\_cailliau\_speech.htm [Consultado el 30/04/2019].

Davison, Daniel B; Chen, Edward (1995): A brief introduction to the internet,

Computers & Geoscience, vol. 21, n.º 6, pp. 731-735

Domingo, J. (2008). El aprendizaje cooperativo. Cuadernos de trabajo social, 21, 231-246. <https://revistas.ucm.es/index.php/CUTS/article/download/CUTS0808110231A/7531> [Consultado el 04/03/2023 18:28].

Espinar, E., Vigueras, J (2020) El aprendizaje experiencial y su impacto en la educación actual. Revista Cubana de Educación Superior. <http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0257-43142020000300012> [Consultado el 04/03/2023 11:25].

García, R., Traver, J. A., & Candela, I. (2001). Aprendizaje cooperativo. Fundamentos, características y técnicas. Madrid: CCS.

Nelson, Ted (1970): No More Teacher´s Dirty Looks, Computer Decisions

OVEJERO, A. (1990). El aprendizaje cooperativo. Una alternativa efi caz a la enseñanza tradicional. Barcelona: PPU.

Poot-Delgado, C. A. (2013). Retos del aprendizaje basado en problemas. Enseñanza e investigación en psicología, 18(2), 307-314. <https://www.redalyc.org/pdf/292/29228336007.pdf> [Consultado el 04/03/2023 14:36].

Raggett, Dave; Lam, Jenny; Alexander, Ian y Kmiec, Michael (1998): Raggett on HTML 4 Capítulo 2. Addison Wesley Longman. <http://www.w3.org/People/Raggett/book4/ch02.html> [Consultado el 04/04/2019 11:10].

Rodríguez, Zulema & Gutiérrez, Martha & BURITICÁ, OLGA. (2011). EL SOCIOCONSTRUCTIVISMO EN LA ENSEÑANZA Y EL APRENDIZAJE ESCOLAR.

Tabarés G, Raul, (2014) El inicio de la Web: historia y cronología del hipertexto hasta HTML 4.0 (1990-99). <http://revistas.usal.es/index.php/artefactos/article/viewFile/12423/12757> [Consultado el 04/04/2019 10:05]

Villalobos, J. (2009). Proyecto cupi2–una solución integral al problema de enseñar y aprender a programar. Bogotá: Universidad de los Andes.

Villalobos, J.A., Casallas, R., Marcos, K. El Reto de Diseñar un Primer Curso de Programación de Computadores. XIII Congreso Iberoamericano de Educación Superior en Computación, Cali, Colombia, Octubre 2005.