

C-ToolP: Herramienta de apoyo para la enseñanza de desarrollo Web-SPA-MVC C-ToolP: a support tool for learning of web development-SPA-MVC

L.H. García-Islas ^{a,*}, K.D. Franco-Sánchez ^a, T.I. Samperio-Monroy ^a, I.H. Pérez-Tavera ^a, A. Ocampo-López ^a

^a Área Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, 42184, Pachuca, Hidalgo, México.

Resumen

El uso de internet hoy en día se ha vuelto parte esencial en la vida de la sociedad. El uso de aplicaciones web aplica en cualquier ámbito científico, tecnológico o social. Es por ello que desarrollar aplicaciones web bajo estándares de calidad es importante para cualquier empresa o institución que requiera interactuar con la gente a través de ellas. Un estándar muy utilizado es el de aplicaciones de tipo SPA (*Single Page Application*), el cual posee una arquitectura de tipo MVC (Modelo-Vista-Controlador). El presente trabajo muestra una herramienta que construye y ejemplifica la forma de estructurar una aplicación de tipo SPA bajo la arquitectura MVC para reafirmar los conceptos teóricos de esta rama del desarrollo web.

Palabras Clave:

Desarrollo web, Modelo Vista Controlador, Aplicaciones de Página Única, Microaprendizaje, Herramientas CASE.

Abstract

Internet use nowadays has become an essential part in society's life. The use of web applications applies for any approach, such as scientific, technological or social. In consequence, developing standard based web applications is important to any enterprise or institution that requires interaction with people through them. A commonly used standard is SPA (*Single Page Application*) which is composed by an MVC (Model-View-Controller) architecture. The present work shows a tool that builds and samples the way to assemble a SPA application under MVC architecture in order to confirm theoretical concepts for this area of web development.

Keywords:

Web Development, Model View Controller, Single Page Application, Microlearning, CASE Tools.

1. Introducción

El software se ha incrustado profundamente en casi todos los aspectos de nuestras vidas y, como consecuencia, el número de personas que tiene interés en las características y funciones que brinda una aplicación específica, ha crecido en forma notable. Cuando ha de construirse una aplicación nueva o sistema incrustado, deben escucharse muchas opiniones y en ocasiones parece que cada una de ellas tiene una idea un poco distinta de cuáles características y funciones debería tener el software.

Los requerimientos de la tecnología de la información que demandan los individuos, negocios y gobiernos se hacen más complejos con cada año que pasa. En la actualidad, los grandes equipos de personas crean programas de cómputo que antes eran elaborados por un solo individuo. El software sofisticado, que alguna vez se implementó en un ambiente de cómputo predecible

y autocontenido, hoy en día se halla incrustado en el interior de todo, desde la electrónica de consumo hasta dispositivos médicos o sistemas de armamento.

La complejidad de estos nuevos sistemas y productos basados en computadora demanda atención cuidadosa a las interacciones de todos los elementos del sistema.

1.1.1. Desarrollo web

En la ingeniería de software, se denomina aplicación web a los aplicativos en los cuales los usuarios pueden hacer uso, por medio del acceso a un servidor web, a internet o una intranet mediante un navegador, de modo tal que el programa resultante es codificado en un lenguaje que puede ser interpretado por los navegadores web (Mikowski y Powell, 2013).

*Autor para la correspondencia: luishg@uaeh.edu.mx

Correo electrónico: luishg@uaeh.edu.mx (Luis Heriberto García-Islas), kristell_franco@uaeh.edu.mx (Kristell Daniella Franco-Sánchez), smtheria@uaeh.edu.mx (Theira Irasema Samperio-Monroy), itavera@uaeh.edu.mx (Iván Horacio Pérez-Tavera), aocampo@uaeh.edu.mx (Arturo Ocampo-López).

1.1.2. Single Page Application (SPA)

Una de las arquitecturas más utilizadas hoy en día es la arquitectura web de aplicaciones de página única (Single Page Application). Las aplicaciones de tipo SPA son aplicaciones web que cargan una única página HTML (*Hyper Text Markup Language*) y actualiza su contenido de manera dinámica conforme el usuario va interactuando con la App (Mikowski y Powell, 2013). SPA hace uso de tecnologías como AJAX (Asynchronous JavaScript and XML) y HTML versión 5 (Duckett, 2014). Para crear aplicaciones web fluidas y responsivas, evitando las constantes recargas de páginas de las aplicaciones web tradicionales. Sin embargo, esto significa que mucho del trabajo ocurre del lado del cliente, por medio de *JavaScript*. La Figura 1 muestra la forma en que se procesa la información para este tipo de aplicaciones web mediante el uso de AJAX (D'mello y Sriparasa, 2018).

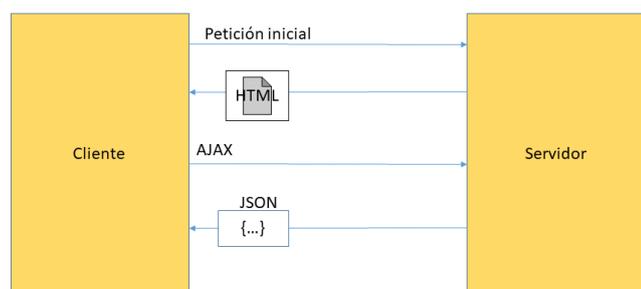


Figura 1: Procesamiento por medio de AJAX.

1.1.3. Arquitectura Modelo-Vista-Controlador (MVC)

Para una mejor organización del código fuente, un modo de hacerlo es a través de la organización con una arquitectura de tipo MVC (Veit y Herrmann, 2003), la cual consta de tres capas. Las Figuras 2 y 3 muestran la interacción entre las capas de la arquitectura MVC. A continuación, se describen cada una de las capas que componen a la arquitectura:

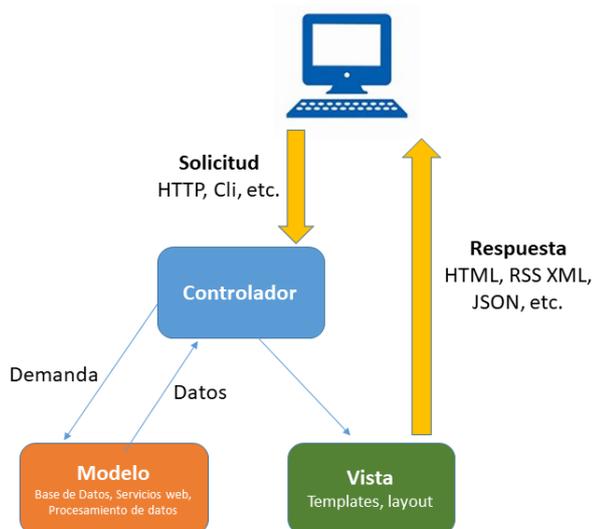


Figura 2: Arquitectura MVC.

- **Vista.** Igual llamada interfaz de usuario, que se compone de la información que es enviada al cliente y los mecanismos que permiten la interacción con éste. Esta capa está representada a través de archivos HTML, los cuales internamente harán uso y referencia a archivos que apoyen a la interacción con la capa de controlador.

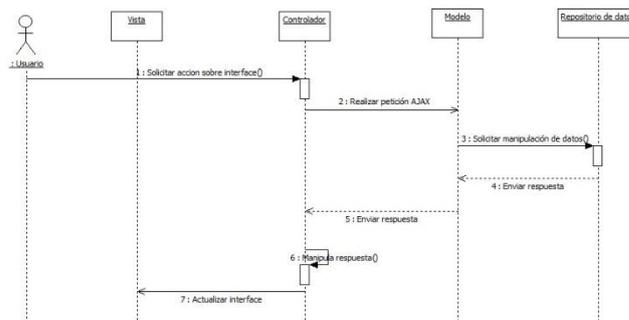


Figura 3: Secuencia de procesamiento por medio de AJAX.

Los archivos que carga internamente son los siguientes:

- Controladores (Archivos de lenguaje Javascript).
- Hojas de estilo (Archivos de lenguaje CSS - Cascading Style Sheet).
- **Controlador.** Actúa como intermediario entre las capas de Modelo y Vista, administrando el flujo de información entre ellas y las transformaciones para adaptar los datos a las necesidades de cada una. La capa de controlador tiene dos funciones principales:
 - Realizar peticiones AJAX.
 - Actualizar la capa de Vista de manera dinámica.
- **Modelo.** Contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

Entender todos estos conceptos requiere de un conjunto de elementos pedagógicos que permita encapsularlos para adquirir una habilidad muy particular. En la siguiente sección, se describe el enfoque utilizado en el presente trabajo, el cual es el microaprendizaje.

1.2. Microaprendizaje

La neuroeducación es la disciplina que estudia cómo aprende el cerebro y afirma que necesita emocionarse para aprender y las emociones sirven para almacenar y recordar de una forma efectiva, por tanto, es importante conocer qué estímulos despiertan la atención para mantenerse motivado, curioso y éstos son la base sobre la que se sustentan los procesos de aprendizaje y memoria (De la Garza, 2020).

El microlearning o microaprendizaje, está asociado inicialmente a las investigaciones y congresos celebrados por la Universidad de Innsbruck, donde se discuten desde el primer congreso Microlearning 2005 (Hug et al., 2006), los elementos que conforman el enfoque y la conexión con el aprendizaje en la era digital, ubicuo, asociado a dispositivos móviles (mlearning) e inicialmente al ámbito informal, además de, fundamentalmente, estar relacionado con el marco de las comunidades de práctica, del aprendizaje a lo largo de la vida (Lifelong Learning – LLL). La idea central resulta de compartir un contenido “micro” que puede ser atendido de forma aislada o de forma secuencial dependiendo de la plataforma que lo comparte y ofrece alternativas a los métodos tradicionales de desarrollo profesional como sistemas tutoriales o cursos en línea, es por ello que se considera una línea independiente de investigación por la creciente necesidad de formación permanente en modalidad virtual o el aprendizaje de la demanda de los miembros de nuestra sociedad, como los llamados trabajadores del conocimiento (Conole y Oliver, 2006).

El microaprendizaje responde a la necesidad de un desarrollo personal o profesional de forma informal y personalizada, donde las personas atienden los contenidos específicos de lo que necesitan aprender y no todos los elementos que conforman la secuencia de temas del aprendizaje formal, es por ello que están relacionados con la visión de conceptos como “Just-in-time learning” o “Learning on demand” (Linder, 2006).

La relación del microaprendizaje (aprendizaje informal) con la escuela (aprendizaje formal) puede ser muy estrecha, primeramente porque el aprendizaje se ve mediado por las TIC y el acceso a la información en la red (información multicodificada o multiformato), pero más aún porque es percibido como un proceso de tomar pequeñas piezas de información o de fragmentar el contenido en módulos que pueden ser aprendidos separadamente en cualquier momento cuando el acceso al entorno de aprendizaje está garantizado (Hug et al., 2006).

Las principales características del Microaprendizaje, según De la Garza (2020) son:

- *Brevedad*: Las lecciones tienden a ser cortas y no superar los 15 minutos.
- *Variiedad*: El contenido de estas lecciones pueden presentarse de distintas formas (juegos, vídeos, diapositivas, preguntas, etc.).
- *Granularidad*: Se centra en un solo tema o concepto para focalizar el contenido. Tiene un único objetivo formativo.
- *Herramienta tecnológica*: Los temas son ofrecidos a través de diferentes formatos según su visualización (tamaño pantalla, ancho de banda, navegación, etc.). Permite al alumno acceder con más facilidad en momentos específicos del día.
- *Aprendizaje informal*: Podría describirse como un aprendizaje informal ya que se adquieren conocimientos en estructura de microcontenido o micromedios y no están relacionados forzosamente con otra secuencia de temas.

1.2.1. Estrategia para la creación de cursos más cortos

En el minicurso titulado “Menos, es más: Microlearning” expuesto por la Mtra. Estela De la Garza (2020), propone los siguientes pasos a considerar en el diseño instruccional para crear cursos más cortos:

- 1) *Organizar el contenido*: Identificar los contenidos que se desean abordar con un objetivo específico y una estrategia, utilizando un guion gráfico para definir la forma de mostrar los contenidos evitando extenderse mucho en ello.
- 2) *Eliminar datos innecesarios*: Sintetizar la información escrita y apoyarse en medios visuales que muestren un equilibrio entre claridad y brevedad.
- 3) *Definir lo que necesitan aprender*: Dar prioridad a enseñar los contenidos que tienen más dificultad de aprender, debe ser lo que los participantes necesitan aprender y que es importante en el curso.
- 4) *Ser muy visual*: Mostrar la información a los participantes a partir de sus estilos de aprendizaje con materiales visualmente agradables en formato de imagen, audio, video, páginas web atractivas y simples de utilizar o aplicaciones interactivas que le permitan practicar.

5) *Eliminar lo repetitivo*: Evitar que los contenidos se aborden de forma recurrente o que siempre se muestren los contenidos en el mismo formato.

6) *Partir el contenido*: El diseño de los materiales o recurso educativos para la enseñanza deben de partir de aspectos importantes como el contenido, el tiempo, la profundidad del tema, la interacción y relación con el contenido, entre otros aspectos.

En resumen, podemos decir que el microlearning es la organización planificada de breves experiencias de aprendizaje diseñadas para cumplir un objetivo concreto de aprendizaje, y que pueden o no insertarse en un itinerario de formación más extenso. Las experiencias de microlearning siempre deben estar orientadas a la acción dentro de dicho objetivo y ayudar a los participantes a practicar lo que están aprendiendo (Abril, 2020).

Existen diferentes formatos atractivos e innovadores que se pueden implementar en el microlearning y son formatos de múltiples dispositivos (compatibles con PCs de escritorio / portátiles, tablets y smartphones) y depende la elección del tipo de contenido a mostrar, entre ellos están las aplicaciones móviles, vídeos cortos, vídeos interactivos, vídeos animados de pizarra, vídeos de animaciones cinéticas basadas en texto, podcast, simulaciones 3D basadas en escenarios de decisión, infografías, eBooks y flipbooks, iPDF (archivos PDF interactivos), interacciones en redes sociales: debates en Facebook y/o LinkedIn, hilos de twitter, historias en Instagram, entre otros.

El presente trabajo presenta la herramienta C-ToolIP como una herramienta CASE, cuyos conceptos se describen a continuación.

1.3. Herramientas case

Son un complemento de la caja de herramientas del ingeniero del software que le proporciona, la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y diseño asistidos por computadora que utilizan de otras disciplinas, las herramientas CASE ayudan a asegurar que la calidad sea algo diseñado antes de llegar a construir el producto. Acorde a Turban et al. (2001) definen al CASE de la siguiente manera: CASE (*Computer Aided Systems Engineering*) cuyo significado en español es Ingeniería de Software Asistida por Computadora, es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias del desarrollo de sistemas y al igual que las herramientas CAD (*Diseño Asistido por Computadora*) o CAM (*Manufactura Asistida por Computadora*) su objetivo es, acelerar el proceso para el que han sido diseñadas, en el caso de CASE para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

El objetivo del presente trabajo es desarrollar una herramienta CASE que permita reforzar la enseñanza del desarrollo web bajo el esquema SPA y organizando los archivos que contienen dicho desarrollo bajo la arquitectura MVC. En la siguiente sección se describe la forma en que fue estructurada la herramienta C-ToolIP.

2. Metodología de implementación del microaprendizaje

En esta sección se muestra la metodología de implementación del microaprendizaje, el papel que desempeña la herramienta C-ToolIP y la forma en que ésta confirma los conceptos y el paradigma aplicado en el desarrollo de aplicaciones Web bajo el esquema SPA con arquitectura MVC. La Figura 4 permite ver el esquema de microaprendizaje propuesto. Como se puede observar, en primer lugar se propone la revisión teórica de las partes que componen un desarrollo web bajo el esquema SPA y

bajo la arquitectura MVC. Posteriormente viene la fase práctica donde se explican las guías y sintaxis. Por último, el uso de la herramienta C-ToolP permite crear en tiempo real el código necesario para ejemplificar tanto los conceptos teóricos como las prácticas desarrolladas para que los estudiantes puedan cotejar lo que ellos desarrollaron con lo que arrojó la herramienta C-ToolP para poder reafirmar conocimientos.



Figura 4: Metodología del microaprendizaje.

En cuanto a la implementación de la herramienta C-ToolP, la arquitectura MVC permite ayudar a separar los datos entre la aplicación, el usuario y la lógica de control, dicha arquitectura está compuesta por tres apartados (Veit, 2003, págs. 140-149). La Figura 5 muestra el desglose de la organización de archivos y la arquitectura del Modelo MVC.

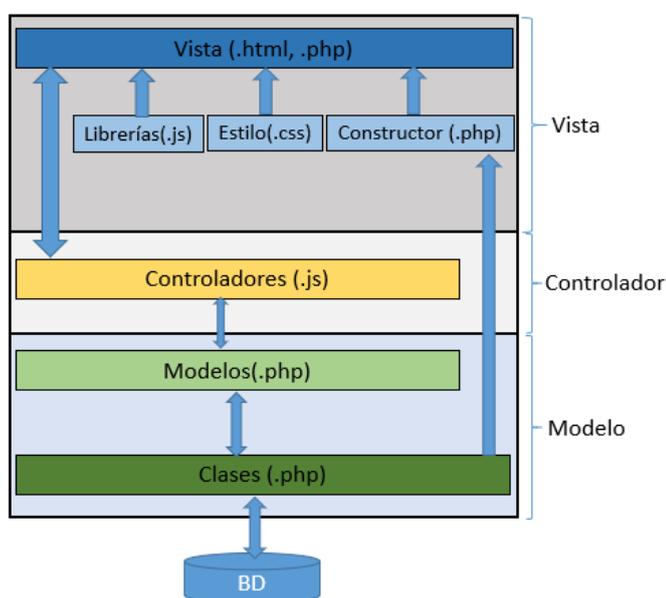


Figura 5: Arquitectura basada en MVC.

Las capas que se encuentran en esta arquitectura son:

- **Vista:** Interfaz HTML para el usuario final. Incluye llamadas a estilos CSS, librerías JavaScript, manejadores de eventos, controladores y el constructor.
- **Estilo:** Configura colores, fuentes y demás estilos visuales.
- **Widgets:** Controles con funcionalidad especial.
- **Librerías generales:** Framework (jQuery, prototype), librerías propias de uso general para vistas, AJAX, etc.
- **Controlador:** Es el encargado de activar los widgets y los eventos de la vista.
- **Modelo:** Es el que recibe la llamada AJAX del controlador, instancia las clases y obtiene información de éstas a través de

métodos propios de las mismas. Envía un objeto JSON al controlador de widgets o al controlador.

- **Clases:** Se encargan de las llamadas de bases de datos hacia vistas y procedimientos almacenados como parte del manejador de base de datos.

Con base en la descripción anterior, la herramienta es capaz de proporcionar una interfaz gráfica que permite configurar los módulos y los componentes de cada uno de ellos y con base en dichos parámetros generar el código necesario para definir las capas de vista, controlador, así como un código que simula la capa de modelo para que, sin necesidad de objetos de base de datos, se puedan crear modelos funcionales que completen la funcionalidad de un contexto bajo la arquitectura MVC. La Figura 6 muestra la arquitectura que ha sido diseñada para la herramienta C-ToolP.

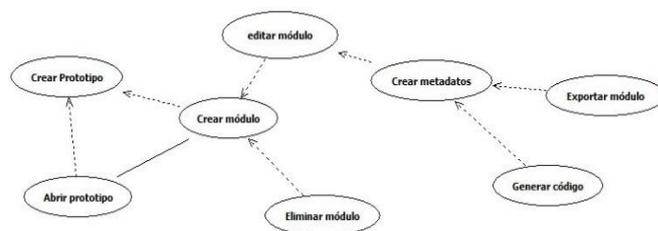


Figura 6: Arquitectura basada en MVC.

Una vez mostrado el esquema general de la metodología de implementación del microaprendizaje, a continuación, se presenta la interfaz desarrollada para la herramienta propuesta.

3. Diseño de la interface de enseñanza

Al ejecutar el Sistema, la primera pantalla permite seleccionar la opción de crear un nuevo prototipo o abrir uno ya creado, tal como lo muestra la Figura 7.

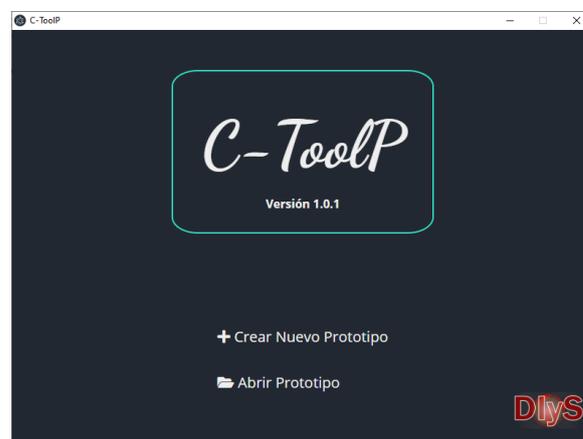


Figura 7: Pantalla de inicio de la herramienta.

Al momento de seleccionar un prototipo nuevo o al abrir uno creado, la siguiente pantalla permitirá seleccionar la ruta donde guardar o donde abrir (según sea el caso) el prototipo a utilizar, como se muestra en la Figura 8.

Una vez seleccionado el archivo del prototipo, la siguiente pantalla, mostrada en la Figura 9, permite exponer el listado de módulos que componen al prototipo.

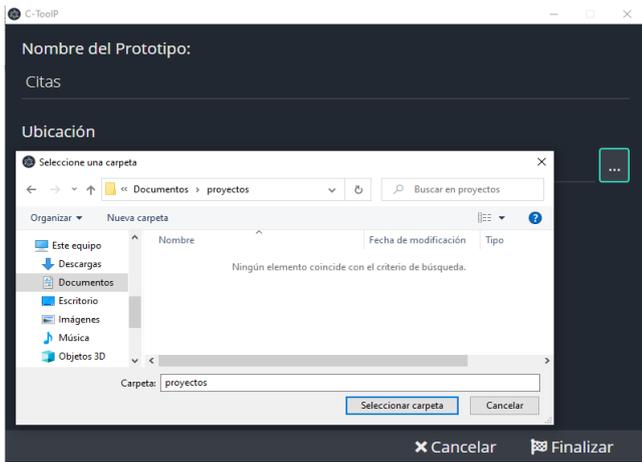


Figura 8: Selección de ruta y archivo que contiene prototipo.



Figura 9: Listado de módulos del prototipo.

Al seleccionar un módulo en particular, el siguiente paso será especificar la estructura de datos que compone al módulo. Como muestra la Figura 10, en cada campo se debe especificar nombre, tipo de dato y opcionalmente también se puede agregar una descripción a cada campo creado.

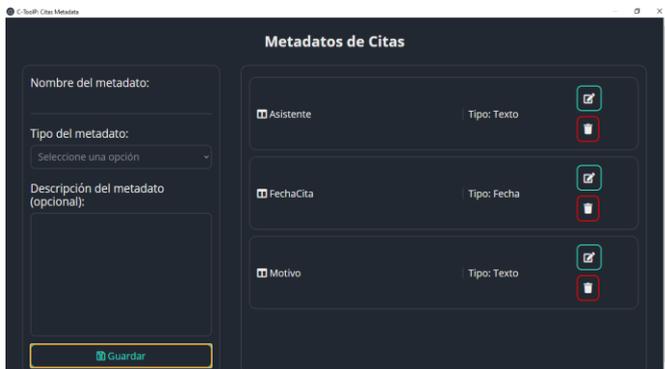


Figura 10: Definición de la estructura de datos del módulo.

4. Resultados

A partir de los módulos creados, el programa genera en una carpeta de nombre <<out>>, tal como se muestra en la Figura 11, la siguiente estructura de archivos:

- Archivos estáticos para el funcionamiento del prototipo.
- Módulo de administración, donde se generan:
 - Carpeta de controladores.
 - Carpeta de modelos.
 - Carpeta de vistas.

- Carpeta de widgets.
- index.html.
- Módulos para cada módulo creado en la aplicación, en los cuales se generan:
 - Carpeta de controladores.
 - Carpeta de modelos.
 - Carpeta de vistas.
 - Carpeta de widgets.
 - index.html.
- index.html, el cual contiene el acceso a todos los módulos incluyendo el de administración.

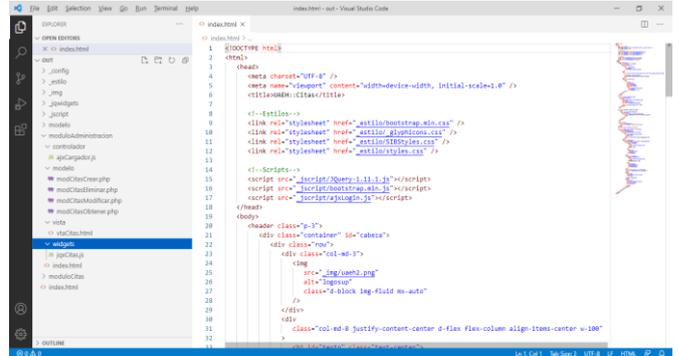


Figura 11: Archivos creados por la herramienta.

Los archivos de código resultante permiten ejemplificar los conceptos y el funcionamiento de un desarrollo web bajo el esquema SPA y con una organización bien definida, como lo es la arquitectura MVC, el cual es perfectamente funcional como se puede apreciar en las Figuras 12 y 13.



Figura 12: Prototipo funcionando.

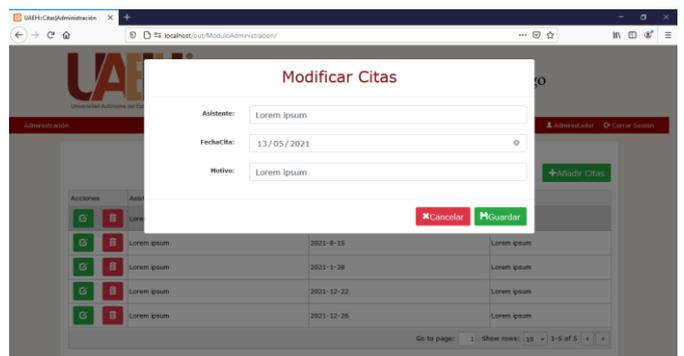


Figura 13: Prototipo funcionando (edición).

5. Conclusiones

El desarrollo web requiere no solo de estándares bien definidos, sino de personas con habilidades y características cada vez más especializados que permitan el desarrollo de aplicaciones web con tiempos de desarrollo cada vez menores y con niveles de calidad que permitan al usuario de dichas aplicaciones una experiencia de uso y ergonomía en sus actividades cotidianas. El uso de Herramientas CASE es una gran ayuda para poder desarrollar códigos de forma muy uniforme y con mayor rapidez. La herramienta CASE mostrada en el presente trabajo permite, además de generar el código que se requiere para una aplicación web de tipo SPA estructurado bajo la arquitectura MVC, ejemplificar y mostrar todas y cada una de las partes que se requieren para dicho aplicativo, abriéndole un mundo de posibilidades a los estudiantes ya que puede identificar las partes del código generado por la herramienta y la forma de estructurar un aplicativo web bajo las características antes mencionadas. El uso de la presente herramienta combinado con una serie de prácticas y conceptos teóricos, permitirán una curva de aprendizaje muy baja y con resultados adecuados en la enseñanza del desarrollo web.

Referencias

- Abril, H. (2020). Microlearning: conceptos básicos para diseñar micro-experiencias de aprendizaje. Soyprofesoronline.com recuperado desde: https://soyprofesoronline.com/microlearning-conceptos-basicos-para-diseñar-micro-experiencias-de-aprendizaje/#Que_es_y_que_no_es_microlearning
- Conole, G., & Oliver, M. (2006). Contemporary perspectives in E-learning Research themes, methods and impact on practice. New York, Routledge.
- De la Garza, E. (2020). Menos en más: Microlearning. Minicurso de idea “Innovación y Diseño de Experiencias de Aprendizaje”. Recuperado desde: https://idea.itesm.mx/wp-content/uploads/2017/12/Minicurso-Microaprendizaje_EsteladelaGarza.pdf
- D’mello, B. J., y Sriparasa, S. S. (2018). Javascript and json essentials: Build light weight, scalable, and faster web applications with the power of json, 2nd edition (1st ed.). Packt Publishing.
- Duckett, J. (2014). Javascript and jquery: Interactive front-end web development (1st ed.). Wiley Publishing.
- Hug, T., Lindner, M., & Bruck, P. (2006). Microlearning: Emerging concepts, practices and technologies after e-Learning. Proceedings of Microlearning 2005. Learning & Working in new Media. Innsbruck: Innsbruck University Press.
- Lindner, M. (2006). Use these tools, your mind will follow. Learning in immersive macromedia and microknowledge environments. En whitelock, d.
- Mikowski, M., y Powell, J. (2013). Single page web applications: Javascript end-to-end (1a ed.). Greenwich, CT, USA: Manning Publications Co.
- Turban, E., McLean, E. R., Wetherbe, J. C., Westfall, R., Rainer, K., y Nagore, C. G. (2001). Tecnologías de información para la administración. México: Compañía Editorial Continental.
- Veit, M., y Herrmann, S. (2003). Model-view-controller and object teams: A perfect match of paradigms. En Proceedings of the 2nd international conference on aspect-oriented software development (pp. 140-149). New York, NY, USA: ACM. Descargado de <http://doi.acm.org/10.1145/643603.643618>