



UNIVERSIDAD TÉCNICA DE BABAHOYO

FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA.

PROCESO DE TITULACIÓN

NOVIEMBRE 2020 – MAYO 2021

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PREVIO A LA OBTENCION DEL TITULO DE INGENIERO(A) EN SISTEMAS

TEMA:

ESTUDIO DE FACTIBILIDAD DE USO DEL FRAMEWORK VUE.JS COMO
HERRAMIENTA, PARA EL DESARROLLO DEL FRONT END DE UN APLICATIVO
WEB, PARA LA GESTIÓN ACADÉMICA EN LA UNIDAD EDUCATIVA MIGUEL
ÁNGEL SAMANIEGO JIMÉNEZ EN LA PARROQUIA LA UNIÓN

EGRESADO(A):

ROBLES HERRERA DIDIER EMMANUEL

TUTOR:

ING. MEJÍA VITERI JOSÉ TEODORO, MSC

AÑO 2021

INTRODUCCION

Desde la aparición de la famosa arquitectura cliente-servidor en la década de los 60, siendo IBM/OS 360 el precursor en esta arquitectura, la misma conforme han pasado los años, se ha desarrollado de manera exponencial, hoy en día los términos front end y back end hacen alusión a esta técnica; siendo el front end el estandarte del cliente mientras que el back end es el estandarte del servidor.

La construcción de un front end es una tarea delicada dentro de la creación de un proyecto, debido a que no solo puede regirse a los gustos del desarrollador o del cliente, sino también debe regirse a las capacidades técnicas que nos brindan determinadas herramientas de desarrollo. Una página web tradicional se diferencia a una aplicación web porque, la primera hace referencia a una página con información estática no susceptible a la manipulación, mientras, que la aplicación es aquella que se ejecuta en la web así manipulando, procesando presentando y transmitiendo los resultados de manera online, siendo el usuario el que se relaciona directamente con el resultado de ese proceso, denominado producto de información.

El presente caso de estudio se enfoca de manera primordial en determinar qué tan plausible es el uso de un framework específico como herramienta para la construcción del front end de una aplicación web que se encargara de la gestión académica de la unidad educativa Miguel Ángel Samaniego Jiménez.

Determinar la factibilidad de un framework a más de ser una problemática para el desarrollo de la aplicación web también representa un interés académico dentro de la formación de un profesional, reunir información teórica de un determinado framework, fomenta una implementación rápida, una determinación de ventajas y desventajas más precisa y ayuda a desarrollar ese sentido común que todo desarrollador debe poseer al momento de la elección de herramientas para enfrentarse a nuevos proyectos en su vida profesional

Para llevar a cabo el caso de estudio se usara las técnicas y herramientas que brindan la investigación cualitativa, este tipo de investigación se caracteriza por brindar a través del uso de técnicas o herramientas como la entrevistas o encuestas, datos no numéricos, de los cuales se puede partir para recopilar información útil y transfórmala en conocimientos teóricos, una vez obtenido estos conocimientos se puede desglosar las ventajas y desventajas de una posible implementación de dicho framework.

Con la presente investigación se busca analizar de manera detallada algunas de las funciones que transforman a este framework en un opción viable para usarla como herramienta de desarrollo, de la misma manera se buscar contrastar la información obtenida con las preferencias del cliente y las limitaciones del propio framework. Buscando así, también, considerar si es factible implementar un front end desarrollado con esta herramienta.

La investigación aborda todos los temas a tratar en su marco de desarrollo, detallando desde la historia de su creación hasta sus ventajas y desventajas en el amplio mundo del desarrollo del front end. En el marco de desarrollo también se detalla el estudio de factibilidad dirigido hacia las especificaciones técnicas que conlleva implementar un front end desarrollado por completo en VueJS.

DESARROLLO

Aplicación web

Un aplicación web es una aplicación que se ejecuta completamente online, siendo así que sus datos deben de ser subidos por el usuario y el producto de información generado por esta aplicación, debe ser descargado por el mismo usuario, según la definición de (LOPEZ GOMEZ & CÁCERES ESCUDERO, 2019) “Consiste en un software basado en Internet, en el cual una población extensa de usuarios, por medio un navegador, hacen peticiones remotas y esperan una respuesta que puede implicar una mezcla de publicación impresa y desarrollo de software”.

Las diferencias entre una página web (tradicional), y una aplicación web, son bastante claras. En función de la página web los datos se muestran orientados hacia un propósito meramente informativo, y el procesamiento de datos es bajo, o quizás nulo. Una aplicación web es más compleja puesto que tiene un alto procesamiento de datos y su función va más allá que solo brindar información.

Historia del front end

De acuerdo a (González, 2019) investigador e historiador afirma que la arquitectura cliente servidor dio sus primeros pasos en la década de los 60's con el IBM OS-360, este sistema operativo formaba parte del computador IBM S-360 que dio inicio a la tercera generación de computadores. Dentro de las funciones de este revolucionario computador encontrábamos un sistema de peticiones y respuestas en tiempo real, el cual fue el pilar base para los inicios de la arquitectura cliente-servidor

De acuerdo a (López Sanz)“Cuando hablamos de la arquitectura en capas de la web, nos referimos a la forma en la cual se distribuye la infraestructura, es decir los elementos de hardware en los que se ejecutaran los procesos y que constituyen el sistema”, este enunciado

nos muestra de un forma clara hacia donde está dirigido la arquitectura cliente-servidor, dividiendo en dos capas la manera en la que una aplicación web funciona; esta división da como resultado la capa que pertenece al cliente y la capa que pertenece al servidor, estableciendo que el equipo en el que se ejecuta la aplicación es el servidor mientras que el equipo en el que se presentan los datos visuales es el cliente.

Por último pero no menos importante está el enfoque de (VALDIVIA MIRANDA, 2017) que detalla que “La arquitectura cliente servidor es un modelo de gestión en el que un equipo (Servidor) ofrece un servicio a otros equipos (Clientes) que lo necesitan.” Este enfoque está dirigido a exclusivamente que es lo que demanda el cliente y que es lo que oferta el servidor.

Front end

Según una definición propuesta por (Stefaniak, 2019)“Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación”. El concepto de front-end es fácil de explicar, se define como todo componente, texto, imagen etc., que el cliente puede ver e interactuar desde su equipo, considerándose a equipo como todo dispositivo que pueda ejecutar un navegador web.

VueJS

VueJS es un conjunto de reglas, procedimientos y criterios que se usan para construir la capa de visualización de una aplicación web, dentro los conceptos de los autores tenemos a (Villalaín, 2019) declarando que “VueJS se hace llamar a sí mismo un framework progresivo para el desarrollo de interfaces Web. Lo que esto quiere decir es que la plataforma cimienta su funcionamiento sobre la generación de código HTML a partir de información dinámica”, dicho concepto generaliza la forma en la que VueJS funciona.

VueJS nació bajo el concepto de ser una librería muy completa y sofisticada, desde su lanzamiento el 2014, la comunidad lo transformo en lo que es actualmente, incluso llegando a contar en la actualidad con dos versiones estables y una versión aun en desarrollo. Su creador Evan You participo en proyectos importantes en google, y apporto en el desarrollo de Angular.

Inspiración de otras herramientas

La magia que ofrece este framework no radica en que todas sus herramientas, librerías o funciones son nuevas, ni revolucionarias; este framework combina en sí mismo lo mejor de un framework y una librería ya creada:

React.

DOM (Document Object Model). - Esta función presente en React e implementada en VueJS, permite crear un puente entre el estado de la aplicación y el DOM real de la aplicación. Esto se detalla mejor en la fig.

Reactividad.-Esta característica proveniente de React y que está estrechamente vinculado a la utilización del DOM, la implementa VueJS permitiendo que cada componente se actualice al momento de cambiar un dato. (Filipova, 2016).

Angular.

Directivas.- Las directivas de angular que son implementadas en VueJS hacen referencia a la sintaxis del código que se usan tanto en VueJS como en angular para ejecutar estructuras selectivas y de control, para el renderizado de componentes en la vista o pantalla, el autor (Macrae, 2018) considera a estas directivas como el corazón de VueJS, aunque estas directivas le dan un gran potencial a VueJS, existen más componentes que fomentan su usabilidad

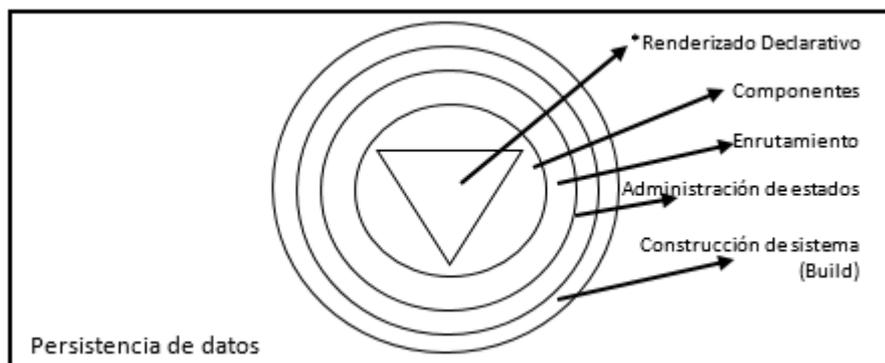
Los conceptos que se manejan en el presente caso de estudio abarcan las versiones de VueJS desde la versión 0.6 hasta la versión 2.6, la última versión lanzada en 2020, se obvia ya que todavía, se encuentra siendo desarrollada y perfeccionada.

VueJS como framework progresivo

VueJS se autoproclama como el framework progresivo, lo cual lo hace idóneo para desarrollar el front end de una aplicación web. Un framework progresivo es aquel que gracias a su modularidad y estructura permite ser usado en cualquier parte del proyecto, en cualquier tipo de proyecto web, sea cual sea el lenguaje que este enfocado, es decir, mediante la implementación de una línea de script en una página HTML sencilla podemos obtener todas las ventajas de este framework.

Figura 1

Capas que hacen de VueJS un framework progresivo, estable y poderoso



Fuente: Propia

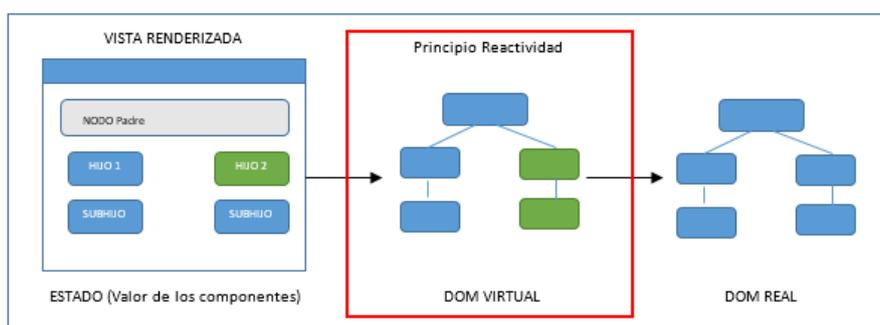
En la figura 1 se puede ver desglosado en forma de capas la estructura de VueJS, casi a nivel jerárquico, se puede, observar como su corazón o motor se basa en el renderizado declarativo este hace referencia a la manera que construye los componentes en pantalla, la segunda capa en la estructura de VueJS, hace referencia a los componentes que VueJS ofrece para construir las vistas.

La tercera capa se enfoca al enrutamiento, aunque VueJS está dirigido a aplicaciones de una sola página, también brinda una capa de enrutamiento que permite de mejor manera organizar el contenido en la aplicación. En la cuarta capa encontramos la administración de estado que de manera resumida se puede definir como un almacén de variables. La última capa una de la más importante hace referencia a la manera en que VueJS empaquetara o construirá las demás capas en una sola, para poder ser desplegada en un servidor.

Antes de detallar a breves rasgos que es el renderizado declarativo, se debe dejar en clara que es un renderizado según (Luna, Millahual, & Iacono, 2017) "Se trata del proceso que ejecutan los motores de navegadores web, leen en tiempo real los códigos (...) para transformarlos en componentes visuales". En la figura 2 se puede apreciar como este funciona y se integra con el DOM.

Figura 2

Renderizado y principio de reactividad



Fuente: Propia.

Una vez interpretados los códigos y transformados a componentes visuales VueJS liga ese componente a un estado que representa su valor, el principio de reactividad consiste en el actualizado automático de cada componente y sus componentes hijos, cada vez que su estado se modifique.

El DOM Virtual representa la gran ventaja que VueJS cuenta por encima de otro framework, esta tecnología traída desde React, permite a VueJS, no perder tiempo ni rendimiento al cargar todo el árbol de nodos o componentes.

Los componentes en VueJS son objeto que pueden ser reutilizados, y pueden ser considerados desde un botón, hasta un conjunto de objetos. Estos componentes cuando son renderizados son organizados en estructuras similares a árboles respetando su herencia y jerarquía (Véase figura 2 el principio de reactividad-DOM virtual).

Los componentes forman la capa de diseño y están ligados a la estructura visual de una aplicación web entonces es importante definir los aspectos primordiales de los componentes, según (Hong, 2018) en el diseño podemos encontrar factores como Rejillas (grids), iconos, tipografía, colores, usabilidad, consistencia.

Todas las herramientas detalladas en el anexo 2 (tabla 4), coinciden con los factores primordiales para un buen diseño propuesto por (Hong, 2018) en su libro, es importante contar con un amplio abanico de donde escoger al momento de diseñar o maquetar el front-end de una aplicación web, en la tabla 1, solo detallamos las herramientas más conocidas.

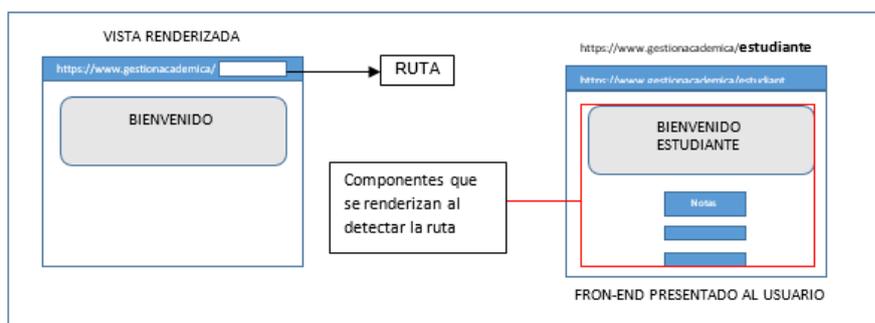
La capa de enrutamiento es aquella, que nos permite organizar mediante el uso de url's, las vistas o componentes que mostraremos en la página, una definición traducida "This library allows developers to customize the URL pathing of their application and also control the navigation security of it" [Esta biblioteca permite a los desarrolladores personalizar la ruta URL de su aplicación y también controlar la seguridad de navegación] (Tran, 2020) esta capa es fundamental dentro de la seguridad de una aplicación web, ya que permite detener accesos indebidos a vistas destinadas a otros tipos de roles.

No se debe confundir esta funcionalidad de vuejs con la funcionalidades que ofrece el back-end para responder a las peticiones hechas por el mismo front-end, esta funcionalidad se

limita al front-end y contribuye a la seguridad en la aplicación por que permite no renderizar ciertos componentes en la pantalla, si no se renderizan no se podrán acceder a esos datos.

Figura 3

Funcionamiento del enrutamiento en VueJS



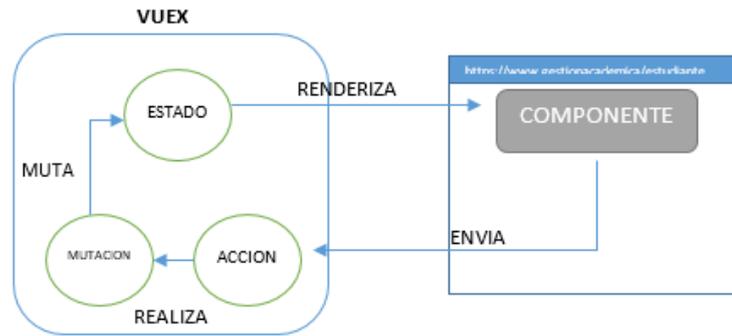
Fuente: Propia

Para lograr el enrutamiento Vue cuenta con una biblioteca oficial llamada vue-route, esta biblioteca aporta con todos los recursos necesarios para renderizar mostrar y actualizar, todos los componentes, existen enrutadores propuestos por la comunidad tales como page.js y director.

La administración de estado en conceptos amplio se refiere a la manera en la que VueJS manipulara los datos entre componentes estén o no relacionados, generando una especie de almacén donde los componentes podrán acceder para recuperar sus datos con respecto a la definición “Vuex looks to help us achieve better state management by enforcing a centralized store, essentially a single source” [Vuex busca ayudarnos a lograr una mejor administración del estado imponiendo una tienda centralizada, esencialmente a una sola fuente] (Halliday, 2018).

Figura 4

Diagramas de procesos en Vuex



Fuente: Propia

La figura 4 muestra la interacción y la manera en la que Vuex administra los procesos, se puede apreciar como el principio de reactividad se vuelve hacer presente, de una manera intrínseca.

La construcción de sistemas o Build System es la función que implementa VueJS para la compilación y producción del front-end. Esta función no solo está centrado en la producción sino también se centra en la manera en la que VueJS organiza los proyectos a gran escala, de acuerdo a sus capas de estructura VueJS. La persistencia de datos es otra ventaja importante que implemente VueJS. Esta función permite que los datos se guarden en el navegador, para que puedan ser usados por los componentes dentro de la vista creada por VueJS.

La persistencia de datos es una pieza clave dentro del rendimiento y la filosofía de desarrollar en VueJS, cualquier aplicación web debe contar con un sistema persistente de datos, para optimizar los flujos de peticiones entre back-end y front-end.

Metodología

La metodología a usar en la resolución de este caso de estudio será la metodología de investigación cualitativa, esta metodología nos permite llegar a la solución de un problema, a través del planteamiento de una hipótesis, el análisis de datos no numéricos, y el uso de la

experiencia del investigador, la recolección de datos no numéricos lo hacemos mediante la entrevistas a los sujetos que participan en el fenómeno.

Diseño de la investigación

El diseño de la investigación planteada para determinar la factibilidad de uso, es un diseño de investigación no experimental, bibliográfica, debido a que no se intervendrá ninguna variable y las condiciones de los hechos no serán cambiantes, también para profundizar en los temas se realizará una investigación teórica. Este estudio de factibilidad se aleja mucho de estudiar los requerimientos funcionales del cliente, enfocándose directamente a los requisitos no funcionales y sus enfoques (Velocidad, seguridad, diseño y rendimiento) contrastando esta información, con la información técnica reunida acerca de la herramienta VUEJS, participando en esta información sus ventajas, desventajas debilidades y fortalezas.

Población y muestra

La entrevista aplicada en el presente caso de estudio está dirigida hacia el responsable de la coordinación académica de la institución. La muestra a usar es el individuo con mayor representatividad dentro de la institución.

Técnicas de recopilación de información

La entrevista abierta es una técnica usada para la extracción de información no numérica, textual, esta técnica se utilizó para la recopilación de la información acerca del caso de estudio. Debido a que no existen datos cuantitativos para evaluar y tabular; la entrevista es el instrumento perfecto para extraer los requerimientos o “cualidades” sobre la aplicación.

En un primer momento, el desarrollador debe escuchar y observar tratando de descubrir el máximo de información; a continuación, deberá interrogar y tratar de aclarar al máximo la información recogida; seguidamente el desarrollador deberá comprobar la

información y sugerir soluciones; finalmente una vez comprendido lo suficiente el problema escribir el documento con la especificación de requisitos. (Noriega Martínez, 2017)

Con respecto al concepto citado, en la sección de anexos (Anexo 1) se encuentra la entrevista con su respectivo cuestionario; la utilización de esta técnica y el planteamiento de las preguntas se enfocan a la extracción de requerimientos de la aplicación web de la UEMASJ.

Determinación de requerimientos funcionales y no funcionales

Los procesos y requerimientos atendiendo al concepto de (Avilés Matute, Avila-Pesantez, & Avila, 2020) “son importantes para la organización. Por tanto, es necesario la implementación de un sistema web eficiente, robusto y funcional, que proporcione la seguridad y disponibilidad de la información, así como, la eficiencia de desempeño”. En la sección de anexos se encuentra la entrevista conjuntamente a sus respuestas, desde la pregunta 1 hasta la pregunta 5 son preguntas dirigidas a la obtención de requisitos funcionales. Desde la pregunta 6 hasta la pregunta 12, son preguntas que se enfocan a los requerimientos no funcionales. Obteniendo así los siguientes requerimientos:

Requerimientos funcionales.

- Registro de usuarios siendo estos tales como: personal docente, personal administrativo, y estudiantes.
- Creación, aperturas y cierres de periodos lectivos.
- Creación de asignaturas y cursos.
- Registro de calificaciones de los estudiantes de acuerdo a su asignatura y curso
- Imprimir informes de calificaciones
- Imprimir listas de estudiantes vinculados a cursos

Requerimientos no funcionales.

- Las páginas de la aplicación deben tener una rápida velocidad de carga.
- La aplicación debe manejar un modelo SPA (aplicación de página única)
- El diseño de la aplicación debe ser limpia e intuitiva (no contener tantos elementos visuales)
- Contar con un diseño responsive (Adaptable para cualquier tamaño de pantalla)
- Seguridad en el acceso a las páginas
- Ejecución en la mayor cantidad de navegadores

Proceso para determinar la factibilidad del framework VueJS

Un importante distintivo del presente caso de estudio es su conclusión, para llegar a esta, se usara el trabajo en conjunto de dos modelos de estudios; estudio de factibilidad y factibilidad de uso. El primero se centra en detallar o estudiar la factibilidad técnica, operativa y económica. Mientras que el segundo se limita a demostrar si es válido o no usar esta herramienta para el desarrollo, en base al cumplimiento de los requerimientos no funcionales y los enfoques estructurales del front end.

Factibilidad de uso**Especificaciones técnicas del front end de la aplicación.**

Como métrica para determinar las especificaciones técnicas del front end se plantea cuatro enfoques específicos, esto se crea debido a la dificultad de evaluar la calidad que el front end obtendrá.

Lo ideal sería poder medir la calidad del software como se miden ciertos aspectos de calidad de otros productos (...). Desgraciadamente esto no resulta fácil, y las técnicas

de aplicación de métricas precisas a los productos de software están todavía en evolución. (Gómez Palomo & Moraleda Gil, 2020)

El autor anterior hace énfasis en la dificultad de evaluar un producto de software, para obtener un front end de calidad que cumpla con los requisitos no funcionales, es necesario que la herramienta para su creación abarque de la manera más óptima los cuatro enfoques propuestos:

- **Velocidad-conectividad.**-Este enfoque está orientado a la velocidad y conectividad, que el front-end debe establecer con el back-end, también se enfoca en medir la tolerancia que el usuario final tendrá para el cargado de su página, es decir cuánto tiempo esperara que su página se cargue, antes de abandonar el sitio.
- **Diseño.**-Este enfoque está dirigido a la ubicación de los componentes, sus colores, sus tamaños y la gran variedad de forma de combinarlos.
- **Seguridad.**-Aunque este enfoque este mejor optimizado y sea más usado en el back-end, en el front-end también se cuenta con pequeñas interacciones de seguridad, tales como accesos indebido a páginas y roles, este enfoque está estrechamente ligado al back-end.
- **Rendimiento-interacción.**- Este enfoque abarca la manera en la que la front-end se comportara en diversos navegadores o dispositivos.

Las relaciones que guardan estas especificaciones del front end directamente con la estructura de capas de VueJS se especifican en el anexo 3 (tabla 5).

Fortalezas y ventajas de VueJS como herramienta de desarrollo.

- El termino framework progresivo por sí mismo es una ventaja, VueJS es un framework que se puede aplicar a cualquier proyecto que tenga implementado otras tecnología, tan solo con importar VueJS traído desde un CDN (Content Delivery Network).

También se puede crear proyectos desde cero pensados en escalar su tamaño y sus funcionalidades.

- No crear nuevas cosas, aunque contradictoriamente el no haber creado nada nuevo, le da una ventaja, puesto que VUEJS logro tomar lo bueno de todas las tecnologías creadas y las transformo en una herramienta más fácil de usar, sin perder robustez.
- Su entorno para el desarrollo de front-end genera un buen ambiente, sus herramientas son intuitivas y fáciles de usar, obviamente para personas familiarizadas con los conocimientos en el desarrollo front-end.
- Su reactividad es un framework que se orienta a brindarle a su front–end información constantemente actualizada, bajo esta orientación los desarrolladores solo deben preocuparse en su propósito, generar un buen diseño e una buena interacción con el usuario.
- El manejo de sus componentes, archivos y ficheros, la organización que brinda el entorno de desarrollo para el desarrollador front-end es muy limpia y clara.
- Su comunidad, al tener una curva de aprendizaje baja, muchos desarrolladores se suman día a día a esta comunidad desarrollando nuevas herramientas y resolviendo problemas.

Debilidades y desventajas de VueJS como herramienta de desarrollo.

- La robustez de sus herramientas no oficiales, al ser un framework libre y escalado por la comunidad, sus herramientas complementarias no reciben apoyo de los grandes exponentes en el desarrollo web.
- El statu quo en el desarrollo de front-end, esta desventaja aunque ajena a la propia herramienta, pone a VueJS fuera de la mira de grandes empresas o proyectos, los desarrolladores tiene una tendencia de mantener las mismas tecnologías.

Factibilidad de implementar o desplegar un front end desarrollado con VueJS

Situación actual.

La Unidad Educativa Miguel Ángel Samaniego Jiménez, no cuenta con una aplicación web desplegada para gestionar sus actividades académicas, el presente caso de estudio se dirige en determinar qué tan factible es usar a VueJS para el desarrollo del front end de la aplicación, para fines prácticos determinaremos la factibilidad de implementar este front end, es decir determinaremos los recursos necesarios para poder ejecutar esta aplicación web.

Factibilidad técnica.

En la tabla 1 se describe las especificaciones técnicas que un equipo debe poseer para ejecutar una aplicación desarrollada en VueJS, se detalla las especificaciones mínimas de un equipo para poder ejecutar un navegador web con motor de renderizado compatible con las tecnologías de VueJS.

Tabla 1

Especificaciones técnicas para ejecutar el front end construido con VueJS

NIVEL HARDWARE	NIVEL SOFTWARE		
EQUIPO MINIMO	SISTEMAS OPERATIVOS	NAVEGADORES	MOTORES DE RENDERIZADO
Velocidad de procesador 1GHz Memoria RAM 2Gb Espacio en disco 32 Gb	WINDOWS 10	Google Chrome 88.0.4324.190	BLINK, V8, WEBKIT
		Firefox 86,0	GECKO, SERVO
		Microsoft edge	BLINK
	LINUX	CHROMIUM 88,0	BLINK, V8
		Firefox 86,0	GECKO, SERVO
		Iceweasel 32,2	GECKO

Fuente: Propia

Factibilidad operativa.

Para el uso correcto del sistema diseñado con VueJS, las personas que operen la aplicación deben poseer conocimientos básicos acerca de informática, además de contar con conocimientos sobre la navegación web. Los usuarios finales deben poseer conocimientos prácticos sobre la aplicación y su funcionamiento. Para que el usuario final pueda llevar a cabo la realización de los procesos de una forma adecuada el front end debe contar con una vista intuitiva, fácil de usar. La mayoría de procesos que se pretenden abordar, estarán realizados por el recurso humano de la institución. El recurso humano de la institución se encuentra capacitado en las áreas de informática y posee conocimientos de navegación web, de igual manera, la aplicación web contara con su respectivo manual de usuario.

Factibilidad económica.

En la tabla 2 se recoge los precios aproximados de los equipos de cómputo necesarios para ejecutar la aplicación web desarrollado con VueJS, cabe destacar que en la tabla se

especifica, los equipos mínimos para ejecutar el front end. Todo equipo que se encuentre por encima de estas especificaciones, será útil para poder ejecutar el front end desarrollado con VueJS.

Tabla 2

Precios referenciales de los equipos para ejecutar el front end construido con VueJS

Equipos de computo		Precio promedio
<i>CPU</i>		
Velocidad de procesador 1GHz Memoria RAM 2Gb Espacio en disco 32 Gb	Equipos nuevos de mayor prestaciones y de última generación	350
<i>MONITOR</i>		
19 Pulgadas	Con una resolución mínima de 768p y entradas a VGA o HDMI	140
<i>TECLADO</i>		
USB	Cualquier teclado que posea una entrada USB	20
<i>MOUSE</i>		
OPTICO	Cualquier mouse óptico con conector USB	20
Total en equipos de cómputo:		530
Nota.-Los precios sugeridos se calcularon en base de una búsqueda de equipos en tiendas como Novicompu, Computron, Advance, Tecno mega Store entre otras.		

Fuente: Propia

Tabla 3

Precios referenciales acerca de los planes de internet para acceder al front end construido con VueJS

INTERNET		Precio
<i>PAGO MENSUAL</i>		
10 Mb/s	Corporación nacional de telecomunicaciones 60\$ instalación	27,89
25 Mb/s	Claro Gratis instalación	23,51
60 Mb/s	Netlife	40,88
Nota.- Todos los datos consultados corresponden a la publicidad difundida por las compañías en la web, están sujetas a cambio.		

Fuente: Propia

En la tabla 3 se detallan los valores de los planes de internet fijo, cabe destacar que con una conexión de 10Mb/s de velocidad, la aplicación se ejecutará sin problemas. Todos los montos en la tabla 2 y la tabla 3, son montos despreciables, y sirven de referencia para la adquisición de nuevos equipos en caso de ser necesario.

CONCLUSIONES

- Dentro de la presente investigación se determinaron las especificaciones necesarias para un buen front end, resaltando entre ellos un buen diseño y un correcto control de seguridad. Extrayendo los requerimientos funcionales los cuales se enfocan a las funciones primordiales que cumplirá el servidor de aplicación (back end) y por otra parte los no funcionales que se centran en la manera de cómo el front end protegerá, manipulara y mostrara los datos en la vista del lado del cliente.
- El framework VueJS autodenominado como framework progresivo cumple con los requisitos para generar un front end de calidad para la institución, partiendo desde las especificaciones demandadas por la institución. Es factible usar como herramienta de

desarrollo, puesto que toda su estructura, entorno de desarrollo, su familiaridad con otras herramientas, abundancia de componentes y librerías, lo hace uno de los más idóneos para ser adoptado como herramienta para producir este front end. Aunque su mayor debilidad sea la robustez de librerías no oficiales, sus librerías oficiales están construidas de manera óptima; cumpliendo con las especificaciones propuestas.

- Con respecto a su factibilidad de implementación, la factibilidad técnica para implementar un front end desarrollado con VueJS no es de la más exigente, puesto que hoy en día los motores de renderizado para los navegadores se encuentran muy bien optimizados. Las personas involucradas en el uso de la aplicación, deberán contar con conocimientos básicos de computación y conocimientos específicos sobre la aplicación, este será alcanzado mediante capacitaciones, manuales de usuarios o técnicas a fin. De igual manera para ejecutar la aplicación desarrollada con VueJS no es necesario la adquisición de nuevos equipos, todos se encuentran por encima de los requerimientos mínimos sin embargo el costo de su desarrollo se encontrara determinado por la demanda de este framework.

BIBLIOGRAFÍA

- Avilés Matute, S., Avila-Pesantez, D., & Avila, M. (2020). Desarrollo de sistema Web basado en los frameworks de Laravel y VueJs, para la gestión por procesos: Un estudio de caso. *Revista Peruana de Computación y Sistemas 2020*, 3-10.
- Filipova, O. (2016). *Learning Vue.js 2*. Packt Publishing.
- Gómez Palomo, S. R., & Moraleda Gil, E. (2020). *Aproximación a la ingeniería del software*. Editorial Centro de Estudios Ramon Areces SA.
- González, G. A. (2019, Noviembre). *Significado*. Retrieved from Significado: <https://significado.com/cliente-servidor/>
- Halliday, P. (2018). *Vue.js 2 Design Patterns and Best Practices: Build enterprise-ready, modular Vue.js applications with Vuex and Nuxt*. Packt Publishing Ltd.
- Hong, P. (2018). *Practical Web Design: Learn the fundamentals of web design with HTML5, CSS3, Bootstrap, jQuery, and Vue.js*. Packt Publishing Ltd.
- LOPEZ GOMEZ, Y. T., & CÁCERES ESCUDERO, J. L. (2019). Creación de un FRAMEWORK de desarrollo de aplicaciones web, basado en JAVASCRIPT, HTML y CSS. *CIINATIC Congreso internacional de innovación y apropiación de las tecnologías de la información*, 3.
- López Sanz, M. (n.d.). *Programación web en el Entorno Cliente. (MF0491_3)*. Grupo Editorial RAMA.
- Luna, F., Millahual, C. P., & Iacono, M. (2017). *PROGRAMACION WEB Full Stack 1 - Ecosistema Web: Desarrollo frontend y backend - Curso Visual y Práctico*. RedUsers.
- Macrae, C. (2018). *Vue.js: Up and Running: Building Accessible and Performant Web Apps*. O'Reilly Media, Inc.
- Noriega Martínez, R. (2017). *El Proceso de Desarrollo de Software: 2ª Edición*. IT Campus Academy.

Stefaniak, P. (2019, 07 26). *descubre COMUNICACIÓN*. Retrieved from *descubre*

COMUNICACIÓN: <https://descubrecomunicacion.com/que-es-backend-y-frontend/>

Tran, N. (2020). APPLYING VUE.JS FRAMEWORK IN DEVELOPING WEB APPLICATIONS

(Bachelor's thesis).

VALDIVIA MIRANDA, C. (2017). *Informática industrial*. Ediciones Paraninfo, S.A.

Villalaín, J. S. (2019). *Desarrollo de servicios web para consulta de datos en entornos médicos*.

CARTAGENA.

ANEXOS

Anexo 1

CUESTIONARIO Y RESULTADOS

Objetivos: Determinar las preferencias más importantes del cliente o usuario, para con base a estas preferencias realizar la debida recopilación de información acerca los requerimientos y posibles usos del framework de desarrollo VUE.js

Enfoque: usuario

1.- Pregunta: Cuáles son los usuarios objetivos que usaran la aplicación.

Respuesta

Estudiantes profesores y el personal que brinda servicio dentro de la institución.

Idea principal: Usuarios de la aplicación

Explicación de la pregunta: Mencionar los usuarios que participarán en la aplicación, todo aquel usuario que tendrá una interacción con la aplicación.

2.- Pregunta: Cuáles son las necesidades para cada usuario descrito.

Respuesta

Los estudiantes necesitan poder registrar sus datos y visualizar sus notas, mientras que los profesores necesitan poder organizar las notas y subir las calificaciones de los estudiantes. El resto de personal necesita desempeñar funciones como crear cursos, periodos, y vincular a los estudiantes.

Idea principal: Roles de los usuarios, requerimientos de los usuarios

Explicación de la pregunta: Explicar a breves rasgos cuales son las necesidades para cada usuarios

Enfoque: proceso.

3.-Pregunta: En base a qué necesidad se planteó el desarrollo de una aplicación web para la gestión académica.

Respuesta

En base a la necesidad de agilizar los procesos y centralizar la información.

Idea principal: Planteamiento del problema.

Explicación de la pregunta: Explicar cuál fue la necesidad principal por la cual se pretende realizar una aplicación web.

4.-Pregunta Actualmente, ¿cómo se manejan los procesos dentro de la institución?

Respuesta

Actualmente los procesos se realiza de manera manual, y con la combinación de herramientas de ofimática y algunos software de terceros

Idea principal: Situación actual

Explicación de la pregunta: Se debe explicar la manera en la que actualmente se siguen los procesos por ejemplo manual, en hoja de datos, con software de terceros

5.-Pregunta: ¿Cuáles son las actividades o procesos que se pretende abordar con la creación de aplicación para la gestión académica?

Respuesta

En la institución se realizan el ingreso de los datos del estudiante y representante, también se necesita registrar la información del personal que trabaja dentro de la institución. Se debe crear los periodos lectivos y generar los cursos para poder agregar los estudiantes, y poder registrar las notas académicas para cada estudiante. Cada profesor debe poder subir las notas de manera independiente a cada estudiante y para cada estudiante se debe poder generar un reporte de notas.

Toda esa información se debe generar en constante físico es decir hojas, así como poder obtener listas actualizadas de los estudiantes que pertenecen a cada curso.

Idea principal: Requerimientos funcionales.

Explicación de la pregunta: Explicar las actividades que deberá realizar el software y la interacción con los usuarios, de manera general a modo de narración.

Enfoque: front end.

6.-Pregunta: De los siguientes criterios importantes: Velocidad, Diseño, Seguridad, Rendimiento.

Cual considera usted de mayor importancia y cuál cree usted que es el de menor importancia.

Respuesta

La seguridad la considero como de mayor importancia, un buen diseño, y rendimiento

Idea principal: Enfoques para un buen front end.

Explicación de la pregunta: Según su perspectiva en que criterio se debe poner más énfasis para desarrollar el front end.

7.-Pregunta: ¿Medido en segundos cuanto tiempo usted esperaría la carga o recarga de su página, independientemente de factores tales como la velocidad de internet o la capacidad técnica de su pc (asumiendo que posee una buena conexión de internet, y posee un computador que cumpla los requisitos mínimos para ejecutar un navegador web)?

Respuesta

Esperaría un mínimo de 15 segundos en caso de que el internet y mi computador este bien.

Idea principal: Tolerancia y latencia

Explicación de la pregunta: Determinar el grado de paciencia que el usuario final tendrá al momento de navegar en la aplicación, para considerarlo en el desarrollo del front end

8.-Pregunta: ¿Usted como usuario final, cree que la actualización de los datos en la aplicación sea constante y automática o se manual?

Respuesta

Estoy de acuerdo con que los cambios se hagan de manera automática, y de forma constante para ahorrar tiempo.

Idea principal: Reactividad.

Explicación de la pregunta: Determinar cuál es la manera en la que usted prefiere por ejemplo, si cambia la foto de perfil de usuario que prefiere, que el cambio sea inmediato en su perfil o tenga que actualizar la página.

9.-Pregunta: ¿Cree usted que una página web debe estar cargada visualmente de elementos (botones, imágenes, links, etc.)?

Respuesta

Creo que una página intuitiva es aquella que tiene pocos elementos.

Idea principal: Diseño

Explicación de la pregunta: Explicar si prefiere una aplicación con muchos objetos o una página con objetos intuitivos y fáciles de navegar por la página.

10.-Pregunta: Usted como usuario final ¿cree que es necesario que la aplicación web para gestión académica, debería contar con un diseño específico para ejecutarse en equipos móviles?

Respuesta

Si creo que es útil y necesario poder ejecutarla en un celular, en muchas ocasiones no se cuenta con equipos de cómputos pero si con equipos móviles.

Idea principal: Diseño responsive

Explicación de la pregunta: Determinar si cree que es útil que la aplicación pueda ejecutarse de una manera correcta y optima en un equipo móvil

11.-Pregunta: La seguridad para cualquier aplicación es fundamental, de acuerdo a su criterio usted ¿cómo considera este enunciado?

Respuesta

Creo que la seguridad en todos los aspectos es importante, y mucho más en una aplicación en la que se maneja las notas y datos personales de los estudiantes

Idea principal: Roles y seguridad URL

Explicación de la pregunta: Determinar si es necesario que aparte de la seguridad que la aplicación cuenta, se debe implementar seguridad extra en el front end, como por ejemplo prohibir el ingreso no autorizados a rutas.

12.-Pregunta: Usted como usuario final con respecto al equipo técnico (Computador, portátil Tablet o celular) ¿Cree o tiene conocimiento si su equipo podrá ejecutar sin problemas un navegador web?

Respuesta

Si los dispositivos con los que cuento ejecutan correctamente navegadores web, además la institución cuenta con sala de computo con equipos.

Idea principal: Rendimiento, especificación técnica, interacción

Explicación de la pregunta: Explicar en su respuesta si cree o esta seguro que sus dispositivos con los que cuenta, tienen capacidad para ejecutar navegadores web tales

como google Chrome, Mozilla, safari o incluso navegadores predeterminados por diversos dispositivos móviles.

Anexo 2

Tabla 4

Herramientas de terceros más usadas para el diseño de front-end en VueJS

TIPO DE HERRAMIENTA	NOMBRE	Función
Framework	Eagle.js	Permite la creación de animaciones slides y temas
Librería de iconos	Font Awesome Vue-fontawesome	Contiene estilos css e iconos vectoriales para los componentes de VueJS
Framework	Bootstrap-Vue VueStrap	Posee gran variedad componentes para el diseño responsive para el front end
Framework	Vuetify	Competidor directo para BootStrap, con componentes similares a los de BootStrap.
Framework	Buefy	Es la adaptación del framework Bulma CSS más Vue.
Administrador de plantillas	CoreUI	Administrador de plantillas, permite crear muchos componentes, con compatibilidad a frameworks como BootStrap y Vuetify
Librería de componentes	Element UI	Es una librería con componentes para la representación de datos, y control de datos

Fuente: Propia

Anexo 3

Tabla 5

Matriz de contraste

Enfoque Capas VueJS	VELOCIDAD- CONECTIVIDA D	DISEÑO	SEGURIDAD	RENDIMIENTO- INTERACCION
Renderizado declarativo	Reactividad	Transformación de código a componentes visuales		Enlaza los componentes a su estado y al DOM real, actualización instantánea.
Componentes	Reusabilidad de componentes	Personalización de objetos o grupos de objetos (bloques de componentes HTML)		Cada componentes maneja su propio estado
Enrutamiento		Permite construir vistas para cada URL	Permite determinar roles de acceso mediante la URL	El manejo de rutas ayuda al rendimiento del front-end, con cada ruta se renderizan componentes específicos
Administración de estados	Reactividad	Renderización de componentes al cambiar estado	Mantener datos cifrados, demás información de cuidado	Cada componente reacciona en función de un estado

Fuente: Propia

Anexo 4



UNIVERSIDAD TÉCNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA
DECANATO

Babahoyo, febrero 22 de 2021
D-FAFI-UTB-048-UT-2021

Ledo.

Javier Humberto Samaniego Pérez, MSc.

RECTOR DE LA UNIDAD EDUCATIVA MIGUEL ÁNGEL SAMANIEGO JIMÉNEZ
La Unión.-

De mis consideraciones:

La Universidad Técnica de Babahoyo y la Facultad de Administración, Finanzas e Informática (FAFI), con la finalidad de formar profesionales altamente capacitados busca prestigiosas Empresas e Instituciones Públicas y Privadas en las cuales nuestros futuros profesionales tengan la oportunidad de afianzar sus conocimientos.

El Señor **ROBLES HERRERA DIDIER EMMANUEL**, con Cedula de Identidad N° 120714492-2, Estudiante de la Carrera de Ingeniería en Sistemas, matriculado en el proceso de titulación en el periodo Noviembre 2020 – Mayo 2021, trabajo de titulación modalidad Estudio de Caso para la obtención del grado académico profesional universitario de tercer nivel como **INGENIERO EN SISTEMAS**. El Estudio de Caso: **ESTUDIO DE FACTIBILIDAD DE USO DEL FRAMEWORK VUE.JS COMO HERRAMIENTA, PARA EL DESARROLLO DEL FRONT END DE UN APLICATIVO WEB, PARA LA GESTIÓN ACADÉMICA EN LA UNIDAD EDUCATIVA MIGUEL ÁNGEL SAMANIEGO JIMÉNEZ EN LA PARROQUIA LA UNIÓN.**

- Es por esta razón, solicito a usted, si es posible se sirva autorizar el permiso respectivo para que el señor Robles pueda desarrollar la investigación en la institución de su acertada dirección.

Por su gentil atención al presente, se extiende el agradecimiento institucional.

Atentamente.

Ledo. Eduardo Galeas Gaijarro MAE.
DECANO

c.c. Archivo

Recibido
12:45 - 10/03/2021
[Firma]
[Firma]
[Firma]

