

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingenieras de Sistemas**

**TEMA:
UTILIZACIÓN DE TECNOLOGÍAS DEL LADO DEL CLIENTE PARA
DOTAR DE CAPACIDADES DE PERSONALIZACIÓN VISUAL A UNA
APLICACIÓN FRONT END QUE INTERACTÚE CON MOODLE**

**AUTORAS:
PAOLA ALEXANDRA OLMEDO VILCA
MISHELL ARACELY ROSERO ALCOSER**


**TUTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

Quito, febrero del 2018

CESIÓN DE DERECHOS DE AUTOR


Nosotras, PAOLA ALEXANDRA OLMEDO VILCA, con documento de identificación N° 1718597451, y MISHELL ARACELY ROSERO ALCOSER, con documento de identificación N° 1721745444, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autoras del trabajo de titulación con el tema: “UTILIZACIÓN DE TECNOLOGÍAS DEL LADO DEL CLIENTE PARA DOTAR DE CAPACIDADES DE PERSONALIZACIÓN VISUAL A UNA APLICACIÓN FRONT END QUE INTERACTÚE CON MOODLE”, mismo que ha sido desarrollado para optar por el título de INGENIERAS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autoras nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
PAOLA ALEXANDRA
OLMEDO VILCA

CI: 1718597451



.....
MISHELL ARACELY
ROSERO ALCOSER

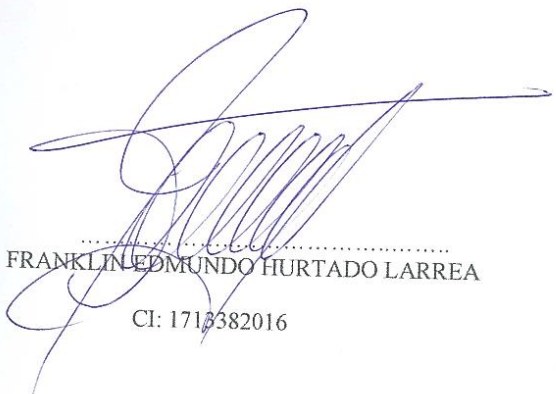
CI: 1721745444

Quito, febrero del 2018

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “UTILIZACIÓN DE TECNOLOGÍAS DEL LADO DEL CLIENTE PARA DOTAR DE CAPACIDADES DE PERSONALIZACIÓN VISUAL A UNA APLICACIÓN FRONT END QUE INTERACTÚE CON MOODLE” realizado por las estudiantes PAOLA ALEXANDRA OLMEDO VILCA y MISHELL ARACELY ROSERO ALCOSER, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser consideradas como trabajo final de titulación.

Quito, febrero del 2018



FRANKLIN EDMUNDO HURTADO LARREA
CI: 1713382016

Dedicatoria

Para empezar, quiero dedicar este trabajo de titulación a Dios, por haber sido mi guía en mi diario caminar y darme su fuerza para continuar en cada paso que he dado y obstáculos que se me han presentado.

A mis padres, quienes han sido un pilar fundamental en mi vida, se las dedico por haber estado conmigo en todo momento y haberme inculcado principios y valores.

A toda mi familia por sus consejos y alientos a no decaer. A mis compañeros y amigos que me he encontrado en este largo camino por la Universidad.

Paola Alexandra Olmedo Vilca

Mi tesis de grado la dedico con toda la fe a Dios, que es mi guía y mi fortaleza.

A mi gatita Jammee Adamari, que desde el cielo es mi compañía e inspiración. ¡Te extraño!

Desde lo más profundo de mi corazón a mis padres, Jaime y Patricia, por su apoyo incondicional, por inculcarme a ser mejor cada día, por confiar en mí y por darme la oportunidad de tener un mejor futuro. A mis amados abuelitos, mi orgullo y ejemplo: Jorge Alcoser, Juanita Guadalupe y Juanita Pineda y, en el cielo, Antonio Rosero.

A mis pequeños: "Shavito, Adriancito, Heidy y Sofy", que llenan de luz, alegría y esperanza a mi vida. A mis tíos y tías: Alcoser-Guadalupe y Rosero - Pineda, por todo su cariño. A todos mis primos, a los que llevo en mi corazón y con los que compartimos ocurrencias, locuras y tristezas: Mari, Karito, Cris y Fer, Shavi, Sofy y Dennis. ¡Gracias a todos, vivirán siempre en mi corazón!

Mishell Aracely Rosero Alcoser

Agradecimiento

Queremos en primer lugar agradecer a Dios, por bendecirnos y guiar nuestro camino para llegar hasta al final de este periodo.

A nuestros padres por darnos la oportunidad de estudiar para ser profesionales honestas e íntegras, al servicio de nuestra Patria.

A nuestra familia por su apoyo incondicional, sabemos que nada sería igual sin ustedes.

A nuestro director de tesis, Ing. Franklin Hurtado, por su esfuerzo y dedicación, por compartir con nosotras sus conocimientos y experiencia, por su paciencia y por motivarnos a terminar nuestra carrera profesional.

También agradecemos a todos nuestros profesores que, a lo largo de la carrera estudiantil, aportaron a nuestra formación profesional.

En estos años son muchas las personas que formaron parte de nuestro paso por las aulas de la Universidad Politécnica Salesiana a las que nos encantaría agradecerles por su amistad, sus consejos, su apoyo, por animarnos a seguir y por acompañarnos en los momentos más difíciles de nuestras vidas. Algunas están aquí con nosotras, otras viven en nuestros recuerdos y corazón, sin importar en donde estén. Para ellos nuestra gratitud por formar parte de nuestra vida, por lo que nos brindaron y por todas sus bendiciones.

Paola Alexandra Olmedo Vilca

Mishell Aracely Rosero Alcoser

ÍNDICE

INTRODUCCIÓN	1
Problema	2
Justificación	2
Objetivo General	2
Objetivos Específicos.....	3
Marco Metodológico.....	3
Adaptaciones a la metodología XP.	3
Capítulo 1	6
1.1. Personalización	6
1.2. Personalización de aplicaciones web	7
1.3. Usabilidad	9
1.4. Usabilidad Pedagógica.....	10
1.4.1. Parámetros a considerarse para el diseño de un sitio web enfocado en la usabilidad pedagógica.	10
1.5. Diseño Web.....	11
1.5.1. Componentes principales de un sitio web.....	14
1.5.2. Diseño web enfocado en la usabilidad del usuario.	16
1.6. Diferencia entre aplicación web y sitio web	19
1.7. Maqueta web	20
1.8. Aplicaciones del lado del servidor	20
1.8.1. Lenguajes de Programación principales del lado del servidor.....	21
1.9. Aplicaciones del lado del cliente.....	23
1.9.1. Tecnologías.	24
1.9.2. Otras tecnologías del lado del cliente.....	26
1.10. Aplicaciones del lado del servidor vs Aplicaciones del lado del cliente	27
1.11. Ejemplos de aplicaciones realizadas del lado del servidor	28
1.12. Frameworks.....	29
1.12.1. ReactJs.	29
1.12.2. AngularJs.	29
1.13. Herramientas	31
1.13.1. JustInMind.	31
1.13.2. WebStorm.	31
1.13.3. Gulp.....	32
1.13.4. Bitbucket.	32
1.14. LMS.	32
1.14.1. Moodle.	33
1.15. Metodología ágil	34
1.15.1. XP (Programación Extrema).	35
Capítulo 2.....	39
2.1. Evaluación de funcionalidades de AngularJs.....	39
2.1.1. Planificación.....	39
2.1.2. Diseño.	40
2.1.3. Prototipo.....	43
2.1.4. Evaluación.....	54
Capítulo 3.....	56
3.1. Primer prototipo funcional	56
3.1.1. Diseño.	57

3.1.2. Prototipo.....	62
3.1.3. Evaluación.....	66
Capítulo 4.....	67
4.1. Prototipo Producto final.....	67
4.1.1. Diseño.....	68
4.1.2. Prototipo.....	69
4.1.3. Evaluación.....	83
Capítulo 5.....	87
5.1. Plan de Pruebas.....	87
5.1.1. Pruebas de medición.....	87
5.1.2. Pruebas de simulador al servidor.....	89
5.1.3. Tabla comparativa entre una aplicación convencional cliente/servidor y una aplicación solo de lado del cliente.....	90
Conclusiones.....	92
Recomendaciones.....	93
Glosario de Términos.....	94
LISTA DE REFERENCIAS.....	95

ÍNDICE DE TABLAS

Tabla 1. Resumen de iteraciones.....	5
Tabla 2. Evolución de la web.....	12
Tabla 3. Lenguajes de programación más usados en el lado del servidor	22
Tabla 4. Evolución de HTML	24
Tabla 5. Alternativas a escoger en el menú “Opciones”	42
Tabla 6. Elementos que forman parte del menú de navegación.....	43
Tabla 7. Función de cada uno de los elementos del drag & drop	48
Tabla 8. Descripción de cada uno de los campos de la tabla "Temas"	61
Tabla 9. Detalle de los campos de la tabla "Estudiante"	62
Tabla 10. Medición de ingreso al sitio	88
Tabla 11. Medir consumo al cargar la ventana de personalización	88
Tabla 12. Medición de recursos a la carga de datos.....	88
Tabla 13. Medición en el tiempo consumido	89
Tabla 14. Comparación de consumo de recursos.....	91

ÍNDICE DE FIGURAS

Figura 1. Perspectivas principales tomadas en cuenta en la elaboración del proyecto	4
Figura 2. Parámetros a considerar para diseñar un sitio web enfocado en la usabilidad pedagógica	11
Figura 3. Componentes principales de un sitio web	14
Figura 4. Proceso para diseñar un sitio web	16
Figura 5. Diferencia entre sitio y aplicación web	20
Figura 6. Tecnologías que permiten crear un sitio web	26
Figura 7. Planificación de acuerdo a los requerimientos	40
Figura 8. Diseño de la presentación del sitio web	41
Figura 9. Ubicación de componentes iniciales	42
Figura 10. Elementos del menú de navegación	43
Figura 11. Seleccionar fondo de pantalla	44
Figura 12. Opciones de fondos de pantalla a agregar	45
Figura 13. Fragmento de código para seleccionar un fondo de pantalla	45
Figura 14. Código que contiene la función a la llamada de los parámetros	46
Figura 15. Ubicación de componentes iniciales	47
Figura 16. Fragmento de código que muestra el movimiento de componentes	48
Figura 17. Funciones del drag y drop	49
Figura 18. Selección de color para los componentes	50
Figura 19. Fragmento de código que permite la llamada a la función	51
Figura 20. Función para la selección de colores	51
Figura 21. Selección de la forma para los componentes	52
Figura 22. Fragmento de código de la vista que llama a la función	53
Figura 23. Función que permite seleccionar el tipo de figura	54
Figura 24. Evaluación de los componentes probados	55
Figura 25. Árbol de estructura de las funciones de AngularJs	57
Figura 26. Diseño de la presentación inicial del sitio web	57
Figura 27. Ubicación de componentes iniciales	59
Figura 28. Diseño de la ventana emergente	59
Figura 29. Presentación del primer prototipo funcional	62
Figura 30. Ventana emergente que presenta la selección de las plantillas	63
Figura 31. Fragmento de código que muestra la ubicación	64
Figura 32. Función que permite mostrar u ocultar las plantillas	65
Figura 33. Fragmento de código que contiene la función	65
Figura 34. Presentación de un sitio web personalizado	66
Figura 35. Diseño definitivo de la presentación del sitio web	68
Figura 36. Usuarios creados en Moodle	69
Figura 37. Listado de materias en Moodle	70
Figura 38. Inicio de sesión de usuario estudiante	70
Figura 39. Fragmento de código que permite inicio de sesión	71
Figura 40. Función que permite el inicio de sesión	72
Figura 41. Fragmento de código que trae el identificador del estudiante	72
Figura 42. Pantalla que muestra opción “Mi Perfil”	73
Figura 43. Fragmento de código que llama a la función	74
Figura 44. Función que permite almacenar los cambios realizados	75
Figura 45. Forma de guardar los componentes	76
Figura 46. Fragmento de la función llamada desde la vista	76
Figura 47. Creación de una nueva plantilla al gusto del usuario	77

Figura 48. Función que permite ubicar los componentes	78
Figura 49. Presentación de la opción “Mis Materias”	79
Figura 50. Componentes de una de las materias	79
Figura 51. Fragmento de código del listado de materias	80
Figura 52. Fragmento de código que muestra el identificador de la materia	80
Figura 53. Comparación de sitio web personalizado y no personalizado	81
Figura 54. Llamada a la ventana de Asistente de personalización.....	82
Figura 55. Ventana de cambios realizados.....	83
Figura 56. Información de la tabla con los cambios guardados.....	84
Figura 57. Comparación del menú principal.....	86
Figura 58. Pruebas realizadas al servidor.....	90

Resumen

Con el fin de aplicar los conocimientos obtenidos durante el transcurso de la carrera de Ingeniería de Sistemas, el presente proyecto tiene como propósito permitir la personalización de un sitio web (aula virtual), lo que demuestra que es más eficiente consumir tecnologías del lado del cliente que del lado del servidor. Razón por la cual se cuenta con objetivos como: insertar un fondo de pantalla en un sitio web, cambiar color y forma de los componentes que son parte de este sitio y cambiar la ubicación de los mismos. Se pretende demostrar que es posible realizar la personalización de un sitio web, ya que el ser humano por naturaleza personaliza todo y por ende puede resultar eficiente personalizar el sitio en el que se encuentra navegando. Para cumplir con estos objetivos se utilizó: AngularJs como framework, que permite trabajar del lado del cliente, junto con la plataforma e-learning Moodle como un proveedor de contenidos, juntos permitieron el desarrollo de este proyecto.

Abstract

Applying the knowledge acquired during the Systems Engineering Career, this project aims to allow the customization of a website (virtual classroom), demonstrating that it is more efficient to consume the technologies on the client side than on the server side. Should consider: insert a wallpaper on a website, change the color and shape of the components and change the location of them. It is going to prove that it is possible to make personalization of a website, and users can customize the site and browse freely with ease. To achieve these objectives: AngularJs as a framework, which allows working on the client's side, together with the Moodle e-learning platform as a content provider, together allowed the development of this project.

INTRODUCCIÓN

La personalización es la identidad de todo ser humano, por lo que ha tratado de personalizar su ambiente más próximo, ya sea su ambiente laboral, su forma de vestir, entre otras, a esta conclusión han llegado varias páginas que promueven el servicio de marketing digital, debido a que un producto puede ser personalizado de acuerdo con los gustos de un individuo.

Los sistemas de información no son la excepción a la personalización, con el creciente incremento de usuarios y la llegada de Internet, ahora es una tendencia que las personas busquen personalizar, sin embargo, existe una limitante en la tecnología cliente/servidor convencional, pero con las tecnologías del lado del cliente, así como son: AngularJs, JavaScript, HTML5, han permitido que se puedan explotar algunas aplicaciones del lado del cliente.

Se ha partido desde un proyecto de titulación, cuyo propósito fue la “Creación de un Front End usando HTML5, para explotar las funcionalidades de la plataforma Moodle”, realizado en la Universidad Politécnica Salesiana, logrando tener como resultado final un aula virtual, al cual, este proyecto le dio una mejora, añadiendo funcionalidades de personalización, consumiendo recursos del lado del cliente.

En el capítulo uno “Marco teórico” se explica los conceptos principales que sostienen al proyecto, en los capítulos dos, tres y cuatro se describen las “Iteraciones” del proceso de desarrollo, donde se analizan los componentes con los que se cuenta y en el capítulo cinco “Plan de Pruebas” comprende las pruebas realizadas de acuerdo a los objetivos planteados con el producto final.

Problema

La Internet ha facilitado realizar cambios tanto en productos o servicios de acuerdo a los gustos de cada individuo, a esos cambios se los llama personalizar, por lo que se pretende con este proyecto dar al usuario la posibilidad de que pueda personalizar un sitio web (aula virtual), acompañado de JavaScript y HTML5, con el objetivo de tener una aplicación Front End y así demostrar que es posible a través de estas tecnologías reducir el consumo de recursos.

Justificación

Con el advenimiento de las tecnologías del lado del cliente, ha empezado a crecer la posibilidad de desarrollar aplicaciones para el usuario, teniendo como ventaja consumir sus recursos y así permitir que estas aplicaciones sean más eficientes, debido a que, normalmente al realizar una aplicación del servidor se consumirán más recursos, ocupando espacio en memoria, creciendo exponencialmente estos resultados, en cambio con el uso de las tecnologías del lado del cliente se consume los recursos únicamente de este, sin embargo este tipo de aplicaciones son poco utilizadas. En el contexto de la personalización es evidente que se van a consumir menores recursos del lado del servidor, al utilizar los datos que el usuario previamente ha guardado. Con el desarrollo de este prototipo se brindará un mejor rendimiento al sitio web y así aprovechar el uso de dichas tecnologías.

Objetivo General

Utilización de tecnologías del lado del cliente para dotar de capacidades de personalización visual a una aplicación Front End que interactúe con Moodle.

Objetivos Específicos

Investigar las características de HTML5 y JavaScript que permitan lograr la personalización más apropiada para el usuario, desde la perspectiva Front End.

Brindar mayor flexibilidad al usuario, con la personalización, ya sea en forma, fondo, color y ubicación de componentes.

Implementar y probar el Front End con capacidad de personalización para determinar la calidad de la arquitectura.

Marco Metodológico

La metodología seleccionada fue XP, ya que es una metodología ágil, siendo aquellas que se caracterizan por su adaptabilidad, es decir permiten la flexibilidad en el cambio de requerimientos o tiempos de entrega.

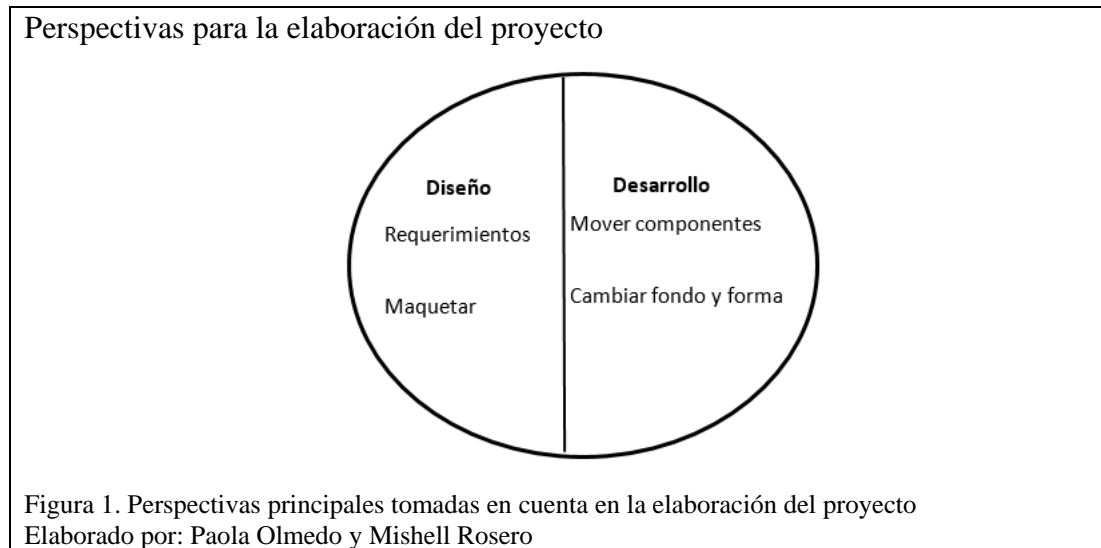
Se planteó el uso de la metodología ágil XP, que propone la organización de tareas en base en fases, teniendo como fin que todo proyecto se los divida en partes pequeñas y se realicen entregas periódicas. En el transcurso del proceso se optó por otras prácticas no utilizadas en el desarrollo sino en el diseño, manteniendo la esencia iterativa propuesta por el XP.

Adaptaciones a la metodología XP.

Fue necesario hacer adaptaciones a la metodología propuesta, por la misma naturaleza del proyecto que es cambiante, estas adaptaciones son las siguientes:

XP propone en una de sus fases tener “historias de usuario”, esta fue omitida debido a que el presente proyecto no contó con un cliente final con el cual se levante requerimientos para las historias de usuario.

Otra de las adaptaciones realizadas es a la fase de “diseño”, ya que XP propone un diseño funcional, sin embargo, en este proyecto se buscó la fusión con el proceso de diseño basado en lo visual, dado que el diseño es el complemento de la funcionalidad que se utilizó, a continuación, se observan las perspectivas que se encontraron en el desarrollo del proyecto.



La figura No. 1, muestra la separación del proyecto, teniendo una perspectiva de diseño, que es la adaptación realizada a la metodología y la otra perspectiva que es el desarrollo, que estuvo enfocado en la parte funcional de acuerdo a los objetivos.

Desde la perspectiva del diseño se tienen las siguientes prácticas:

Requerimientos: permitió una planificación con los requerimientos planteados inicialmente.

Maquetar: esbozar el diseño de cada uno de los prototipos.

Prototipo: una presentación de cada una de las iteraciones.

El proceso del diseño se realizó en todas las iteraciones, se hizo una planificación que se llevó a cabo una sola vez, entendiendo que los requerimientos serían generales para

todas ellas. El desarrollo se abordó considerando las tecnologías a ser utilizadas tal como se observó en la figura No. 1 enfocado en la funcionalidad. El proyecto se dividió en iteraciones documentando el resultado final de cada una de ellas.

La siguiente tabla muestra un resumen de las iteraciones que se llevaron a cabo en el desarrollo del proyecto.

Tabla 1. Resumen de iteraciones

Iteración	Resumen	Resultados
1: Evaluación de funcionalidades de AngularJs	Se logró comprender y evaluar las funcionalidades que AngularJs ofrece para cumplir con los objetivos planteados	Se demostró que las funcionalidades de cambio de color, forma, posición y fondo son posibles
2: Primer prototipo funcional	Este prototipo cumple con todos los objetivos de personalización propuestos, sin embargo, todavía no tiene una conexión con Moodle	El sitio web permite el cambio de fondo, color, forma y posición, en una base local.
3: Prototipo Producto final	El sitio web cumple con todos los objetivos, para esta iteración Moodle se encuentra instalado en un servidor, y tiene la tabla mdl_personalizacion donde los datos se guardan	El sitio web interactúa con Moodle y se puede visualizar las materias y notas del usuario, además de contar con las opciones de personalización de cambio de fondo, color, forma y posición,
4. Pruebas	Es una iteración especial en la que no arroja un producto, pero permitieron evaluar la eficiencia de consumir los recursos con las tecnologías de lado del cliente	Se realizó un simulador que representa el envío de los datos al servidor con tecnología cliente/servidor, para de este modo hacer una comparación y demostrar lo eficiente que resulta consumir recursos del lado del cliente.

Nota: Tabla que muestra las iteraciones que se realizaron en el proyecto

Capítulo 1

Marco teórico

1.1. Personalización

De acuerdo con el diccionario de la Real Academia Española, la palabra personalizar significa: “Dar carácter personal a algo”, es decir, el hombre por naturaleza personaliza todo a su alrededor.

A lo largo del tiempo la humanidad ha buscado la manera de personalizar los objetos que se encontraban a su alrededor, esta actividad anteriormente se dificultaba dado que las personas no contaban con los recursos necesarios para cumplir su objetivo y a medida que el tiempo ha avanzado la personalización ha evolucionado volviéndose más eficaz.

En cierto modo la personalización es nuestra identidad, lo que refleja nuestros gustos, nuestras tendencias y nuestro propio yo, así como lo que queremos proyectar. La personalización por tanto ofrece un toque de identidad, pero a la misma vez de exclusividad. (Psicología digital, 2012)

El ser humano personaliza, con la finalidad de satisfacer sus gustos, siendo la personalización la muestra de diferencia ante las demás personas (Castelló, 1967). De acuerdo con lo expuesto por el autor la personalización ayuda al ser humano en su distinción.

Por ejemplo, si una persona desea una mesa de oficina, la puede personalizar a su gusto, eligiendo el tipo de mesa, color, tamaño, entre otras. (Porcar Seder & Such

Pérez, 2002), este ejemplo es claro al momento de realizar una personalización de un bien. Otro ejemplo se da también con la personalización de calzado.

En la actualidad los jugadores de baloncesto pueden personalizar sus zapatillas, dándoles diversos cambios en cuanto a estampados, colores, añadiendo su nombre, apodo, etc. (Martínez J. , 2013), es común ver que la humanidad puede personalizar desde sus prendas de vestir hasta los servicios que puede ir adquiriendo.

1.2. Personalización de aplicaciones web

La tecnología en la actualidad permite a los usuarios personalizar contenidos como, por ejemplo: para ver una película o serie ya no es necesario conectarse a una hora específica, con las transmisiones bajo demanda en el ámbito de las tecnologías llamado “on demand” de Netflix, se puede seleccionar diversas opciones de visualización, de acuerdo con los gustos del usuario.

(Alvarez & An, 2001), afirman que:

La evolución de los servicios y contenidos web ha seguido dos líneas muy definidas: enriquecimiento (en cuanto a volumen, complejidad y prestación) y personalización (como medio para mejorar la experiencia de la navegación del usuario), eliminando redundancias, ahorrando el acceso a información irrelevante o falta de interés, y ajustando los servicios a sus verdaderas necesidades (p.129)

Según lo citado en el párrafo anterior con estos dos tipos de líneas, tanto de enriquecimiento, como de personalización se podría mejorar en alguna medida la experiencia del usuario en la navegación de un sitio web.

La Internet ha sido un acontecimiento que sin duda ha permitido que hoy en día se tengan beneficios tales como el marketing. Juan José Castaño y Susana Jurado, mencionan que: “el marketing digital nace con el auge de las nuevas tecnologías y la nueva forma de usar y entender internet, que consiste en utilizar las técnicas del marketing tradicional en entornos digitales” (Castaño & Jurado, 2016).

Cabe señalar que el marketing digital está inclinado a una gran cantidad de individuos, teniendo como fin un trato único al usuario, tal y como era en épocas pasadas, el trato con el cliente debe ser amable. En el entorno digital debe pasar lo mismo, así como un cliente va personalmente a comprar algo a una tienda, debe recibir un buen trato por parte del vendedor, ahora al tener al alcance aplicaciones web que comercializan servicios y productos, el cliente debe sentirse cómodo al comprar y percibir una buena interacción con la aplicación que está visitando, sintiendo a su alcance la facilidad de uso.

Anteriormente se citó al marketing digital como un avance tecnológico, sin embargo, otro evento importante en aplicaciones web son las aulas virtuales, permitiendo al usuario tomar una carrera o curso de forma online, o enviar una tarea vía web, en la cual los profesores evidencien el potencial que tienen las nuevas tecnologías informáticas y de comunicaciones para transformar sus prácticas educativas. (Baggetun, 2006)

Hoy en día es fundamental que las aplicaciones web puedan ser personalizables, es decir se busca rediseñar o cambiar componentes en una página web, basado en los gustos de cada usuario y lo que el mismo busca, tal es el caso de la personalización de un dispositivo móvil, teniendo la posibilidad de agregar o quitar componentes, cambiar

el fondo de pantalla, esto según el gusto del usuario. Así mismo, es indispensable que un sitio web sea comprensible, intuitivo y atractivo para el que lo usa.

1.3. Usabilidad

La palabra usabilidad enfocada a un sitio web se refiere al atractivo del sitio web, de tal forma que permita a los usuarios interactuar cómodamente y comprender sus contenidos.

(Montero, 2002) Afirma que:

La mejor forma de crear un sitio web usable es realizando un diseño centrado en el usuario, diseñando para y por el usuario, en contraposición a lo que podría ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad (p.35).

En la cita anterior el autor manifiesta, que con la finalidad de hacer a un sitio web usable hay que pensar como el usuario, quien va a hacer uso del mismo, el diseño de dicho sitio es importante en la navegación del usuario.

Para probar que la usabilidad en un sitio está funcionando es necesario realizarle evaluaciones, debido a que, si el diseño no está acorde a lo que se pide, será una pérdida de recursos. (Montero, 2002), a cada aplicación web es necesario realizarle un seguimiento, e ir midiendo el nivel de apreciación del usuario.

Con la finalidad de captar más usuarios es indispensable que los componentes se encuentren bien proporcionados y tengan una jerarquía visual, así como el texto y

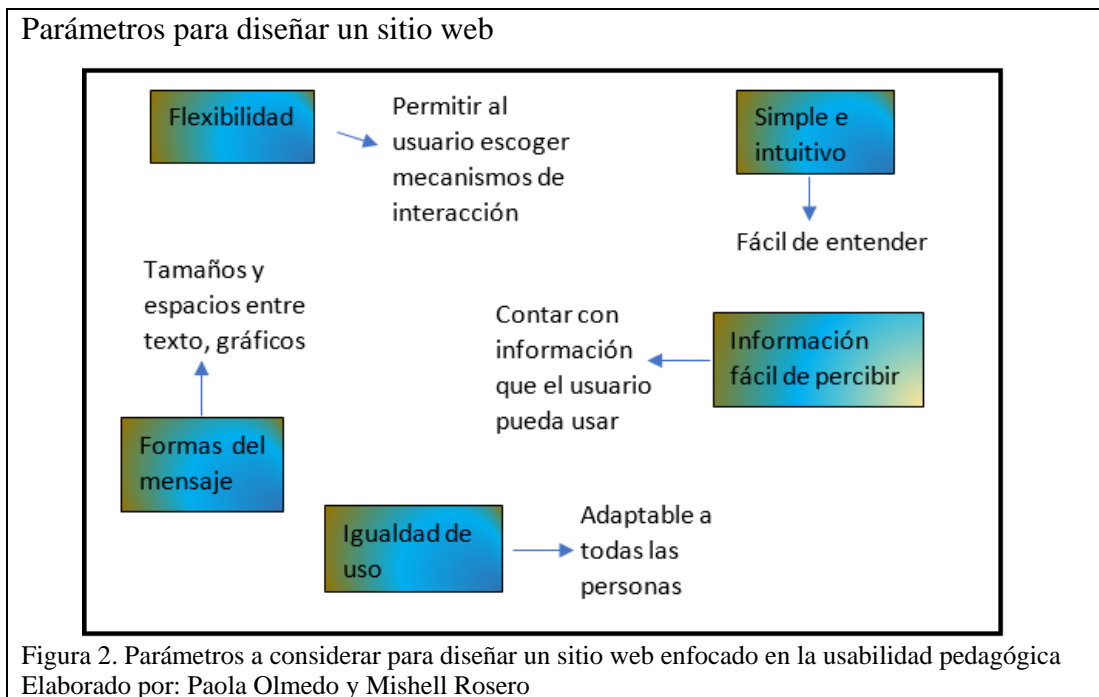
gráfico deben ser comprensibles, estableciendo una correcta navegación, además, de tener la capacidad de realizar configuraciones y personalización (Domínguez, 2012) .

1.4. Usabilidad Pedagógica

Muchas instituciones educativas hoy en día han optado por implementar aulas virtuales, con facilidad de uso y aprendizaje, la importancia de incorporarlas está dado por lograr la eficiencia, asegurando la rapidez en cuanto al aprendizaje y exponiendo la calidad al mostrar los contenidos, así los usuarios que acceden sienten que pueden tener el control sobre su sitio e ingresan con mayor seguridad y confiando en los recursos que disponen a su alcance, su grado de satisfacción será mayor (Gebera, 2012).

1.4.1. Parámetros a considerarse para el diseño de un sitio web enfocado en la usabilidad pedagógica.

Para realizar el diseño de un aula virtual, no existe un formato específico, sin embargo, se puede tomar en cuenta ciertos parámetros, tal y como indica la figura No. 2 “Parámetros para diseñar un sitio web”.



1.5. Diseño Web

De acuerdo con la autora Cristina Paredes, en su libro “Los mejores diseñadores gráficos del mundo”, el diseño es “una disciplina que propone soluciones a necesidades concretas de comunicación, lo que supone solventar problemas, expresar ideas y, en definitiva, realizar un gran ejercicio intelectual” (Benítez, 2012). El diseño es considerado en la resolución de problemas, dado desde la imaginación del ser humano con la capacidad para resolverlos, llegando a ser este un arte.

Con un concepto de diseño en un contexto general, el diseño web radica en planificar, crear e implementar sitios web, para lo cual se deben considerar varios parámetros, tales como, usabilidad, accesibilidad y la integración de contenidos multimedia, entre ellos: imagen, texto, video y audio (Carol, 2016).

De acuerdo a la historia de Internet, se han realizado varias pruebas e investigaciones, las cuales dieron paso a un hito relevante, que es el lanzamiento de la red mundial conocida como World Wide Web (Tovar-Cardozo, Pimienta-Acero, & Ramírez-Saavedra, 2015). Posteriormente Tim Berners Lee enlazó Internet con el protocolo HTTP, llegando a crear la web y así ampliando las formas de comunicación y transmisión de información.

Cabe señalar que HTTP es el protocolo de comunicación, que proporciona la transferencia de datos entre ordenadores para conectar páginas web y archivos multimedia, permitiendo hacer peticiones y recibir respuestas de los servidores web.

De acuerdo a la historia de la web, se tiene una tabla que muestra cómo fue su evolución hasta la web 3.0 que en la actualidad se está usando.

Tabla 2. Evolución de la web

Web	Características	Ventajas	Desventajas	Tipos de servicios
Web 1.0	Empieza con navegadores solo de tipo texto, presentando contenido de páginas estáticas	El creador de la página web es el que puede agregar contenidos	Decadencia de base de datos, por lo que no existía registro de usuarios Gasto de recursos, como el tiempo al tratar de realizar algún cambio	Yahoo! Altavista

Web 2.0	Nace en el año 2004, con un contenido multimedia, surgen las redes sociales que permiten compartir contenidos	Usan tecnologías con base de datos El usuario puede crear contenidos	Saturación en las páginas web Poca seguridad informática	Facebook MySpace Foros
Web 3.0	Web más inteligente, incorpora ya una navegación geoespacial, permitiendo al usuario ubicarse u obtener la dirección de algún lugar en específico, pudiendo obtener información de coordenadas.	Fácil de realizar modificaciones al diseño Mejor manejo y organización de contenido	Costoso y tedioso la reestructuración de los contenidos	Social Commerce (usuario comparte sus gustos y compras con otros)

Nota: Tabla que presenta la evolución de la historia de la web

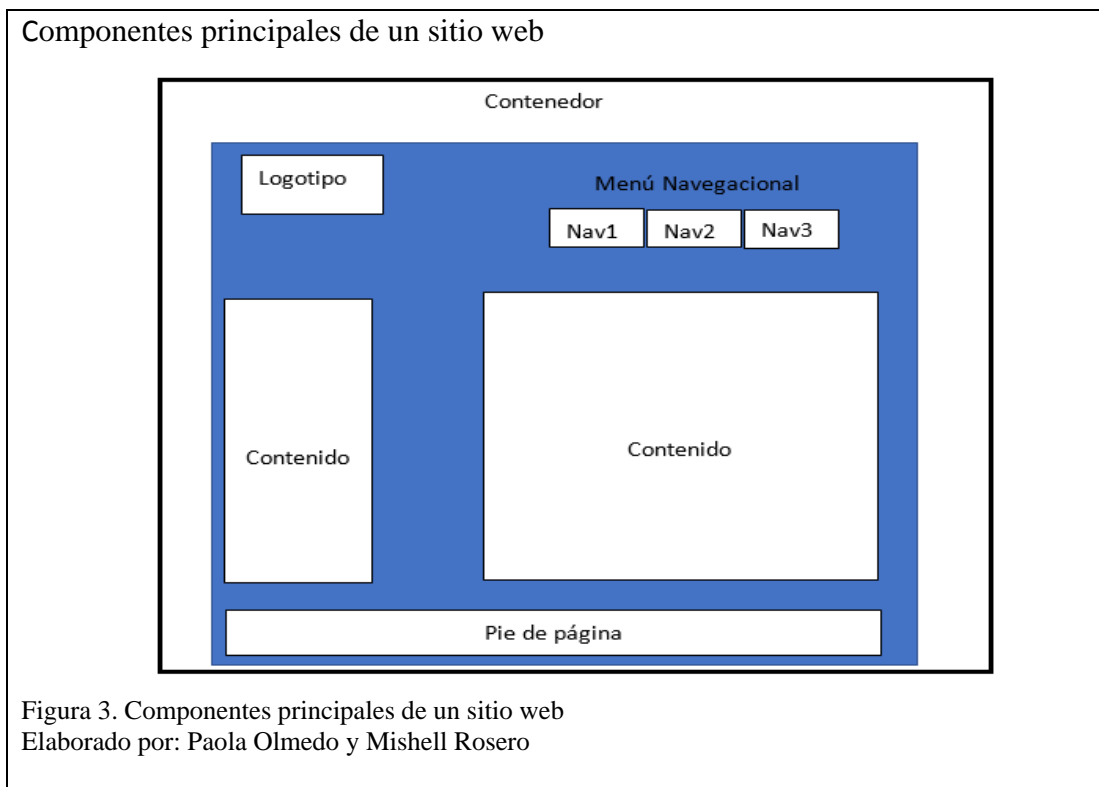
Como se observó en la tabla No. 2, la web avanza constantemente, sin embargo, existe una gran diferencia entre cada una, haciendo hincapié en que la web 3.0 da paso a la realización de aplicaciones web, llegando a ofrecer una forma evolucionada en ellas, así como las búsquedas inteligentes, permitiendo al usuario que se le presente una forma de búsqueda más personalizada, de acuerdo a lo buscado antes en el navegador web. Otra de las evoluciones que posee este avance es que, con la capacidad de crear aplicaciones web dinámicas, permiten mejorar la experiencia del usuario (ConexiónEsan, 2013).

Las páginas web que llegan a ser personalizables, engloban varias funciones, como son gestores de descarga, creación de banner, gestión de comercio electrónico, entre

otras. Para realizar la personalización se debe basar en las últimas tecnologías, que ayudan en la usabilidad y a mejorar la experiencia del usuario, así como HTML5, JavaScript, CSS, PHP, MySQL, etc.

1.5.1. Componentes principales de un sitio web.

Los sitios web se estructuran y se componen de elementos, tales como contenedor, contenido, etc., la figura No. 3 muestra una estructura de los componentes.



1.5.1.1. Contenedor.

Se define en el cuerpo o en algún contenedor para etiquetas (div) de la estructura, dentro del contenedor se encuentran elementos como: contenidos, imágenes, módulos, etc.

El contenedor puede ser fijado por un ancho estándar, que será igual para todos los dispositivos que se adaptarán al ancho de la ventana (Suez, 2013), hay que considerar el borde del sitio web, ya que los componentes no deberán moverse de ninguno de los lados.

1.5.1.2. Logotipo.

Es la imagen de la empresa o institución que identifica la marca en todo el sitio, el mismo debe ser ubicado en la parte superior del sitio ayudando a reconocer a la empresa (Suez, 2013), es indispensable que un sitio web tenga un logotipo, ya que así podrá diferenciarse de los demás.

1.5.1.3. Menú navegacional.

Es el menú en el cual se ubican elementos que permiten la navegación dentro del sitio, localizándose de forma horizontal o vertical, los menús verticales se usan como complemento o sub menú de los horizontales (Suez, 2013), estos menús también se ubican en la cabecera de la página, permitiendo redirigir a otras páginas.

1.5.1.4. Contenido.

El contenido es la parte fundamental de un sitio web, siendo el responsable de mantener al usuario interesado en encontrarse en dicha web sin cambiarla por otra, por lo que debe enfocarse y centralizarse al momento de su ingreso (Suez, 2013), se pueden ubicar desde texto, gráficos, o videos a reproducirse, dependiendo el giro del negocio.

1.5.1.5. Pie de página.

En inglés llamado footer, en el contexto de las aplicaciones web es la parte que está ubicado al final del sitio, usualmente se lo emplea para información de derechos de autor (Copyright) o menús secundarios, a pesar que con la ayuda de las redes sociales

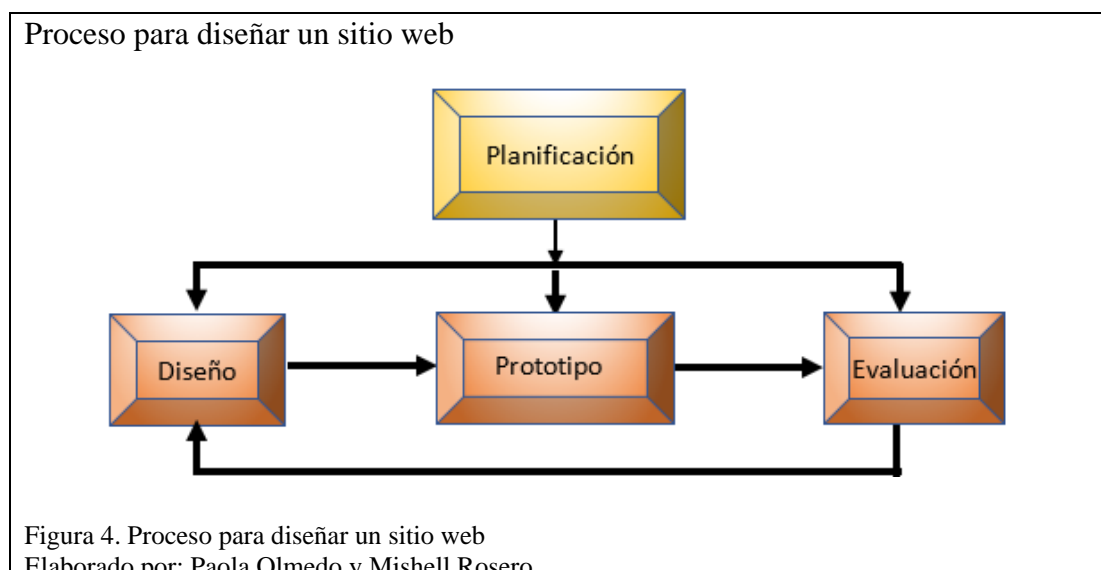
se las usa para incluir información de noticias o direcciones (Suez, 2013), algunos sitios web no suelen hacer uso del pie de página, razón por la cual no es obligatorio que sea llenado.

1.5.1.6. Espacio negativo.

Es conocido como espacio blanco, refiriéndose a todo el espacio que no disponga de información, sin embargo, cumple un rol importante en la web, ayuda a dar balance y equilibra el diseño, en el caso que el cliente desconozca de este sitio, se puede llenar los espacios, considerando no saturarlo con demasiado contenido, dado que se convertiría en un sitio innavegable (Suez, 2013), tal y como lo expresa el autor, el usuario no debe ver muy cargado el sitio, porque puede llegar a perder su interés.

1.5.2. Diseño web enfocado en la usabilidad del usuario.

Se refiere al proceso que el diseñador realiza con los requerimientos, para posteriormente obtener un producto final, tomando en cuenta que la experiencia del usuario mejore (Hassan, Martín, & Iazza, 2004), el propósito de un sitio web debe ser que la experiencia del usuario vaya mejorando continuamente, por eso es necesario realizar evaluaciones constantes al sitio.



En la figura No. 4 se evidenció que el proceso de diseño, prototipo y evaluación se repiten, puesto que, cada vez que realice un diseño este debe ser evaluado constantemente y así corregir errores en cuanto a usabilidad desde que se empiece a desarrollar (Hassan, Martín, & Iazza, 2004), es recomendable que estas etapas del proceso se repitan, ya que así se tendrá como producto final un sitio web más agradable ante la vista del usuario.

1.5.2.1. Planificación.

Permite identificar, analizar y ordenar los requerimientos para el sitio web, es importante que los contenidos y el diseño cumplan con los objetivos planteados desde el inicio (Hassan, Martín, & Iazza, 2004), con una debida planificación se puede ir mejorando el proceso de diseño.

1.5.2.2. Diseño.

Permite realizar un borrador previo de acuerdo a los requerimientos planteados y a la planificación, logrando conocer si es necesario volver a rediseñar (Hassan, Martín, & Iazza, 2004), entre los tipos de diseño se tienen los siguientes:

1.5.2.2.1. Diseño conceptual.

Busca definir un esquema para la organización, navegación y funcionamiento del sitio, sin mostrar la apariencia final, concentrándose en la Arquitectura de Información. Cabe señalar que actualmente se cuenta con la estructura, la cual trata de las relaciones y conexiones entre las páginas, además cuenta con la navegación, la misma que se refiere a las probabilidades y formas en donde cada página muestra sus opciones de

desplazarse a otras páginas (Gauchat, 2012), este tipo de diseño permite definir conceptualmente la ubicación visual de los componentes del sitio web.

Para definir la estructura se tienen dos enfoques diferentes, tales como: aproximación ascendente y descendente. Con el enfoque ascendente se pueden definir bloques de estructuración e información, mientras que el enfoque descendente lo que hace es dividir los contenidos en diferentes páginas, además, definen los enlaces de conexión entre ellas. Cuando se haya definido la estructura es indispensable su documentación, que puede ser con grafos, esquemas, etc., teniendo como objetivo su comprensión (Hassan, Martín, & Iazza, 2004), la estructura del diseño va a depender de la propuesta del sitio web.

1.5.2.2.2. Diseño de contenidos.

Los contenidos deben tener una coherencia informativa, organizativa y comunicacional, manteniendo una redacción acorde al tipo de usuario al que está enfocada la página. No es necesario presentar tanto contenido, puesto que al usuario no le agrada leer la pantalla, se deben transmitir ideas concisas (Hassan, Martín, & Iazza, 2004), este tipo de diseño se refiere más al contenido que se presentará al usuario, ya que puede contener objetos multimedia, así como videos, audios, etc.

1.5.2.3. Prototipo.

Desde las primeras etapas del diseño es importante que se realice la evaluación de la usabilidad del sitio web y esto se lo hace a través de prototipos. El prototipo es realizar un modelo de la interfaz, sin embargo, este no será un producto final, pero ayudará hacer un boceto visual del sitio web (Hassan, Martín, & Iazza, 2004), es

indispensable realizar un prototipado, ya que permitirá presentar un sitio más acorde a lo que se planificó inicialmente.

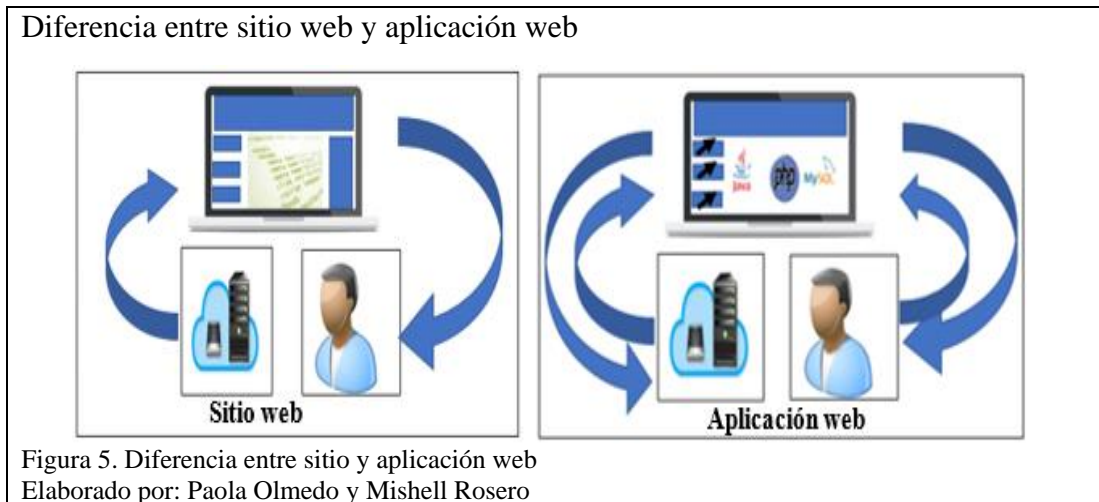
1.5.2.4. Evaluación.

La evaluación se la debe realizar en cualquier proceso del diseño web, dado a que se valorará principalmente la usabilidad del sitio, para de esta forma lograr ver el avance, si está acorde a los requerimientos o si es necesario volver a rediseñar (Hassan, Martín, & Iazza, 2004), se puede evaluar un sitio web preguntándole al usuario si mejoró su experiencia al usarlo.

1.6. Diferencia entre aplicación web y sitio web

Una aplicación web está enfocada en la funcionalidad, es decir, las tareas que el usuario va a realizar, para lo cual se utilizan aplicaciones para su puesta en marcha, el sitio web es el encargado de presentar la información enfocándose en la parte visual hacia el usuario (Pizarro, 2011).

La funcionalidad dependerá del tipo de lenguaje de programación usado, puesto que algunos son de paga y otros de código abierto, una aplicación web puede pertenecer a un sitio web, pero no lo contrario, el nivel de complejidad de una aplicación web es más alto que el de un sitio web (Pizarro, 2011), al referirse al nivel de complejidad señala que es la funcionalidad que la aplicación tenga.



La figura No. 5 expone una clara diferencia entre sitio web y aplicación web, correspondiendo la imagen de la izquierda la que señala un sitio web con una presentación de contenido al usuario, siendo normalmente unidireccional la respuesta desde el servidor, mientras que la figura de la derecha llamada aplicación web señala una comunicación bidireccional, presentando al usuario la posibilidad de realizar consultas desde una base de datos, hacer cálculos, entre otras.

1.7. Maqueta web

Es la representación estática de un diseño visual, en el diseño web se usan para que el equipo tenga una idea clara de cómo se mostrará el sitio web, esta idea es ver cómo será la página web para después continuar al siguiente paso (aulainformativa(s.f), 2016), la maquetación se la utiliza para realizar un bosquejo y presentar al cliente y así llegar a un acuerdo si cumple con sus requerimientos.

1.8. Aplicaciones del lado del servidor

Un servidor se encarga de almacenar información y mostrar mediante la web, para que más adelante los usuarios lo vean, dentro del modelo de cliente/servidor, “un *cliente* es un programa que utiliza los servicios que proporcionan otros programas. Los

programas que proporcionan los servicios se llaman *servidores*.” (IBM Knowledge Center, 2014), por lo que el cliente hace una petición de algún servicio y el servidor llevará a cabo dicho servicio, entre las funciones que realiza el servidor se encuentran gestionar las peticiones del cliente con información o datos de estado.

Físicamente el servidor es una computadora que almacena y envía datos mediante la red, si un usuario ingresa a un navegador inicia una comunicación con este, al enviar y recibir datos, que básicamente es lo que se muestra en la pantalla del cliente. El servidor se puede llegar a saturar con tanta información almacenada, sin embargo, al ampliar componentes de hardware, todo irá funcionando bien.

Según el físico y matemático Tim Berners-Lee creó el primer servidor web debido a que notó que una sola computadora no bastaría para almacenar tanta información y nace la idea de centralizarla (Diez, 2016). Para realizar un sitio web es necesario conocer de lenguajes de programación, que serán los que permitan que una página pueda ser dinámica.

Los lenguajes de programación se iniciaron en los años 50, partiendo de la necesidad de resolver un problema. “El lenguaje de programación debe ser universal. Es decir, cualquier problema debe tener una solución que puede ser programada en el lenguaje y dicha solución ser implementada en cualquier computador” (Ruíz Lizama, 2014). A medida que transcurre el tiempo los lenguajes de programación se siguen actualizando, permitiendo crear páginas web estáticas y dinámicas.

1.8.1. Lenguajes de Programación principales del lado del servidor.

En la tabla No. 3 se observan los lenguajes del lado del servidor que se usan con más frecuencia.

Tabla 3. Lenguajes de programación más usados en el lado del servidor

Lenguaje	Características	Ventajas	Desventajas
PHP	Permite crear páginas web dinámicas, que trabajan dentro de páginas HTML	Multiplataforma Código abierto Aplica técnica de programación orientada a objetos Interactúa con diversos motores de bases de datos	Poseer un servidor web que soporte PHP La mayor parte de las páginas web pueden no ser accesibles en los navegadores
JSP Java Server Pages	Fundamentado en Java, permitiendo crear páginas dinámicas	Open source Código bien estructurado Multiplataforma	No es recomendable usar en proyectos pequeños La lógica de la aplicación se la hace del lado del servidor, corriendo el riesgo de consumir muchos recursos

			Llega a ser difícil, para los que no tengan una preparación en java
--	--	--	---

Nota: Tabla que presenta información de los lenguajes de programación más usados

1.9. Aplicaciones del lado del cliente

Los usuarios antes tenían aplicaciones limitadas, es por eso que se crean aplicaciones en las que el usuario pueda por sí mismo manipularlos:

El método de desarrollo web ha llegado a un punto donde la exigencia en ofrecer aplicaciones con una experiencia de usuario aceptable, que respondan al ritmo evolutivo de la tecnología y necesidades de un usuario en constante movimiento con acceso a la información desde cualquier lugar o dispositivo (Martínez C. , 2013).

Las aplicaciones del lado del cliente, al igual que las del lado del servidor se usan para desarrollar páginas web con contenido dinámico, con la diferencia de, que el servidor ya no ejecuta ni procesa los scripts¹, sino lo hace el cliente, es por eso que se incluyen en documentos HTML, que después llamará a la página principal.

¹Conjunto de órdenes almacenadas en un archivo de texto

1.9.1. Tecnologías.

1.9.1.1 HTML5.

HTML5 (Lenguaje de marcas de hipertexto), predecesor de versiones anteriores de HTML, es una de las tecnologías que se usa más para el desarrollo web, está conformada por: estructura, diseño y funcionalidad, donde HTML será la estructura, a cargo del diseño están los CSS y la funcionalidad se trata de código realizado en JavaScript (Gauchat, 2012).

1.9.1.1.1. Ventajas.

El código abierto contiene etiquetas, permitiendo ordenar y clasificar en diferentes niveles el contenido, además permite realizar páginas menos pesadas, por la simplicidad de su código e incorpora etiqueta de dibujo “Canvas”, (elemento HTML, que se lo puede usar para dibujar gráficos mediante scripts), incrementando los efectos visuales al cliente (Parreño & Arana, 2016).

La tabla No. 4 presenta la evolución y los cambios que ha ido teniendo HTML desde su creación.

Tabla 4. Evolución de HTML

Año	Versión	Característica
1991-1993	HTML	Creado para que pueda ser interpretado por el navegador, se crean los primeros bocetos
1995	HTML2.0	Incorpora etiquetas para imágenes, formularios y tablas, primera versión en convertirse en estándar
1997	HTML 3.2	Pasando a la W3C, se crea una versión mejorada
1998-2000	HTML4	Considerado un paso importante, ya incorpora nuevos métodos, así mismo se crea una versión mejorada llamada XHTML, pero no se usó tanto, debido a que es más estricto

2009	HTML5	Es una mejora de XHTML, dando paso a nuevas actualizaciones, ya incorporan etiquetas, las cuales ninguna de las anteriores versiones integraba
------	-------	--

Nota: Tabla que presenta la evolución a lo largo del tiempo de HTML

HTML5 ahora incorpora actualizaciones que permiten realizar sitios web más atractivos para el usuario, sin embargo, por sí sólo no hace ninguna acción, pero con la ayuda de tecnologías como CSS, permitirá la creación de una página web.

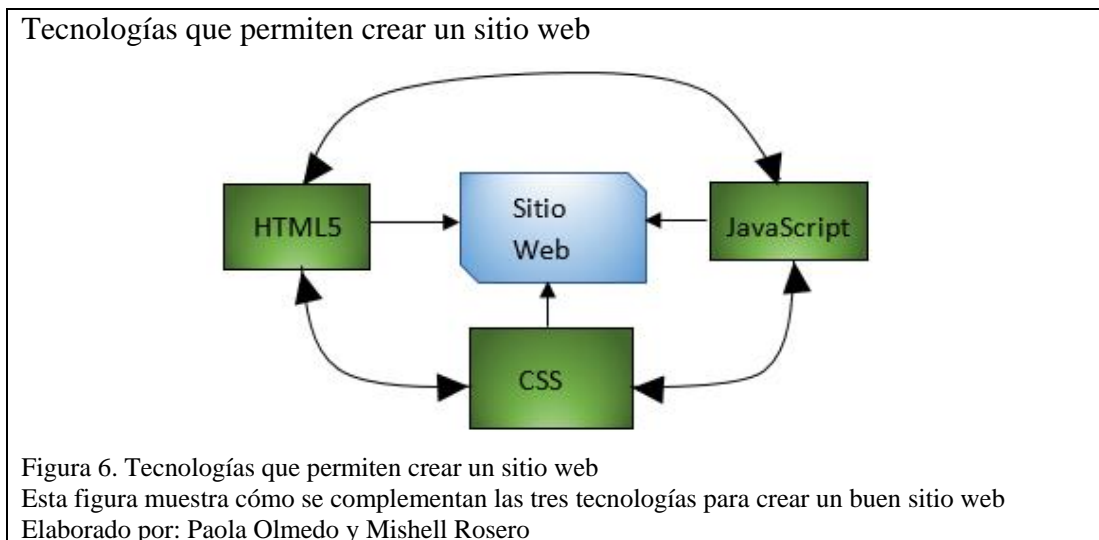
1.9.1.2. CSS.

Las hojas de estilo en cascada son documentos que se los usa para darle estilo a documentos XML y HTML, ayudando a mejorar el diseño de las páginas web, ya sea en su forma, fondo, etc.

Ayuda a los desarrolladores a tener el control del estilo y formato de diversas páginas web y así, si realizara algún cambio en algún elemento del código se cambiarán en todas las páginas que contenga el mismo elemento (Valbuena, 2014).

1.9.1.3. JavaScript.

Es el lenguaje de programación que se usa especialmente en el desarrollo de páginas web del lado del cliente, el cual permite la ejecución de acciones, soportado en la mayor parte de sus navegadores, soporta varios estilos de programación, está orientado a objetos, también se lo usa de forma local (Valbuena, 2014).



1.9.1.3.1. Drag & Drop.

Entre las funciones que incorpora JavaScript existe el drag & drop, en español “arrastrar y soltar”, esta función lo que permite es que los componentes de un sitio web se los pueda llevar a otro lado del que se encuentran actualmente, el usuario debe hacer click sobre los componentes, manteniendo presionado el botón del mouse, lo arrastrará a otra ubicación donde quedará colocado (ethertank, 2015), esta función trabaja con JavaScript, pero también con HTML5.

Puede ser incorporado en cualquier contenedor, como divs, tablas, etc. Esta función permite que el usuario pueda interactuar de una manera muy accesible, tanto en dispositivos móviles como de escritorio (Cruz, 2016).

1.9.2. Otras tecnologías del lado del cliente.

A pesar que se cuenta con HTML5 para realizar un diseño web llamativo, también existen otras herramientas que pueden crear páginas dinámicas, entre ellas está Adobe flash, pero no es recomendable usarla, puesto que para reproducir en esta herramienta

es necesario que el usuario la instale en su navegador, mientras que HTML5 ya incorpora su código al llamarlo desde el navegador (Casale, 2015).

De lo expuesto anteriormente Adobe flash es una tecnología que es frecuentemente usada del lado del cliente, sin embargo, como ha ocurrido con algunas de las tecnologías, ha ido perdiendo su auge.

1.10. Aplicaciones del lado del servidor vs Aplicaciones del lado del cliente

Las aplicaciones del lado del servidor se ejecutan directamente en un servidor reduciendo errores o problemas en cuanto a la compatibilidad, usando un lenguaje de programación, el servidor espera solicitudes de conexión mediante protocolo HTTP por parte del cliente (Techlandia., 2014), por lo que el tiempo de respuesta desde el servidor puede llegar a ser lento, debido a que este procesará la recepción de petición en petición.

Las aplicaciones web por parte del servidor se conforman de páginas sin movimiento, donde el cliente puede visualizar (Luján Mora, 2002). Entre las aplicaciones que se pueden realizar del lado del servidor, están las aplicaciones bancarias, debido a que el servidor permite que no se vulnere la seguridad como claves, datos, etc.

Por otro lado, las aplicaciones del lado del cliente se manejan mediante un código HTML, siendo su lenguaje de código, que se ejecuta desde el navegador (Luján Mora, 2002). Toda la programación se la realiza en el ordenador del usuario permitiendo a la página la facilidad de albergarse en cualquier espacio. El usuario tiene la posibilidad de hacer cambios, puede organizar los componentes a su gusto, siendo de esta forma más llamativa para el mismo usuario, siempre y cuando el programador sepa cómo llegar a él con conocimientos de usabilidad (Diaz, 2014), es por eso que este proyecto

pretende hacer que el usuario tenga a su mando la posibilidad de mover los componentes a su gusto.

En cuanto a las tendencias que ofrecen las aplicaciones del lado del cliente están la forma de presentación de la página web hacia el cliente, basándose mucho en el diseño o CSS, tal como son los menús hamburguesa, este tipo de presentación le da un toque más usable al usuario, se seguirá extendiendo en cuanto a su forma de presentación, adaptándose a otras formas de visualización, puesto que su simplicidad y su claridad convencen a los usuarios, aportando a que se usen solo los elementos mínimos y básicos del diseño, su implementación es muy eficaz (Vecino, 2017).

1.11. Ejemplos de aplicaciones realizadas del lado del servidor

A continuación, se colocan algunos ejemplos de aplicaciones que se han creado del lado del servidor, cabe mencionar que no ha existido algún intento de personalización de componentes, pero sí se han realizado aplicaciones personalizables, así como indicar sugerencias, según los gustos del usuario.

Páginas como Amazon, que construyen resultados de búsqueda, haciendo sugerencias de artículos que se basan en preferencias del consumidor y en compras anteriores, facilitando su comercio. Redes sociales tales como, Facebook, Twitter, Instagram permiten destacar, compartir y tener el control al contenido que sea de interés para el usuario. Estas aplicaciones consumen información almacenada transfiriéndola desde una base de datos (MDN web docs, 2017).

1.12. Frameworks

El framework facilita el desarrollo de aplicaciones, está compuesto por varios componentes personalizables, se lo puede configurar, añadiendo varios elementos para la realización de una aplicación y así agilitando el desarrollo de la misma. Entre sus principales objetivos están promover las buenas prácticas de desarrollo, existen frameworks para lenguajes como JavaScript y HTML5 (Valbuena, 2014). Entre los frameworks más usados del lado del cliente están:

1.12.1. ReactJs.

Fue pensada para la creación de interfaces de usuario, su creador fue Facebook, es una librería de código abierto apoyado en JavaScript, permite unir la vista de los componentes creados por el desarrollador, tal que si se realiza algún cambio en los datos también se modificará la vista (Sanvicente, 2017), tal y como lo dice el autor no es un framework, sino una biblioteca que permite únicamente crear una interface de usuario, sin embargo, se lo puede llegar a combinar con diferentes arquitecturas ya sean de cliente o de servidor.

1.12.2. AngularJs.

Fue creado por Google y es de código abierto, utiliza el patrón de diseño de software MVC (Modelo Vista Controlador), dando la ventaja a que en proyectos con arquitectura cliente/servidor trabajen juntos, sin que llegue a afectar el uno del otro (Parreño & Arana, 2016), este framework fusionado con HTML5 permite la creación de aplicaciones robustas y es compatible con la mayoría de navegadores, además que funciona bien con otras bibliotecas.

AngularJs usa atributos que permiten apoyar a HTML a crear páginas dinámicas, permitiendo al usuario interactuar más en el sitio en el que se encuentre, este framework pasa a un objeto sus dependencias por medio de un parámetro, cuando se crea un controlador, modelo o servicio, AngularJs solo debe cargar sus dependencias para levantar la aplicación web (Alejos, 2014). Se crean objetos que permiten manipular los datos entre el modelo, la vista y el controlador.

Existen otros frameworks como BackboneJs que realizan la misma función de AngularJs, sin embargo, por que usar AngularJs y no los demás, según Alberto Basalo, experto en AngularJs y otras tecnologías basadas en JavaScript, como NodeJs y MongoDB, director de Ágora Binaria, indica: “con AngularJS requieres escribir menos código que con otros frameworks”, comparando con BackboneJs existen mejoras, siendo AngularJs una forma de no usar tanto código, además se encuentra más documentación y componentes (Basalo & Alvarez, 2014).

1.12.2.1. NodeJs.

Es el manejador de librerías preferido de AngularJs, usa un modelo de operaciones orientado a eventos, permitiendo hacerlo más eficiente y liviano, es por eso que en el presente proyecto se lo utiliza, para que conjuntamente trabajen con AngularJs y así se puedan descargar las dependencias, permitiendo una agilidad en el desarrollo. Es el ambiente de librerías de código abierto más grande del mundo (Nodejs, 2018), es por eso que en este proyecto se lo utiliza para que conjuntamente trabajen con AngularJs y así se puedan descargar las dependencias, permitiendo una agilidad en el desarrollo. Este entorno permite trabajar tanto del lado del servidor como del lado del cliente.

1.12.2.1.1. NPM.

Gestor de paquetes (Node Package Manager), administra paquetes para JavaScript y trabaja conjuntamente con NodeJs, permitiendo administrar los módulos, brindando la posibilidad de agregar dependencias de forma sencilla (NPM, 2018), este gestor es el registro de software más grande del mundo.

1.13. Herramientas

1.13.1. JustInMind.

Es una herramienta de código abierto que se usa para crear bocetos, permitiendo diseñar, ya sea para una aplicación móvil o web, siendo una interfaz intuitiva que no involucra código. Se puede crear una plantilla y se la personaliza según el requerimiento del desarrollador, facilitando la presentación al cliente (JustInMind, 2015), para el diseño de cada una de las iteraciones es necesario que sean diseñadas en una herramienta, ya que la misma ayuda a que se logre tener una idea más clara de la presentación del sitio web.

1.13.2. WebStorm.

Es un IDE (Integrated Development Enviroment- Entornos de Desarrollo Integrado), que se lo usa para programar en lenguajes como PHP, Ruby, JavaScript, siendo compatible con algunas tecnologías, que tienen relación con los lenguajes de programación, es de paga, sin embargo, las universidades que tengan convenio con la empresa de desarrollo de esta IDE proveen a los estudiantes poder acceder con una licencia gratuita de un año, realizando el registro con el correo electrónico de la institución (Parreño & Arana, 2016), este IDE ayuda a los desarrolladores a que hagan uso del soporte que ofrece.

1.13.3. Gulp.

Es la herramienta que se usa para desarrollar en JavaScript, permitiendo administrar y reducir varias tareas repetitivas en la construcción de un proyecto (Parreño & Arana, 2016), esta herramienta permite ejecutar un listado de procesos ya predefinidos.

1.13.4. Bitbucket.

Es el repositorio que realiza un control de versiones, permitiendo realizar cambios locales y remotos, trabaja mediante ramas, facilitando hacer modificaciones al proyecto, sin afectar la versión estable que se tenga en la rama principal. Esta herramienta se la usa para una mejor actualización del código fuente (Parreño & Arana, 2016).

1.13.4.1. Git.

El repositorio Git, fue creado por la Empresa Atlassian, que se lo utiliza para alojar en la nube proyectos, permitiendo realizar un control de revisiones, ofreciendo planes gratuitos y de paga (Parreño & Arana, 2016), esta herramienta es parte de Bitbucket que ayudaron en este proyecto a que se tenga un mejor control en cuanto a reutilizar el código.

1.14. LMS

Sistema de Gestión de Aprendizaje - Learning Management Systems, conocido como una herramienta que se usa para la gestión e-learning en la educación a distancia. La gran mayoría de LMS están orientadas en la web, que permiten el aprendizaje en cualquier momento o lugar de acceso a los contenidos subidos (Hurtado, Olmedo, & Rosero, 2017). A más de ser una herramienta, es un Sistema de Gestión de Aprendizaje

que alojada en el servidor permite hacer el control y la administración de cada una de las actividades que posea la organización.

Los LMS proveen módulos administrativos, que permiten opciones como matricular alumnos, configurar cursos, asignar cursos a alumnos, registrar profesores, realizar calificaciones y llevar informes de progreso. Además, permiten facilitar el aprendizaje colaborativo y distribuido partiendo de contenidos ya agregados, usando servicios tales como la comunicación de Internet como foros o chat (Peñalvo, 2005), es indispensable conocer cómo administrar este tipo de herramienta, debido a que, si se trata de una institución educativa y por error se eliminan algunos parámetros, puede llegar a tardar la recuperación de ellos, si no hay una previa capacitación.

1.14.1. Moodle.

Es una plataforma e-learning que se usa para educadores, estudiantes y administradores los cuales hacen uso de esta según su perfil y necesidad, debido a que promueve la enseñanza virtual, tomando en cuenta los principios de seguridad que toda aplicación web debe tener, de esta plataforma se puede sacar provecho la personalización de su ambiente. Moodle es una de las más usadas en el mundo (Parreño & Arana, 2016).

1.14.1.1. MariaDB.

MariaDB es un sistema gestor de base de datos, que dispone de otro tipo de funcionalidades, tales como administrar la recuperación de usuarios y el manejo de información en el sistema si se llegara a corromper, fue creado por los desarrolladores de MySQL, con un mejor rendimiento y con facilidad de uso, se lo considera más abierto y es distribuida como una base de datos, por lo que es compatible con gran

parte de las distribuciones actuales de Linux, permitiendo trabajar con plataformas grandes, así como Moodle (Moodle, 2014).

Es un sistema de gestor de base de datos que dispone de otro tipo de funcionalidades, tales como administrar la recuperación de usuarios y el manejo de información en el sistema si llegara a fallar.

1.14.1.1.1. Autenticación por token.

Moodle ofrece la autenticación de token, lo que permite que el usuario tenga a su disposición el uso de los servicios web, el funcionamiento que se le dé a los tokens va a depender mucho del perfil de usuario que tenga acceso al servicio que esté consumiendo (Parreño & Arana, 2016).

1.15. Metodología ágil

Es una buena práctica que se utiliza para la elaboración y ejecución de un proyecto, permite una mayor satisfacción del cliente. La metodología ágil es un marco de trabajo que permite mejorar la eficiencia en la producción y la calidad de los productos finales, tener la capacidad de respuesta al cambio en los productos y sus definiciones y brindar la mayor satisfacción posible al usuario, a través de la entrega temprana y la retroalimentación continua durante la construcción del proyecto para tener un producto (Vera, 2014).

De acuerdo a lo expuesto anteriormente, las metodologías ágiles fueron creadas con el fin de que un producto pueda ser elaborado con calidad, dando paso para que se le incluya al cliente en la elaboración de este, permitiendo una retroalimentación

frecuente entre él y los desarrolladores, para que al finalizar, el producto llegue a ser lo que el cliente esperaba.

Estas metodologías se basan en las metodologías tradicionales, tomando solo lo más importante de ellas, dejando atrás la documentación exhaustiva (Narváez, 2012), es necesario que toda metodología ágil cuente con una documentación, sin embargo, esta no será extremadamente larga e inentendible y como sus iteraciones son cortas por la previa planificación que se hace, la satisfacción del cliente al final será lo esperado desde el inicio.

Cabe señalar que las metodologías ágiles pueden llegar adaptarse de acuerdo a las necesidades del equipo de trabajo, ya que entre sus principales características es que se llegue a cumplir con lo establecido (Navarro, Fernández, & Morales, 2013). Como se ha mencionado estas metodologías se adaptan a los cambios, depende de las necesidades que el proyecto requiera.

1.15.1. XP (Programación Extrema).

Metodología ágil, creada por Kent Beck en el año de 1996, está enfocada en fomentar el trabajo en equipo, se caracteriza porque se puede mantener una retroalimentación constante con el cliente. Permite conseguir buena documentación del software que se está desarrollando. “Busca dos objetivos claramente: hacer un software bien (con calidad) y de la forma más rápido posible” (Cortizo, Expósito, & Ruiz, 2003), esta metodología pretende presentar un producto de calidad y a su vez permite también contar con una documentación este, que sea fácil de entender.

Esta metodología plantea fases cortas de desarrollo (iteraciones), que consiste en entregables por cada una, para cada iteración existe un ciclo completo de análisis, diseño, desarrollo y pruebas (Joskowicz, 2008), estas iteraciones pueden ir cambiando de acuerdo a la naturaleza del proyecto.

1.15.1.1. Características.

Se consigue tener productos duraderos y más eficientes, integrando todo el trabajo más fácil, se consigue realizar un producto desde el inicio en versiones pequeñas, donde el cliente va observando la evolución del software.

1.15.1.2. Fases de XP.

1.15.1.2.1. Fase I: Planificación.

Los clientes plantean las historias de usuario, siendo de suma importancia en la primera entrega del producto, paralelamente el equipo encargado del desarrollo entiende las tecnologías y herramientas que se usarán, ya probadas las tecnologías se construye un prototipo, esta fase dura pocas semanas o meses y se tienen la cantidad de iteraciones que el proyecto llegue a requerir (Bustamante & Rodríguez, 2014).

1.15.1.2.2. Fase II: Diseño.

El desarrollador debe presentar diseños sencillos y simples, procurando tener menos complicaciones para obtener un diseño fácil de implementar y de comprender que llegará a poseer menos tiempo y esfuerzo en el desarrollo que además debe incluir un glosario de términos (Bustamante & Rodríguez, 2014).

1.15.1.2.3. Fase III: Codificación.

Es necesario codificar las historias de usuario, tomando en cuenta que se encuentre presente, debido a que, de esto dependerán las entregas o iteraciones que tenga el proyecto, para programar se deben considerar estándares de programación, como por ejemplo el interlineado (Bustamante & Rodríguez, 2014).

1.15.1.2.4. Fase IV: Pruebas.

Realizar pruebas a cada una de las iteraciones es primordial, ya que toda metodología ágil debe tener la capacidad de ofrecer pruebas, estas pruebas se las hace junto con el usuario, permitiendo concluir que se realizó de acuerdo a los requerimientos iniciales (Bustamante & Rodríguez, 2014).

1.15.1.3. Adaptaciones a la metodología XP.

XP se caracteriza por su gran capacidad de adaptación ante algún acontecimiento imprevisto, puesto que no se trata de mantener los requerimientos desde que se empieza a elaborar el proyecto, más bien es durante todo el proceso, los requerimientos pueden ir cambiando o evolucionando progresivamente sin contratiempos. Los creadores de esta metodología consideran que es preferible que el desarrollador se adapte en el proceso a los requerimientos que continuamente van apareciendo, a empezar con requerimientos y desarrollar en base a eso un proyecto (OkHusting, 2012), no siempre se van a tener los requerimientos claros, es por eso que mientras se va desarrollando el proyecto se les puede ir incorporando, ya que al ser una metodología adaptativa no tiene algún problema en que se añadan más requerimientos, siempre y cuando no merezca el cambio total del proyecto o empezar a desarrollar de nuevo.

Esta metodología se la usa según como sea necesario, debido a esto, es considerada como la que más resalta de las metodologías ágiles (OkHusting, 2012), es una de las metodologías ágiles más populares, porque se la usa en proyectos pequeños o que no tengan claros sus requerimientos.

Gregorio Robles y Jorge Ferrer dicen en su artículo “Programación eXtrema y Software Libre” que, “el propio desarrollador suele ser a la vez el propio usuario, por lo que no necesita un proceso de formalización de historias de los usuarios” (Robles & Ferrer, 2002). El proyecto que se va a desarrollar no cuenta con historias de usuario, es por eso que necesita de la realización de adaptaciones.

Capítulo 2

Primera Iteración

2.1. Evaluación de funcionalidades de AngularJs

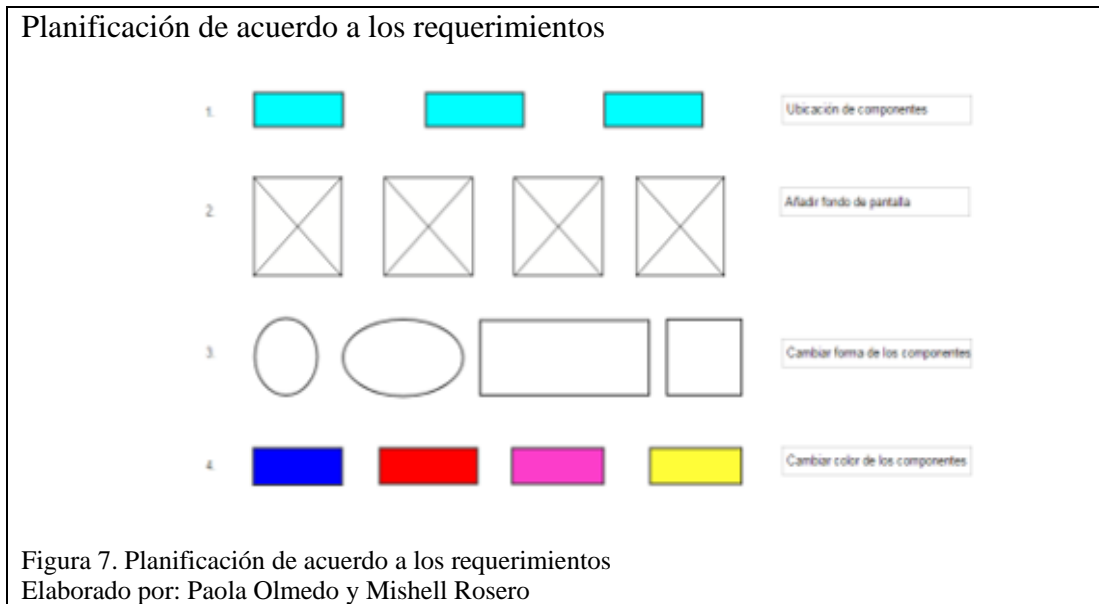
El propósito de este capítulo es describir la iteración, cuyo fin fue realizar la evaluación de componentes, utilizando el código del lado del cliente, con tecnologías tales como JavaScript y HTML5, consumiendo la menor cantidad de recursos del lado del servidor, tomando en cuenta los requerimientos para la personalización que fueron: agregar fondo de pantalla, mover ubicación de componentes y cambiar color y formas de estos, por lo que se empezó desde el diseño web.

De acuerdo a los conceptos mencionados en el acápite del marco teórico, el diseño web es fundamental al momento de la creación de un sitio web, debido a que este se enfoca en lo visual, muestra cómo será presentado al usuario evaluando la usabilidad.

Cabe recalcar que el fin de este proyecto está enfocado principalmente en lo visual, sin embargo, el sitio web si será funcional. Para realizar el diseño de un proyecto, es necesario realizar un proceso, iniciando con un prototipado, la idea se da desde los conceptos de diseño web antes mencionados.

2.1.1. Planificación.

Una vez desarrollada la planificación se estableció la localización de los componentes, estimando la cantidad de prototipos que se iban a desarrollar, sin embargo, inicialmente fue difícil establecer con exactitud el número de iteraciones finales.



En la figura No. 7 se evidenciaron los requerimientos que se tuvieron, la cantidad de componentes se mantuvieron en todas las iteraciones, ya que tal como se mencionó, este proyecto partió de un proyecto de titulación anterior, en el que se presentan tres componentes. En cuanto a los fondos de pantalla por ser la primera iteración se pretendió agregar cuatro, sin embargo, en las siguientes iteraciones se modificaron, para presentar la opción de figuras y colores se seleccionaron los más básicos y conocidos por el usuario.

2.1.2. Diseño.

Después de realizar una planificación se dio paso al siguiente proceso, en el que se utilizó un tipo de diseño conceptual, de acuerdo a los conceptos mencionados en el marco teórico, este tipo de diseño permite realizar maquetas o diagramas.

Posteriormente se definió la forma inicial de presentación al usuario, para diseñarlo se empleó una herramienta que permitió modelar y maquetar el lugar en el que se ubicarían los componentes, dando a entender que será solo visual. La herramienta

usada para maquetar fue JustInMind, debido a que posee muchas bondades para el diseño de maquetas web, permitiendo guardar el boceto en diferentes formatos, ya sea como imagen o en un documento de texto. Cabe mencionar que esta herramienta se la utilizó en todas las iteraciones.



Para el desarrollo visual del sitio web se tomó en cuenta el diseño que algunas instituciones educativas muestran, así también atendiendo a los parámetros indicados en el capítulo uno de los conceptos teóricos, en el que indica que un sitio web debe ser simple e intuitivo y el usuario tenga facilidad de interacción. Por lo que se tomó como muestra el posicionamiento inicial de las materias presentadas en el aula virtual de la Universidad Politécnica Salesiana, permitiendo tener como base inicial el mismo posicionamiento.

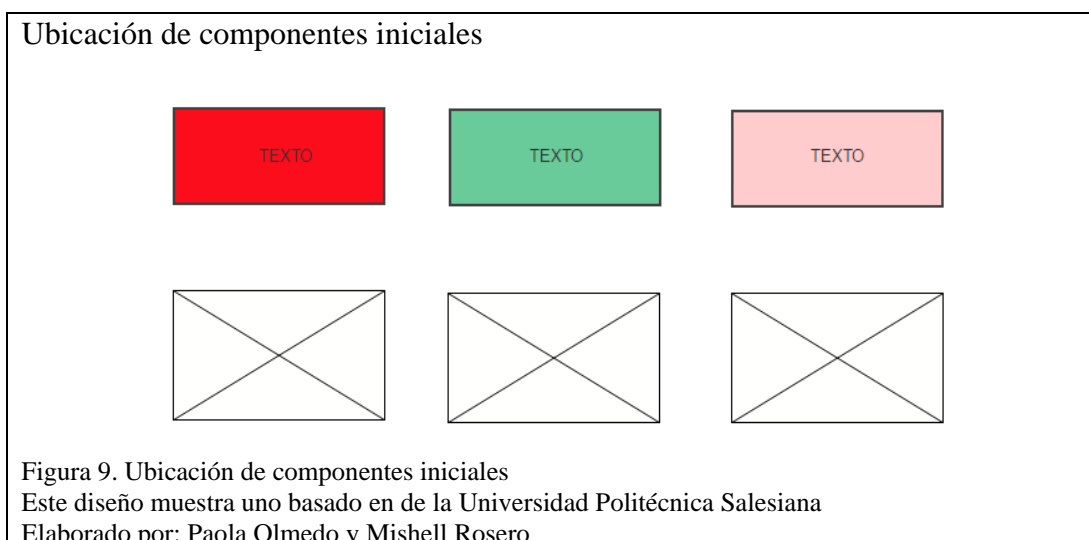
La figura No. 8 evidenció que en el lado izquierdo o side, se muestra un menú en el que se encuentra ubicado un elemento llamado “Opciones”, este menú cuenta con alternativas a escoger, mostrado en la tabla No. 5.

Tabla 5. Alternativas a escoger en el menú “Opciones”

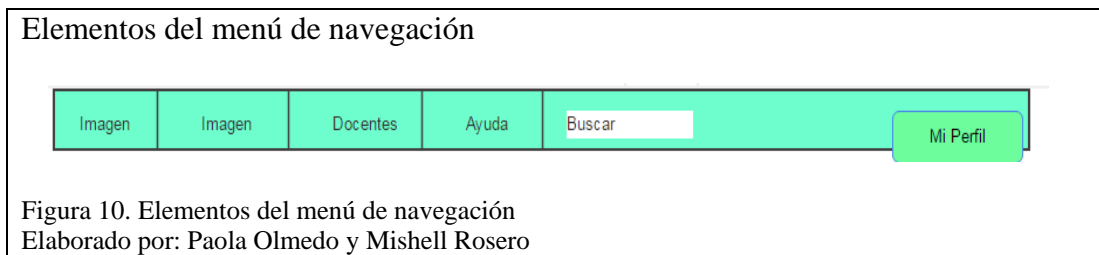
Alternativas	Uso
Administración del Curso	Mostrar las materias matriculadas por el estudiante
Mensajes	Mensajes que permitirán comunicación entre estudiante y profesor
Calificaciones	Mostrar calificaciones por cada materia
Personalizar temas	Mostrar menú de opciones para personalizar los componentes
Calendario	Mostrar calendario en forma general, no tendrá ninguna funcionalidad, más que para ver fecha

Nota: Alternativas que el usuario puede escoger una vez que se encuentra en el sitio

En la cabecera se ubicaron las materias, que son los componentes que más adelante se ubicarán en diferentes sitios dentro de la pantalla, para eso se diseñó un formato que permitirá la ubicación de estas. La figura No. 9 presenta su ubicación.



La figura No. 10 presenta los elementos que serán ubicados en el menú de navegación, permitiendo redirigir estos a otros sitios.



En la tabla No. 6 se detalla la función de cada uno de estos elementos.

Tabla 6. Elementos que forman parte del menú de navegación

Elemento	Uso
Imagen	Se pretende incorporar una imagen como logo del ambiente virtual
Home	Redirigir al usuario a la página principal
Docentes	Mostrar información de los docentes
Ayuda	Mostrar temas de uso del sitio web
Buscar	Permite buscar las materias o perfil del profesor
Mi Perfil	Mostrar información del estudiante permitiendo realizar cambios

Nota: Tabla que muestra el uso de los elementos ubicados en la barra de navegación

Lo mencionado anteriormente solo es la forma visual, permitiendo mostrar la presentación del sitio web, sin embargo, conforme se van dando las iteraciones irán cambiando, ya que la primera iteración ayudó a partir a una idea más clara.

2.1.3. Prototipo.

Junto con un diseño se realizó un primer prototipo incorporando código que le da funcionalidad, el lenguaje en el que se planificó elaborar fue JavaScript ya que junto con HTML5 y usando CSS permiten hacer que un sitio web sea dinámico.

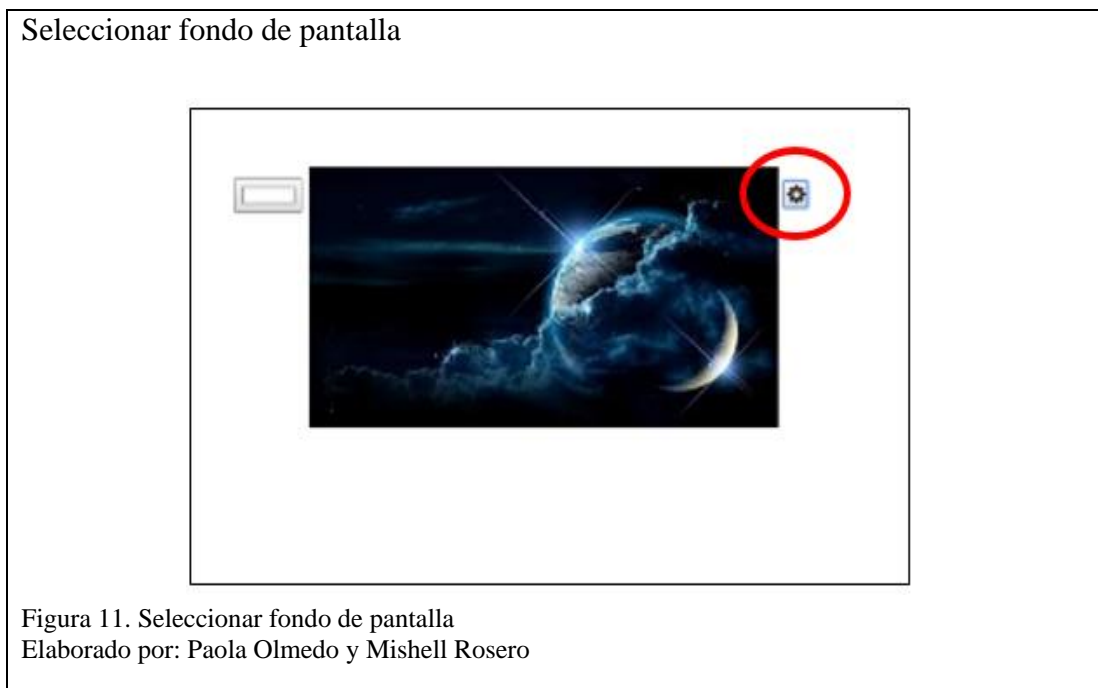
Se usó AngularJs como framework, debido a que posee algunos beneficios, tales como la distribución de sus componentes en una forma que se pueda trabajar, para demostrar

la posibilidad de realizar una personalización solo del lado del cliente, con una reducción considerable de recursos.

2.1.3.1. Añadir fondo de pantalla.

Se inició con la inserción del fondo de pantalla, ya que se consideró que es el más sencillo de realizar, con la ayuda del IDE especializado WebStorm, que es para el desarrollo web con tecnologías JavaScript, se realizó la primera presentación en el navegador.

La figura No. 11 muestra un botón que se colocó, para que, al dar click traiga diferentes fondos desde una base de datos. Se utilizó el gestor de MySQL para almacenar los archivos.



Tal y como muestra la figura No. 11, al dar click en el botón de administración se abre opciones para seleccionar como fondos de pantalla.

Opciones de fondos de pantalla a agregar

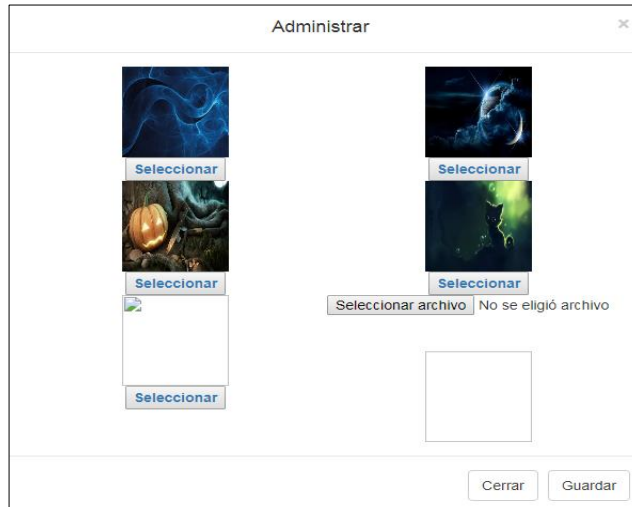


Figura 12. Opciones de fondos de pantalla a agregar
Elaborado por: Paola Olmedo y Mishell Rosero

De acuerdo a la figura No. 12, se evidenció una ventana emergente en la que se puede seleccionar de los fondos de pantalla disponibles, o al dar click en "Seleccionar archivo" cargará uno nuevo desde el ordenador del usuario.

2.1.3.1.1. Código implementado.

La figura No. 13 muestra un fragmento de código de la ventana emergente que permite seleccionar el fondo de pantalla o cargar uno nuevo.

Fragmento de código para selección de fondo de pantalla

```
<div ng-repeat="f in fondo">
  <input name="color" type="color" id="myColor"
  onchange="mvFunction()" />
  <script>
    function mvFunction() {
      var x =
document.getElementById("myColor").value;
      document.body.style.background = x ;
    }
  </script>
  
```

Figura 13. Fragmento de código para seleccionar un fondo de pantalla
Elaborado por: Paola Olmedo y Mishell Rosero

Se usa la directiva “ng-repeat” para mostrar todos los fondos de pantalla disponibles dentro de la etiqueta “” y se da la posibilidad de importar nuevos fondos con la etiqueta “<input>”, se usa un pequeño script para insertar la imagen como fondo de pantalla con el uso “style.backgroud”, al cerrar el pop-up se llama a la función "cambioFondo" con la directiva “ng-click”.

La figura No. 14 presenta que con la llamada a la función "cambioFondo" se recibe como parámetro la imagen, misma que es enviada a la base de datos, mediante un requerimiento \$http se envía a la base la imagen mediante código php, se envía como parámetro la dirección recibida y se ejecuta la directiva "window.location.reload()" para recargar la página con el fondo de pantalla seleccionado.

Función para cambio de fondo de pantalla

```
$scope.cambioFondo=function (Direccion) {  
    console.log(Direccion);  
    var request=$http({  
        method: "POST",  
        url:  
        "http://localhost/cambioFondoPantalla/ws/cambioFondoPantalla.php",  
        data: {  
            Direccion:Direccion  
        },  
        headers: {'Content-Type':  
'application/x-www-form-urlencoded'}  
    });  
    $window.location.reload();  
}
```

Figura 14. Código que contiene la función a la llamada de los parámetros
Elaborado por: Paola Olmedo y Mishell Rosero

2.1.3.2. Ubicación de componentes al centro de la pantalla.

Para la ubicación de los componentes se realizó en el IDE un nuevo proyecto, una vez probada la inserción de un fondo de pantalla, se procedió a unir los tres requerimientos faltantes.

Ubicación de componentes iniciales

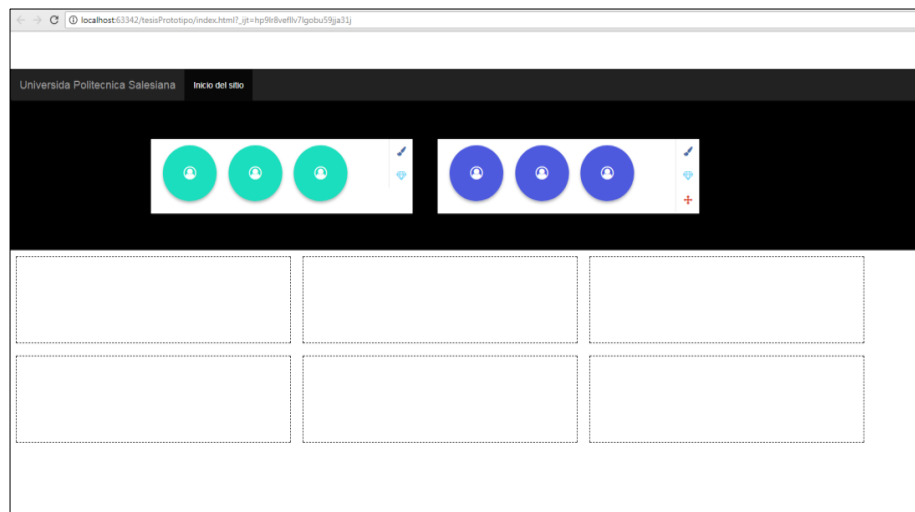


Figura 15. Ubicación de componentes iniciales
Elaborado por: Paola Olmedo y Mishell Rosero

De acuerdo a la figura No. 15 se incorporaron cuadros punteados bajo los componentes que se encuentran en la cabecera, estos cuadros se mantienen desde la idea del proceso de diseño, donde se van a ubicar en el centro del sitio cuadros que permitan la incorporación de los componentes. Estos cuadros se los realizó con un contenedor de etiquetas conocido como “div”, que crean secciones, permitiendo tener una mejor ayuda al momento de incorporar los componentes.

2.1.3.2.1. Código implementado.

La figura No. 16 muestra un fragmento del código que se utilizó para el movimiento de los componentes y ubicarlos al centro, para el cambio de ubicación se empleó la función drag & drop.

Fragmento de código que permite mover los componentes

```
<div class="col-sm-4" id="div1" ondrop="drop(event)"
ondragover="allowDrop(event)" style="width: 460px;
height: 140px">
  <ul class="event-list" draggable="true"
ondragstart="drag(event)" id="drag1">
```

Figura 16. Fragmento de código que muestra el movimiento de componentes
Elaborado por: Paola Olmedo y Mishell Rosero

El elemento más importante que se incorporó es “drag” y “drop”, que están insertados dentro de un contenedor de etiquetas, para tener un arrastre específico hacia su ubicación definitiva.

La tabla No. 7 evidencia los elementos que se usaron en la utilización del drag & drop.

Tabla 7. Función de cada uno de los elementos del drag & drop

Elemento	Función
Ondrop	Acción que permitió llamar a la función para que pueda soltar los componentes
Drop	Función que fue creada dentro de la vista
Ondragover	Acción que llama a la función para que permita mover
Allowdrop	Función que permite incorporar los componentes
Ondragstart	Acción que permite llamar a la función que arrastra a los elementos
Drag	Función que permitió arrastrar los elementos

Nota: Tabla que muestra información de los elementos que se utilizó con el drag & drop

En la figura No. 17 se muestra el código de la cada una de las funciones que está llamando desde las acciones drag y drop.

Funciones del drag y drop

```
<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>
```

Figura 17. Funciones del drag y drop
Elaborado por: Paola Olmedo y Mishell Rosero

Cada una de las funciones permitió mover, soltar e incorporar los contenidos en cada uno del contenedor de etiquetas previamente creado. La función drag & drop se lo realizó manteniendo ciertos parámetros, tales como, que el usuario no pueda colocar los componentes fuera de los lados, o los coloque en el pie de página. Drag & drop es propio de JavaScript, se lo puede usar en cualquier framework.

El propósito fue que al dar click en uno de los dos componentes que están en la cabecera se los mueva, si se selecciona el del lado izquierdo, se mueven todos y si se selecciona los del lado derecho es el mismo movimiento, al mantener presionado el componente presenta la opción de arrastre a uno de los puntos sugeridos. Si el usuario intentase mover los componentes a otro lugar que no sean los cuadros no se posicionarán.

2.1.3.3. Cambio de color de componentes.

Otro de los objetivos iniciales fue cambiar el color de los componentes, así que se decidió incorporar la opción de cambiar el color al dar click en un icono que se lo integró en el mismo cuadro que contiene los componentes.

El icono que permitió cambiar el color es el que tiene la forma de pincel, la idea de escoger al pincel para el cambio del color fue propio de diseño, ya que se deben escoger elementos que el usuario pueda asociar, de acuerdo a los conceptos teóricos mencionados en el capítulo uno.



Según la figura No. 18, al dar click en el icono antes mencionado, se abre un pop-up en el que se muestra una fila de botones, permitiendo seleccionar uno para cambiar el color de los componentes.

2.1.3.3.1. Código implementado.

Fragmento de código que muestra los colores y permite elegirlos

```
<div class="social">
  <div class="modal-content">
    <div class="modal-header">
      <button type="button" class="close" data-v
dismiss="modal">&times;</button>
      <h4 class="modal-title">Selecciona un color Menu
uno</h4>
    </div>
    <div class="modal-body" class="row">
      <button type="button" style=" padding:20px 40px
20px 35px;background-color: #de5fcf" class="btn btn-
default" ng-click="cambioColoMenuUnor(colorRosado)" data-
dismiss="modal" ></button>
    </div>
  </div>
</div>
```

Figura 19. Fragmento de código que permite la llamada a la función
Elaborado por: Paola Olmedo y Mishell Rosero

Tal y como se observó en la figura No. 19, dentro de la ventana emergente se muestran los botones cargados con un color predefinido utilizando el atributo "style".

Para llamar la función "cambioColoMenuUnor" se utilizó la directiva "ng-click", pasa como parámetro el nombre del color el cual se lo coloca como atributo al objeto "miColorMenuUno" como se puede observar en la figura No.20.

Función para la selección de colores

```
$scope.cambioColoMenuUnor=function (color) {
  console.log(color);
  $scope.miColorMenuUno. backgroundColor=color;
};
```

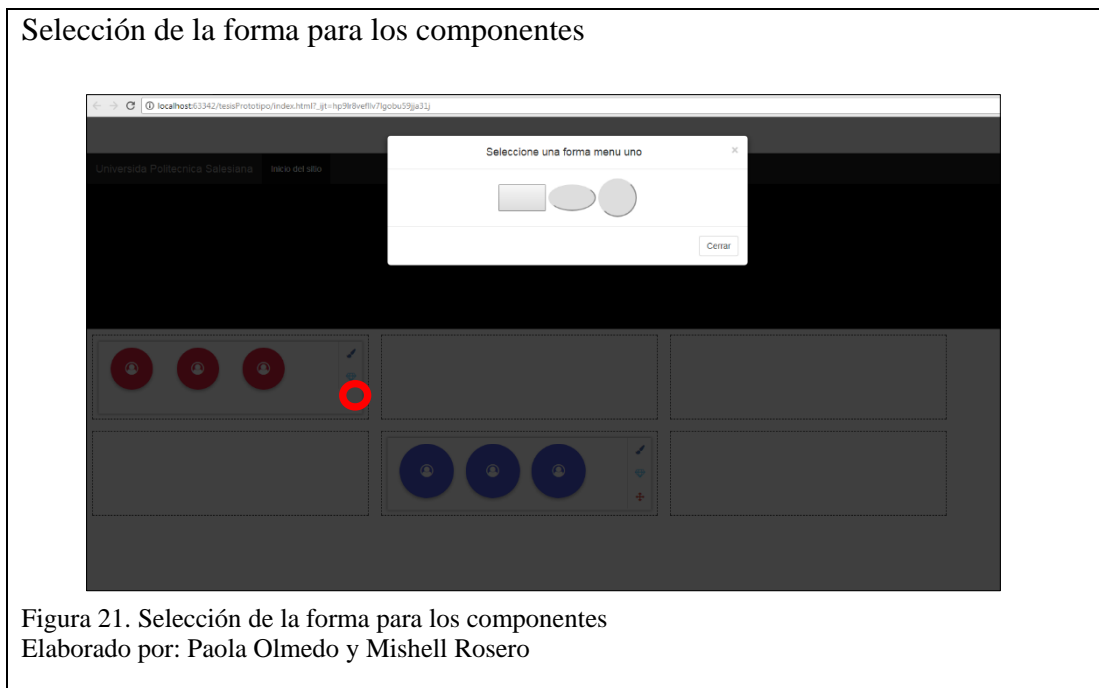
Figura 20. Función para la selección de colores
Elaborado por: Paola Olmedo y Mishell Rosero

Los controladores permiten gestionar los eventos, enlazando la vista mediante la directiva "ng-controller" y se lo declara en el cuerpo del código HTML.

El objeto "\$scope" permite que el controlador pueda interactuar con la vista agregando nuevas funciones o atributos a los objetos que se muestran en el código HTML.

2.1.3.4. Cambio de forma de componentes.

Una vez demostrado que es posible el cambio de ubicación y color de componentes, se procedió a cambiar la forma de estos, para ellos se pensó en las figuras que pueden ser las más habituales y conocidas por los usuarios. Del mismo modo en que se pensó en el icono para el cambio de color, fue para el cambio de forma, eligiendo el icono de gema, al dar click se presenta una ventana emergente con las formas pre establecidas.



Como se evidenció en la figura No. 21, se posee una fila de tres figuras distintas, al dar click en alguna de ellas se cambia la forma de los componentes por la forma seleccionada.

2.1.3.4.1. Código implementado.

En la figura No. 22 se observa que dentro de la ventana emergente se muestran los botones con una forma predefinida.

Fragmento de código de la ventana emergente con figuras

```
<div class="modal fade" id="modalFormasMenuUno"
  <div class="modal-content">
    <div class="modal-header">
      <button type="button" class="close" data-
dismiss="modal">&times;</button>
      <h4 class="modal-title">Seleccione una
forma menu uno</h4>
    </div>
    <div class="modal-body" class="row">
      <button class="button button4" ng-
click="cambioFormaMenuUno('cuadrado')" data-
dismiss="modal" style="padding: 20px 40px 20px
35px;"></button>
    </div>
  </div>
```

Figura 22. Fragmento de código de la vista que llama a la función
Elaborado por: Paola Olmedo y Mishell Rosero

Para llamar la función "cambioFormaMenuUno" se utilizó la directiva "ng-click", esta pasa como parámetro el nombre de la figura con la cual entra a un proceso de condición para definir la forma que llega como parámetro, como se observa en la figura No. 23.

Función que permite seleccionar el tipo de figura

```
angular.module('app', [])
  .controller("controllerImagenes", function($scope, $window)
  {
    var mostrarEditor= false;
    var correctSave=false;
    $scope.options:
    $scope.circulo;
    $scope.rectangulo;
    $scope.oval;

    $scope.cambioFormaMenuUno=function (forma) {
      console.log(forma)
      if(forma=='oval') {
        $scope.miFormaMenuUno =
        document.getElementById("menuUno").style.width = "80px";
        $scope.miFormaMenuUno =
        document.getElementById("menuUno").style.height = "40px";
        $scope.miFormaMenuUno =
        document.getElementById("menuUno").style.borderRadius="50%";
      }
    }
  });
```

Figura 23. Función que permite seleccionar el tipo de figura
Elaborado por: Paola Olmedo y Mishell Rosero

Dependiendo la condición que se haya cumplido se modifican los atributos de cada componente aumentando ancho, alto o radio, en este caso se indica los atributos que fueron modificados cuando se cumple la condición de “óvalo”, el mismo método se implementó en las demás figuras.

2.1.4. Evaluación.

La evaluación se la realizó una vez terminado el prototipo de esta iteración, comprobando que se puede efectuar lo que se requirió desde el inicio, tal como se evidencia en la figura No.24, los componentes ya se encuentran ubicados en diferentes posiciones y con sus respectivas formas y colores.

Evaluación de los componentes probados

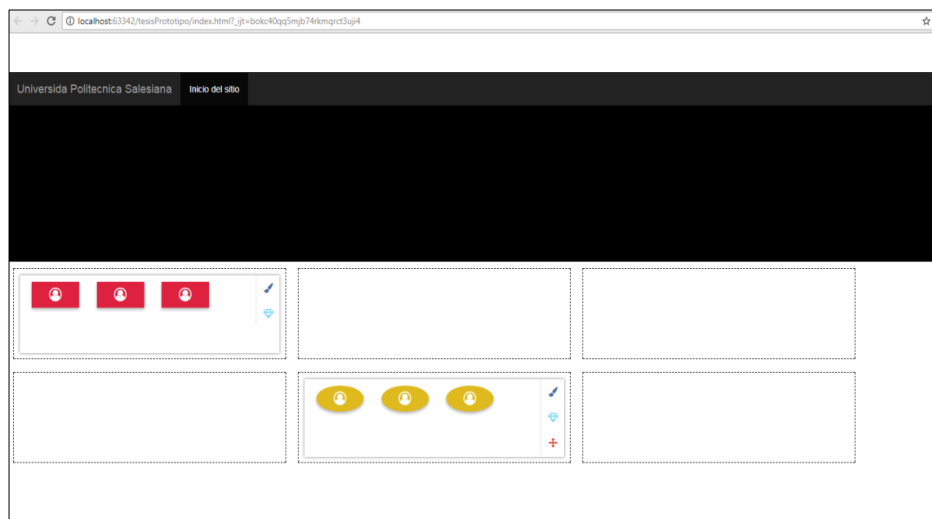


Figura 24. Evaluación de los componentes probados
Elaborado por: Paola Olmedo y Mishell Rosero

El cambio de versión de AngularJs fue un desafío, ya que esta versión incorpora nuevas mejoras y formas de llamar a los componentes. La versión en la que se desarrolló el proyecto de titulación anterior fue 1.4.2 y la versión en la que se desarrolló ahora es la 1.5.8.

Al realizar la primera iteración aún no se tuvo claro el número de iteraciones que se iban a tener, sin embargo, ya se logró conseguir lo planificado que fue evaluar que es posible realizar una personalización usando tecnologías del lado del cliente.

Para finalizar esta iteración se logró entender que es posible la personalización consumiendo recursos del lado del cliente, sin embargo, existen recursos tales como el almacenamiento, que es el que trabajará netamente del lado del servidor, debido a que la carga de imagen para el fondo de pantalla se puede hacer tanto del lado del cliente, como del servidor, sin embargo, el movimiento de los componentes, el cambio de color y forma de los mismos si se los realiza del lado del cliente.

Capítulo 3

Segunda Iteración

3.1. Primer prototipo funcional

Este capítulo se centra en la iteración que tuvo como fin incorporar nuevas funciones a lo realizado en el capítulo anterior, manteniendo los requerimientos iniciales. Para esta iteración ya se logró unificar lo creado, manteniendo los conceptos claros del diseño en un sitio web, siguiendo con el desarrollo, se pretendió realizar un prototipo más funcional. En esta iteración ya no se realizó el proceso de la planificación, puesto que los requerimientos siguen siendo los mismos.

Para esta iteración se cambió la manera de usar de AngularJs para una mejor funcionabilidad y generar una aplicación más robusta, se estructuró el código de una forma eficiente dividiendo las pantallas en componentes para evitar la carga innecesaria de toda la aplicación web al iniciar la página principal y mejorar sus tiempos de respuesta.

La figura No. 25 evidencia la estructura de la forma de distribuir los componentes de AngularJs.

Árbol de estructura de las funciones de AngularJs

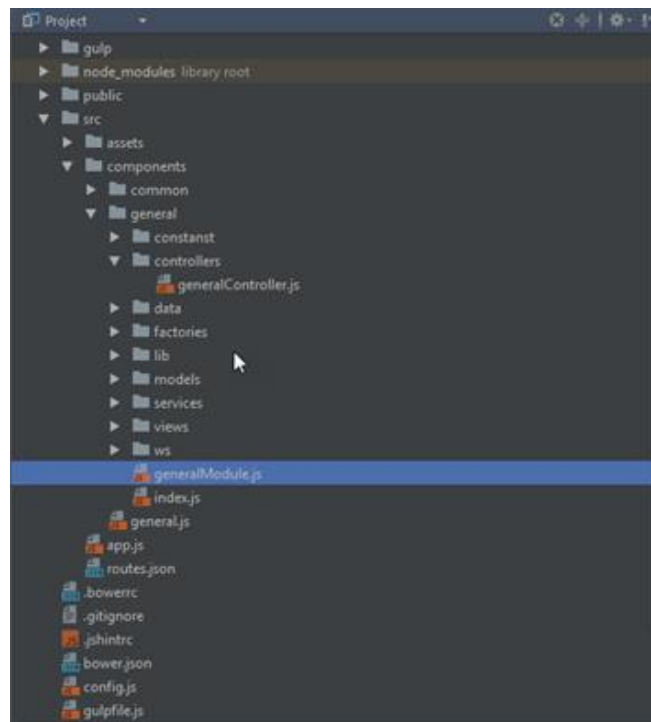


Figura 25. Árbol de estructura de las funciones de AngularJs
Elaborado por: Paola Olmedo y Mishell Rosero

3.1.1. Diseño.

Tal como en la iteración anterior, el tipo de diseño es el conceptual, debido a que es un sitio web visual. La herramienta para realizar la maqueta sigue siendo JustInMind.

Diseño de la presentación inicial del sitio web

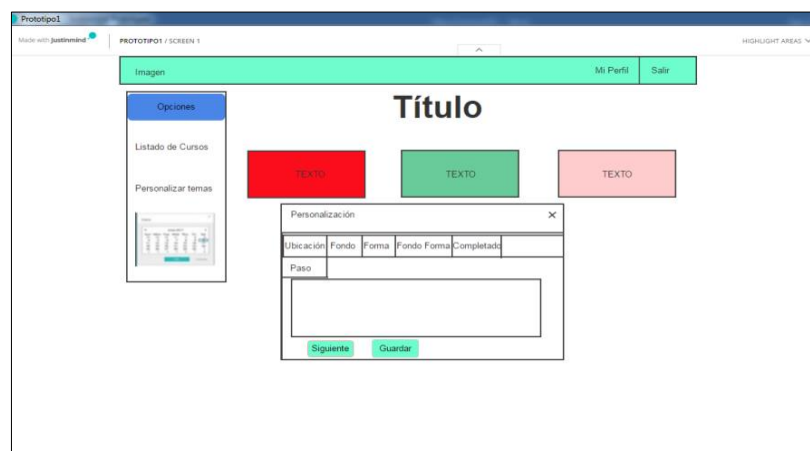


Figura 26. Diseño de la presentación inicial del sitio web
Elaborado por: Paola Olmedo y Mishell Rosero

De acuerdo a la figura No. 26, el diseño es diferente al que se evidenció en la primera iteración, ya se incorporaron nuevos elementos, sin embargo, el enfoque no ha cambiado. Se pensó en darle un aspecto más visible y llamativo. La idea de cambiar la apariencia nació desde la usabilidad, tratando de mostrar al usuario un sitio web más atractivo, en el que sea de fácil uso.

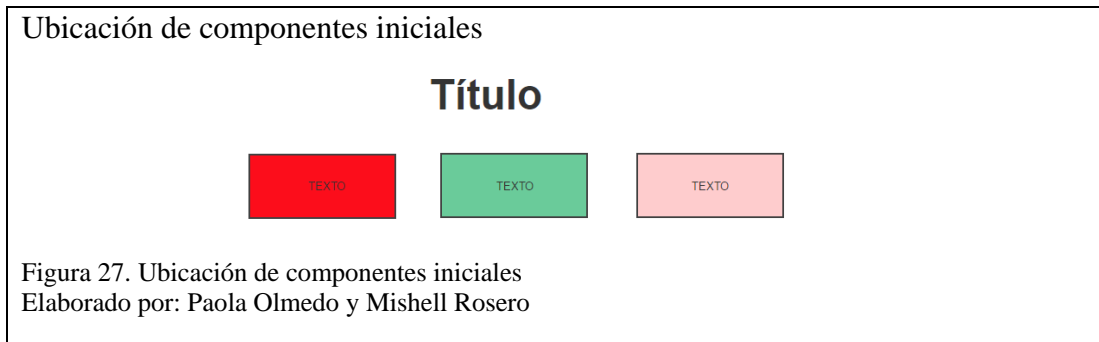
Se incorporó una ventana emergente o pop-up, que permite al usuario presentar un proceso para que vaya guardando los cambios. El pop-up se lo puede llamar dentro de la misma ventana. La idea de incorporar esta herramienta fue para que al usuario no se le haga tedioso abrir de uno en uno los cambios que va a realizar, aquí se mostrarán las opciones para personalizar su sitio web, será llamado desde el menú de “Opciones”.

En la figura No. 26 se observó que el menú de navegación es el mismo desde el inicio, en la imagen del sitio ya se estableció un nombre para la aplicación, se pensó en el nombre y logo llamado "AVE" (Aula Virtual Estudiantil), puesto que este sitio web es enfocado a un aula virtual.

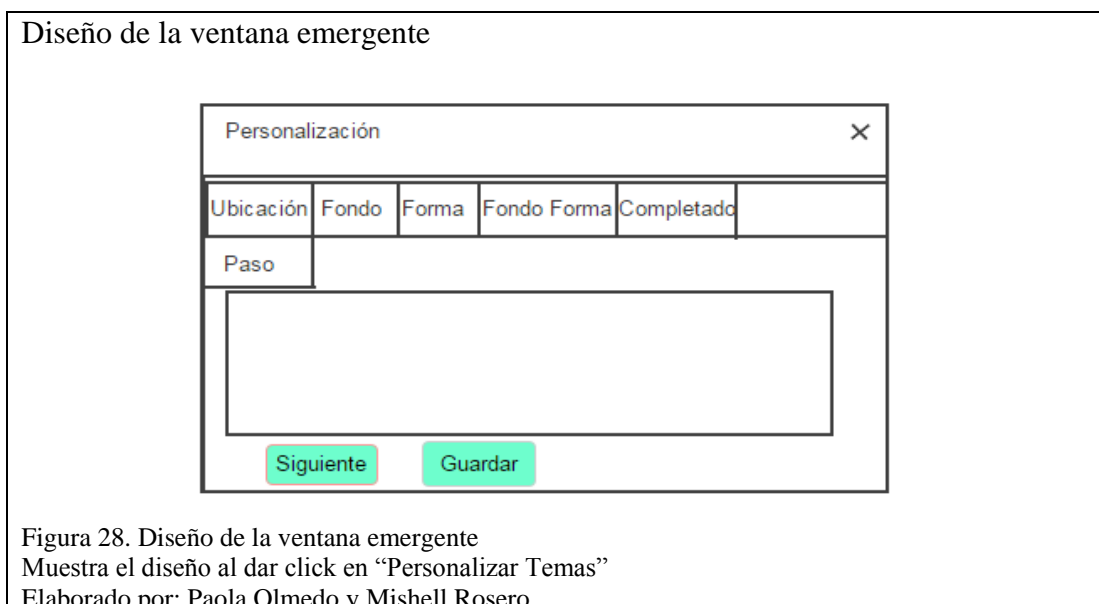
En el menú de “Opciones” se mantienen los elementos iniciales, excepto: "Administración del Curso", "Mensajes" y "Calificaciones". Se los eliminó ya que el usuario estudiante no tiene permisos para administrar sus cursos, por lo tanto, tampoco fue necesario mantener las calificaciones, porque estas deben ir dentro de “Administración del Curso”.

Los "Mensajes" se los incorporó dentro del “Listado de Cursos”, para que el usuario estudiante pueda tener acceso a ellos. En la cabecera se añadió un texto llamado “Título” y más abajo en el centro están ubicados los nombres de las materias que se tuvo desde la iteración anterior, a las que se les personalizará junto con el “Título”.

Al “Título” aún no se le asigna un nombre, puesto que se necesitó agrupar lo realizado en la iteración anterior, en el capítulo cuatro ya se les asignará nombres. Los componentes se los ubicó en el centro del sitio web con el fin de cumplir con conceptos de usabilidad, que se deben colocar los componentes en lugares que el usuario pueda verlos.



La ventana emergente se abre al dar click en la opción del menú “Personalizar Temas”, al abrirse esta ventana presenta diferentes pestañas que le van a permitir al usuario personalizar los componentes que se encuentran en el centro de la pantalla, mostrados en la figura No. 27. La ventana emergente o pop - up se va a presentar sobrepuesta a los componentes antes mencionados, que al cerrar ya se observa el cambio realizado.



3.1.1.1. Definición de elementos de la ventana emergente.

3.1.1.1.1. Ubicación.

Se tienen tres alternativas de plantillas para ubicar los componentes, estas plantillas se las seleccionó realizando una encuesta previa. La encuesta consistió en escoger a 34 alumnos de la Universidad Politécnica Salesiana, a los cuales se les presentó 10 opciones de plantillas, indicándoles que den una calificación del 1 al 10, siendo 1 la más baja y 10 la más alta, las plantillas que les gustaría que se muestren en un aula virtual para personalizar, siendo el caso de la ubicación de componentes. De acuerdo a la encuesta las plantillas que se mostrarán son: 1, 2 y 4. [Anexo 1](#).

El usuario tiene la posibilidad de seleccionar cualquiera de las tres y ubicar los componentes según su elección. Si el usuario no desea ninguna de estas plantillas también tiene a su disposición crear una propia. Se lo realizó con el fin de que el sitio sea más usable y el usuario tenga más oportunidad de personalizarlo.

3.1.1.1.2. Fondo de pantalla.

Para seleccionar el fondo de pantalla se realiza el mismo procedimiento, se presentan tres opciones ya preestablecidas para darle un fondo al sitio web, o el usuario tiene la posibilidad de traer un fondo propio desde su ordenador, tal y como se hizo en la iteración anterior.

3.1.1.1.3. Cambio de fondo de color y forma.

Estas dos opciones, así como las anteriores también presentarán alternativas a escoger, la selección tanto como para forma y color son las mismas desde el proceso de planificación que se explicó en el capítulo anterior.

3.1.1.1.4. Guardar cambios.

El texto más abajo llamado “Paso”, indica al usuario en que opción o paso de la personalización se encuentra, si ya realizó el cambio al dar click en el botón "Siguiete" pasará a la siguiente opción y si ya finaliza y está a su gusto, el usuario puede proceder a guardar. Al guardar los cambios, estos se almacenan en una base de datos de MySQL, con un código según el usuario que haya realizado la personalización, estos cambios se almacenaron localmente. Se crearon en la base de datos dos tablas, una llamada "Estudiantes" y otra llamada "Temas", en esta última se almacenarán los cambios realizados. A continuación, se muestran los campos que se encuentran en la tabla “Temas”.

Tabla 8. Descripción de cada uno de los campos de la tabla "Temas"

Nombre del Campo	Descripción
Id Tema	Código autogenerated
Código de plantilla	Código que se almacena según la plantilla: plantilla uno: código 1 plantilla dos: código 2 plantilla tres: código 3
Nombre de imagen	Se almacena el nombre de la imagen que seleccionó como fondo de pantalla: fondo 1: código a fondo 2: código b fondo 3: código c
Nombre de la forma	Se almacena el nombre de la forma que seleccionó al cambiar en el sitio web
Nombre del color	Se almacena el nombre del color que seleccionó al cambiar en el sitio web
Estado del estudiante	Será 1 si es un estudiante que esté matriculado en el curso o 2 si es un estudiante no matriculado
Id estudiante	Código del estudiante que se almacenó en la tabla estudiantes

Nota: descripción de cada uno de los campos de la tabla “Temas”

A continuación, se muestra la información de los campos de la tabla “Estudiante”

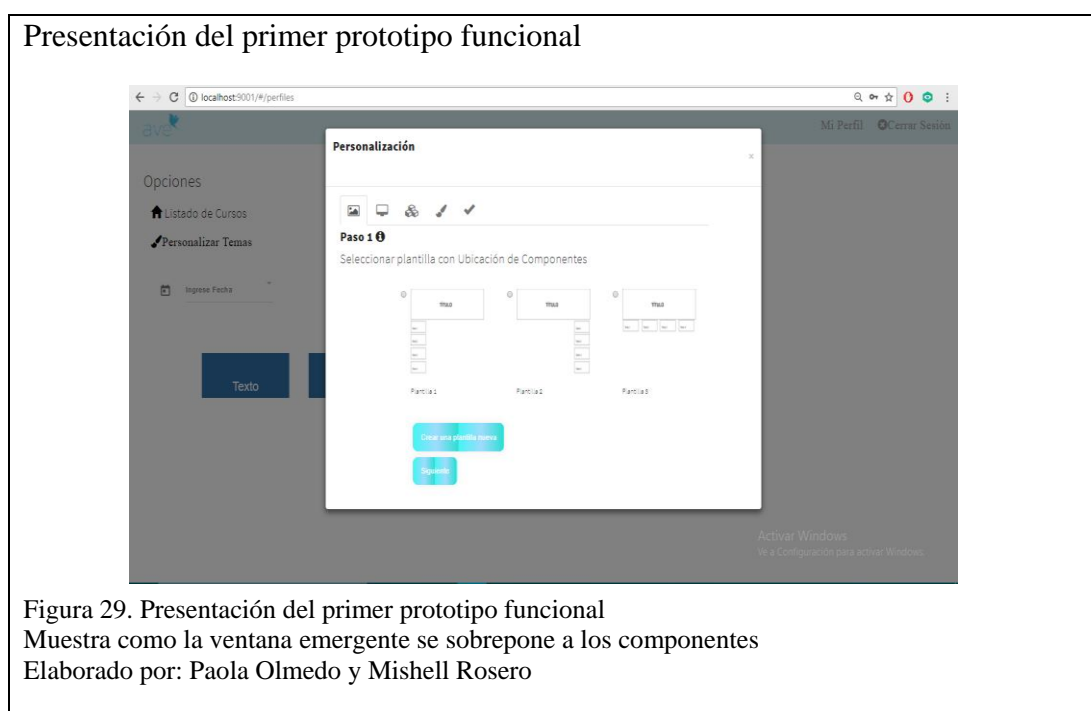
Tabla 9. Detalle de los campos de la tabla "Estudiante"

Nombre del campo	Detalle
Id estudiante	Código autogenerated
Nombre	Nombre del estudiante
Contraseña	Contraseña asignada al estudiante
Estado	Código de estado del estudiante
Nombre usuario	Nombre de usuario para ingresar al sitio web

Nota: Tabla de información de los campos que contiene la tabla “Estudiante”

3.1.2. Prototipo.

Después de hacer el diseño se procedió a realizar el prototipo, la figura No. 29 muestra el prototipo de acuerdo al diseño.



3.1.2.1 Ubicación de componentes.

En la ubicación de componentes se mantiene el drag & drop, que fue realizado anteriormente, al seleccionar el tipo de plantilla los componentes se ubicarán tal y

como muestra la plantilla. Para la ubicación se mantuvo los contenedores de etiqueta o "divs", tal y como fue el primer prototipo. Se usó el código de identificador de cada uno de ellos en donde se están ubicando los botones que son los componentes, se pensó en ocultar y mostrar los botones.

Ventana emergente que presenta la selección de las plantillas



Figura 30. Ventana emergente que presenta la selección de las plantillas
Muestra las opciones de plantillas que el usuario puede seleccionar
Elaborado por: Paola Olmedo y Mishell Rosero

3.1.2.1.1. Código implementado.

Se implementó un wizard o asistente de configuración, dentro de la ventana emergente, en el que se controlan las etapas de la personalización.

Fragmento de código que muestra la ubicación

```
<div class="modal-body" class="row">
  <div class="container">
    <div class="row">
      <section class="col-sm-7" >
        <div class="wizard">
          <div class="wizard-inner"...>
            <form role="form">
              <div class="tab-content">
                <div class="tab-pane active" role="tabpanel" id="step1"...>
                <div class="tab-pane" role="tabpanel" id="step2"...>
                <div class="tab-pane" role="tabpanel" id="step3"...>
                <div class="tab-pane" role="tabpanel" id="step4"...>
                <div class="tab-pane" role="tabpanel" id="complete"...>
              </div>
            </form>
          </div>
        </section>
      </div>
    </div>
  </div>
</div>
```

Figura 31. Fragmento de código que muestra la ubicación
Muestra la llamada a la función para ubicar de acuerdo a cada plantilla
Elaborado por: Paola Olmedo y Mishell Rosero

Se cargaron los diseños de las plantillas en el primer paso del wizard, los cuales se pueden seleccionar y se establecen los componentes principales según la plantilla seleccionada, en el segundo paso se cargaron las imágenes pre establecidas para el fondo y la posibilidad de cargar una nueva imagen desde el ordenador del usuario, en el tercer paso se cargaron las formas y en el cuarto paso se cargaron los colores para el cambio de fondo de los componentes. Estos métodos son los usados en las anteriores iteraciones, se actualizaron a la nueva estructura para su uso, todas las selecciones del usuario son guardadas en objetos temporales dentro del modelo para armar la estructura, para después ser guardada. Para finalizar en el último paso del wizard se envían los datos seleccionados a la base de MySQL. Se usó la directiva "ng-click" para llamar a la función "guardarDiseño" como se puede observar en la figura No. 32.

Función que permite mostrar u ocultar las plantillas

```
<input ng-model="colorBotones" type="radio" name="opradio" value="amarillo" />
<a style="..." class="btn btn-default"></a>
</div>
<div class="col-sm-1"></div>
<div class="radio col-sm-2">
  <input ng-model="colorBotones" type="radio" name="opradio" value="azul" />
  <a style="..." class="btn btn-default"></a>
</div>
</form>
<br>
<br>
<br>
<br>
<br>
<button href="#step1" data-toggle="tab" aria-controls="step1" role="tab" type="button" class="btn btn-1">Antes</button>
<button href="#complete" data-toggle="tab" aria-controls="complete" role="tab" type="button" class="btn btn-1"
  ng-click="centerPerfilesCtrl.comprobarColor(colorBotones)" >Siguiente</button>
</ul>
</div>
<div class="tab-pane" role="tabpanel" id="complete">
  <h3>Finalizado</h3>
  <p>Se han completado satisfactoriamente todos los pasos</p>
  <!-- href="#step1" data-toggle="tab" aria-controls="step1" -->
  <button type="button" data-target="#modalFinPersonalizacion" data-toggle="modal" data-backdrop="static"
    data-keyboard="false" class="btn btn-1" data-dismiss="modal" ng-click="centerPerfilesCtrl.guardarDiseno()">Guardar </button>
</div>
</div>
```

Figura 32. Función que permite mostrar u ocultar las plantillas
Elaborado por: Paola Olmedo y Mishell Rosero

La función "guardarDiseno" envía una petición "http" mediante código php, armando todos los datos necesarios en la tabla de personalización definiendo la información con el nombre de los campos en la tabla y el valor almacenado en el modelo de la selección del usuario, tal y como se observa en la figura No. 33.

Fragmento de código que contiene la función

```
guardarDiseno(){
  var request = this.$http({
    method: 'POST',
    url: 'http://45.77.194.153/aveServer/ave-ng/src/components/ws/guardarPersonalizacion.php',
    data: {
      plantillaPantalla: this.plantillaPantalla,
      imagenPantalla: this.imagenPantalla,
      formaBoton: this.formaBoton,
      colorBoton: this.colorBoton,
      username: this.username,
      token: this.tokenUser
    },
    headers: {'Content-Type': 'application/x-www-form-urlencoded'}
  });
}
```

Figura 33. Fragmento de código que contiene la función
Elaborado por: Paola Olmedo y Mishell Rosero

3.1.3. Evaluación.

En esta parte del proceso se hicieron las pruebas, comprobando que es posible guardar los cambios realizados, según el código del estudiante, estos cambios aún se estaban guardando localmente. La figura No. 34 muestra un sitio web ya personalizado.



Para finalizar esta iteración es necesario mencionar que no se realizó la creación de la nueva plantilla ni la selección de un fondo de pantalla desde el ordenador del usuario, debido a que se planea solo almacenar con códigos de cada uno de estas.

La opción “Mi Perfil” y “Listado de Cursos” tampoco se realizó, porque se pretendía crear los usuarios más adelante desde Moodle y así el usuario almacene sus cambios de perfil. Para guardar las actualizaciones de esta iteración y de la siguiente fue necesario hacer uso de un gestor de versiones de código que tiene una integración con la nube, se utilizó Bitbucket, que es un repositorio Git en la nube, junto con el IDE WebStorm permitieron almacenar los cambios y no perderlos durante el proceso.

Capítulo 4

Tercera Iteración

4.1. Prototipo Producto final

El propósito de este capítulo es describir la iteración, cuyo fin fue almacenar los datos en Moodle y tener listo el producto para en la siguiente iteración llevar a cabo las pruebas.

Con un prototipo más robusto, se realizó lo que quedó pendiente en la iteración anterior que fue: creación de una nueva plantilla, selección de fondo de pantalla desde el ordenador del usuario y darles funcionalidad a las opciones “Mi Perfil” y “Listado de Cursos”.

Se instaló Moodle sobre el sistema operativo Centos, versión 16.04 en un servidor externo alojado en Vultr, que es un proveedor de servicios en la nube, se realizó la configuración necesaria para habilitar los servicios móviles y los servicios de comunicación a través del protocolos REST y recibir las respuestas desde Moodle en formato "json"(documento que permite almacenar datos) para facilitar su manipulación, se probaron sus funcionalidades, se crearon los datos de los usuarios y se analizó la estructura tanto de Moodle como del gestor de base de datos que es MariaDB. Se creó una nueva tabla dentro de la base llamada "mdl_personalizacion" siguiendo las reglas y restricciones de Moodle para no afectar a las funciones existentes, en esta tabla se crearon los campos para guardar la selección del usuario, aquí únicamente el almacenamiento consume los recursos del lado del servidor, ya que toda la personalización se la realiza desde el lado del cliente al cargar inicialmente la información de cada usuario.

4.1.1. Diseño.

En el diseño se realizaron cambios, puesto que, al ser un prototipo final ya debe mostrar un sitio web mucho más elegante, mejorando la experiencia del usuario. La herramienta para su maquetación es la misma de las anteriores iteraciones.



Se pretendió estandarizar el menú navegacional, para que muestre solo las acciones, retirando componentes que no eran necesarios como es el caso de “Mi Perfil”, que al ser una opción debía estar en el menú de “Opciones”, ya que permite tener el control de todo el sitio web.

El calendario se decidió retirarlo puesto que es un prototipo y no se vio necesario utilizarlo. Todas las otras opciones se siguen manteniendo desde el capítulo anterior. De acuerdo a la figura No. 35 el “Título” ya será el “Listado de Materias”, el “Texto” es el nombre de cada una de las materias. Se creó en el servidor de Moodle materias, las que más adelante serán llamadas de acuerdo al estudiante que se haya matriculado.

4.1.2. Prototipo.

El prototipo se mantiene tal cual, desde la iteración anterior, se incrementó la creación de nuevos usuarios en la plataforma de Moodle, creando cuatro usuarios estudiantes, uno como administrador y un usuario profesor.

Vultr provisionó una dirección con la siguiente IP: 45.77.194.153. En esta dirección se encuentra alojado el servidor de Moodle. Una vez ingresado a la plataforma de Moodle se crearon los usuarios y los cursos. La figura No. 36 muestra el listado de usuarios creados.

Usuarios creados en Moodle	
Nombre / Apellido(s)	Dirección Email
Admin Usuario	michuliga16@gmail.com
Andres Mendoza	amendoza@est.ups.edu.ec
Carlos Andrade	candrade@ups.edu.ec
Lorena Perez	lperez@est.ups.edu.ec
Mishell Rosero	mishellaracely@hotmail.com
Paola Olmedo	polmedov@est.ups.edu.ec

Figura 36. Usuarios creados en Moodle
Elaborado por: Paola Olmedo y Mishell Rosero

La figura No. 37 evidencia las materias creadas para que los usuarios sean matriculados y realizar el inicio de sesión desde el sitio web.

Listado de materias en Moodle



Figura 37. Listado de materias en Moodle
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.1 Inicio de sesión de usuarios de tipo Estudiante.

En el sitio web, la figura No. 38 presenta la página inicial, en la que al usuario le permite iniciar su sesión.

Inicio de sesión de usuario estudiante



Figura 38. Inicio de sesión de usuario estudiante
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.1.1. Código implementado.

La figura No. 39, muestra un fragmento del código usado de la vista, en el inicio de sesión.

Fragmento de código que permite inicio de sesión

```
<meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0; user-scalable=0;">
<div id="login" class="container-fluid main dark fullscreen" style="..." >
  <div class="">
    <h2 id="letrallogin" style="...">Inicio de Sesión </h2>
    <form id="loginform" role="form" class="form-signin">
      <input type="text" style="..." class="login-form_input"
        placeholder="Nombre de usuario" ng-model="general.user" required autofocus>
      <input type="password" style="..." class="login-form_input"
        placeholder="Contraseña" ng-model="general.pass" required autofocus>
      <button class="btn btn-info" ng-click="general.login(general.user, general.pass)" style="..." > Ingresar </button>
    </form>
  </div>
</div>
```

Figura 39. Fragmento de código que permite inicio de sesión
Elaborado por: Paola Olmedo y Mishell Rosero

Se creó un formulario de inicio de sesión, en el que se utilizó un modelo para el ingreso de usuario y contraseña, los campos son obligatorios utilizando el atributo de HTML, se usa la directiva "ng-click" para llamar a la función "login", enviando como parámetro el usuario y la contraseña ingresada como se observó en la figura No. 39.

Función que permite el inicio de sesión

```
login(Nombre Usuario, Clave Usuario) {
  console.log(this.user);
  console.log(this.pass);
  if(Nombre Usuario!="" && Clave Usuario!="") {
    this.findpromise = this._generalFactory.getToken(
      {
        username:Nombre Usuario,
        password:Clave Usuario,
        service: "moodle mobile app",
        wsmoodleformat: "json"
      });
    console.log(this.findpromise);
    this.findpromise.then((data)=>{
      if (data.token !== undefined){
        self.tokenUser = data.token;
        self.user=Nombre Usuario;
        let datosSesion={};
        datosSesion.username =self.user;
        datosSesion.tokenUser =self.tokenUser;
        console.log(datosSesion);
        self.inicioEstudiante(datosSesion);
      } else {
        self.mensajeErrorSesion();
      }
    });
  }
}
```

Figura 40. Función que permite el inicio de sesión
Elaborado por: Paola Olmedo y Mishell Rosero

La función "login" recibe como parámetro el usuario y contraseña que llama al servicio para la comunicación con Moodle, enviando como parámetro el usuario y contraseña ingresados, si la respuesta del servicio de autenticación de Moodle no regresa el token del usuario se muestra el mensaje de usuario y contraseña incorrectos, caso contrario se envía una nueva información al servicio que tiene como parámetro el token de usuario y el nombre, cargándose todos los datos del usuario, llamando a la función "inicioEstudiante" enviando como parámetro los datos del estudiante como se observó en la figura No. 40.

Fragmento de código que trae el identificador del estudiante

```
inicioEstudiante(datosSesion){
  this.$state.go('perfiles', {
    'datosSesion': datosSesion,
  });
}
```

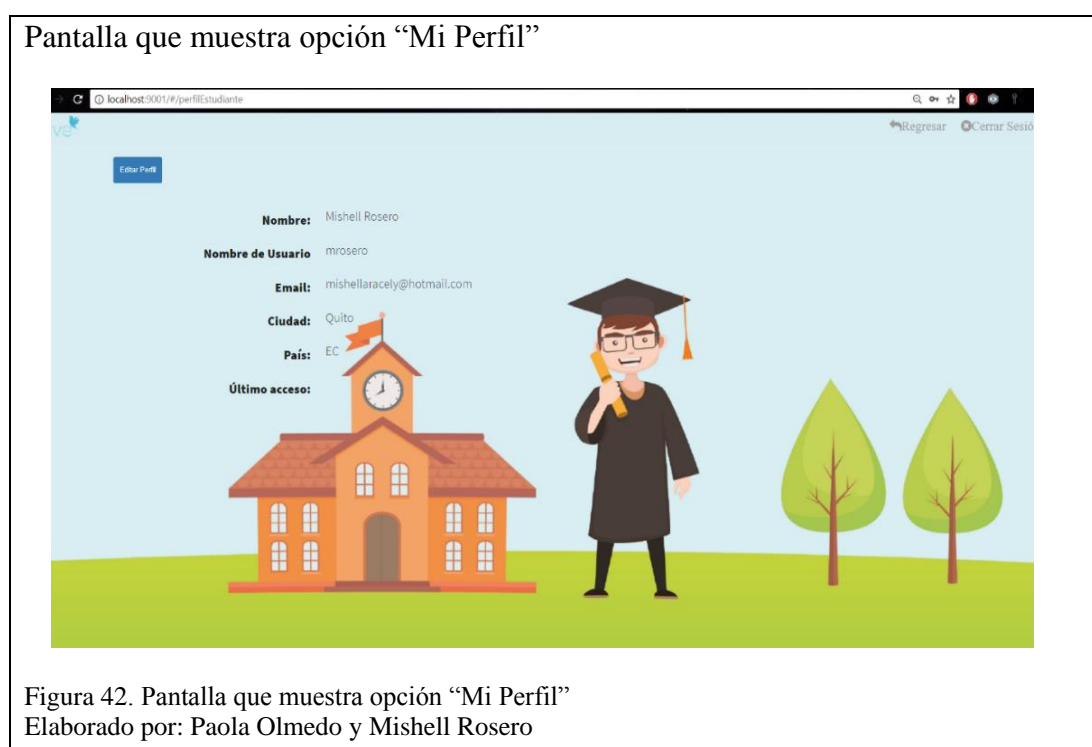
Figura 41. Fragmento de código que trae el identificador del estudiante
Elaborado por: Paola Olmedo y Mishell Rosero

En la función "inicioEstudiante" se reciben los datos de la sesión, se llama al nuevo componente perfiles a través de la directiva "\$state.go()" y se envía como parámetro los datos del usuario como se observó en la figura No. 41.

Se mostrará la pantalla principal para el usuario en donde se tendrá la imagen de perfil, el menú de "Opciones", "Listado de Materias" y el contenido de cada una de ellas.

4.1.2.2 Opción "Mi Perfil".

Se encuentra dentro del menú de "Opciones", al dar click el usuario permitirá cambiar sus datos personales, estos datos se guardarán en el servidor de Moodle. La figura No. 42 muestra el ingreso al Editar el Perfil, y los datos que el usuario puede cambiar.



4.1.2.2.1 Código implementado.

Se creó un formulario para mostrar dentro de etiquetas de texto los datos del usuario cargados desde Moodle como se puede observar en la figura No. 43.

Fragmento de código que llama a la función

```
<div class="row container body1" ng-show="centerPerfilEstudianteCtrl.isEditData==false">
  <form class="form-horizontal " style="..." >
    <fieldset>
      <div class="form-group">
        <label class="col-md-4 control-label" for="name"><b>Nombre:</b></label>
        <div class="col-md-4"...>
      </div>
      <div class="form-group">
        <label class="col-md-4 control-label" for="password"><b>Nombre de Usuario</b></label>
        <div class="col-md-4"...>
      </div>
      <div class="form-group">
        <label class="col-md-4 control-label" for="email"><b>Email:</b></label>
        <div class="col-md-4"...>
      </div>
      <div class="form-group">
        <label class="col-md-4 control-label" for="city"><b>Ciudad:</b></label>
        <div class="col-md-4"...>
      </div>
      <div class="form-group">
        <label class="col-md-4 control-label" for="city"><b>Pais:</b></label>
        <div class="col-md-4"...>
      </div>
    </fieldset>
  </form>
</div>
```

Figura 43. Fragmento de código que llama a la función
Elaborado por: Paola Olmedo y Mishell Rosero

Se puede editar los datos del usuario al dar click en "Editar perfil" y se mostrará el formulario para ingresar los nuevos datos al guardar utilizando la directiva "ng-click" para la llamada a la función "updateUserData()". La figura No. 44 muestra la función antes mencionada.

Función que permite almacenar los cambios realizados

```
updateUserData(){
self.userUpdatePromise=self._perfilEstudianteFactory.editUserProfile(
self.userDataEdit)
.then(data =>{
if(data==null){
self.notificationService.notification({
message : '<p>Registro guardado con
exito.</p>',
layout : 'growl',
effect : 'scale',
type : 'notice',
ctrl:'general',
onClose: function () {
}
});
self.notificationService.show();
self.isEditData=false;
self.userDataEdit=self.userDataEdit;
self.usersModel.userSession.firstname=self.userDataEdit.firstname;
self.usersModel.userSession.email=self.userDataEdit.email;
self.usersModel.userSession.lastname=self.userDataEdit.lastname;
self.$scope.$applyAsync();
};
});
}
```

Figura 44. Función que permite almacenar los realizados
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.3 Ubicación de componentes.

En esta iteración se optimizó el código de las anteriores iteraciones con el cambio de plantilla, de color y de forma de los componentes y la imagen de fondo de pantalla. Se utilizaron nuevas directivas para facilitar la flexibilidad y se agregaron las funciones faltantes de carga de imagen a seleccionar desde el ordenador del usuario y la de creación de plantilla propia del mismo, en la que se realizó una refactorización de código.

En esta iteración final las posiciones para la ubicación de componentes se formó un arreglo compuesto de ceros y unos, para que cada que se lo vaya a abrir se lea y se ubiquen los botones en la posición en la que corresponde, en cambio en la iteración anterior se tuvo siete botones, los que solo se los mostraba o se los ocultaba. Ahora cada que se guarda en la base de datos en el campo que es para la plantilla se guarda tal y como muestra la figura No. 45.

Forma de guardar los componentes

1	0	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---

Figura 45. Forma de guardar los componentes
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.3.1 Código implementado.

Tal y como se inició el drag & drop se sigue manteniendo para el movimiento e incorporación de los botones. A continuación, se muestra el código implementado, al dar click en el botón "Siguiete" permitirá llamar a la función "cambioPosicion", pasando el arreglo armado de ceros y unos, en el cual se usará un bucle para recorrer el arreglo y se va comparando si es cero se deja vacío y si es uno se usa la directiva "appendTo" para mover el componente a esa posición como se observa en la figura No. 46.

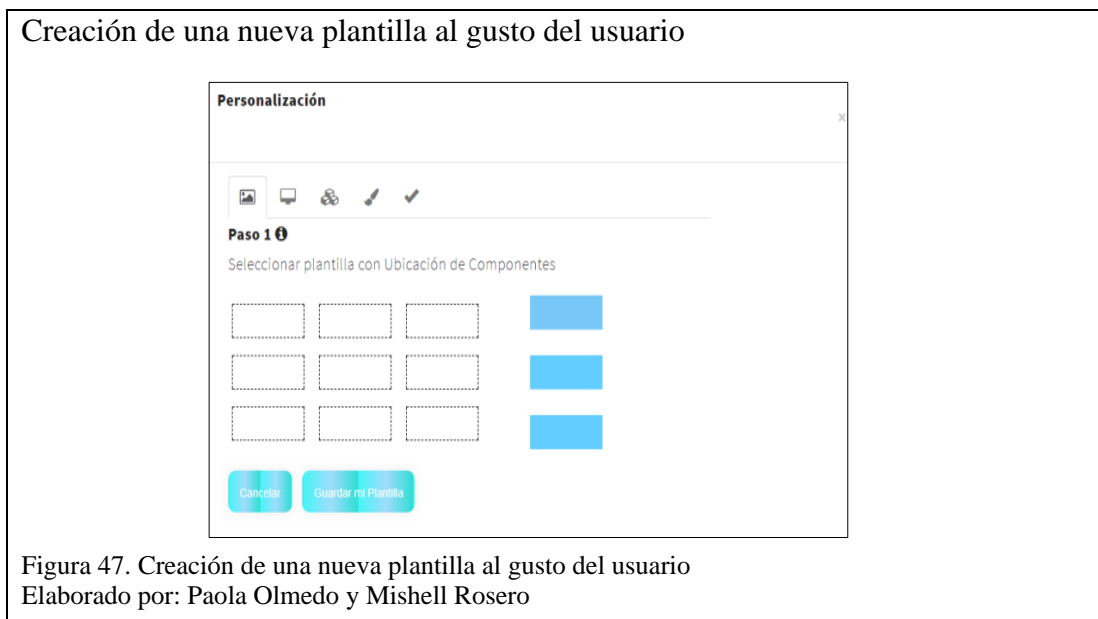
Fragmento de la función llamada desde la vista

```
cambioPosicion(posiciones) {  
  let posicion=1;  
  for (let j=0;j<=8;j++) {  
    if (posiciones[j] == 1) {  
      if (posicion == 1) {  
        $(document.getElementById("menu1")).appendTo(document.getElementById("div" + j));  
        posicion = 2;  
      }else {  
        if (posicion == 2) {  
          $(document.getElementById("menu2")).appendTo(document.getElementById("div" + j));  
          posicion = 3;  
        }else {  
          $(document.getElementById("menu3")).appendTo(document.getElementById("div" + j));  
        }  
      }  
    }  
  }  
}
```

Figura 46. Fragmento de la función llamada desde la vista
Muestra el arreglo en el que se guarda las posiciones
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.4 Creación de nueva plantilla.

Como se explicó anteriormente si el usuario no desea ninguna de las tres plantillas, tiene la opción de crear una nueva a su gusto, manteniendo los parámetros de no alejarse del centro del sitio web. La figura No. 47 muestra la llamada a la creación de una nueva plantilla, desde la ventana emergente. El usuario puede ubicarlos en donde se encuentran cada uno de los contenedores de etiquetas.



4.1.2.4.1 Código implementado.

Se realiza una comparativa que recorre todas las posiciones disponibles de la plantilla, si están o no vacíos con la directiva "innerHTML", si está vacío se colocará un cero, caso contrario uno en el arreglo. La figura No. 48 muestra la función que permite esta ubicación.

Función que permite ubicar los componentes

```
comparar(){
  for (let j=11;j<=19;j++) {
    this.div=document.getElementById("div" +
j);
    if (this.div.innerHTML == "") {
      this.coleccionPlantilla.push("0");
    }else{
      this.coleccionPlantilla.push("1");
    }
  }
  console.log(this.coleccionPlantilla);
  this.cambioPosicion(this.coleccionPlantilla);
  this.plantillaPantalla=this.coleccionPlantilla.toString();
}
```

Figura 48. Función que permite ubicar los componentes

El botón llama a la función “comparar”, la cual está enviando una comparación de los botones para proceder a guardar.

Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.5 Opción “Mis Materias”.

En las iteraciones anteriores se contaba con “Listado de Cursos”, se decidió cambiar por “Mis Materias”, debido a que el proyecto de titulación del que se partió posee el mismo nombre, por lo que se dejó con este. La figura No. 49 muestra las materias en las que el usuario está matriculado, esta información la trae desde el servidor de Moodle.

Presentación de opción “Mis Materias”



Figura 49. Presentación de la opción “Mis Materias”
Elaborado por: Paola Olmedo y Mishell Rosero

De acuerdo a la materia el usuario accederá a cada una de ellas, con la opción de encontrarse con los contenidos que poseen. La figura No. 50 muestra los contenidos de una de las materias, tal y como se ve, los cambios que el usuario ha realizado previamente de personalización se mantienen.

Componentes de una de las materias

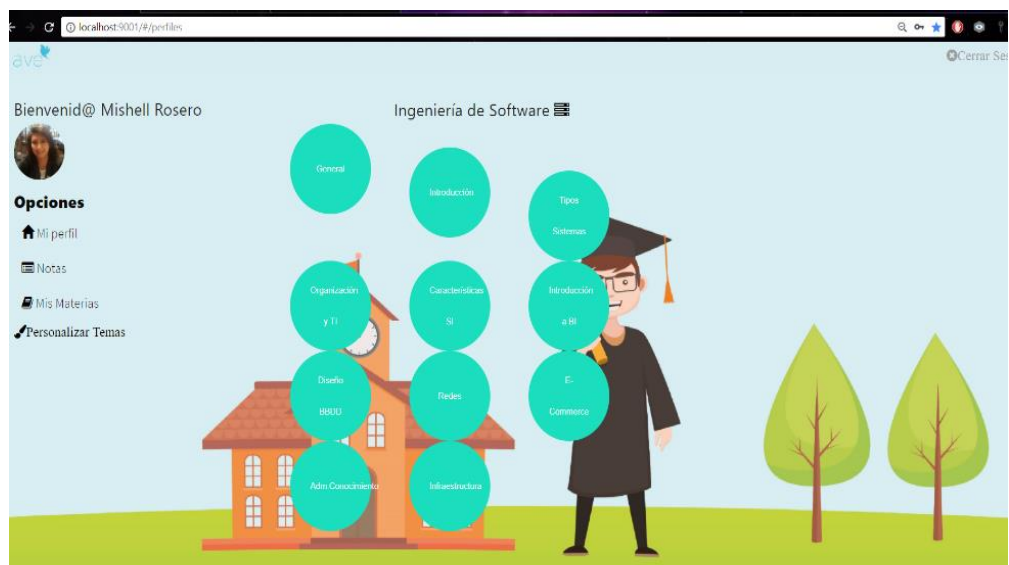


Figura 50. Componentes de una de las materias
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.5.1. Código implementado.

Se cargan los cursos de cada usuario al iniciar la pantalla enviando como parámetro el código de identificación que se recibió de los datos de sesión desde el inicio de sesión al sistema. La figura No. 51 muestra la llamada a este código de identificación.

Fragmento de código del listado de materias

```
getCoursesUser(id) {  
  var coursesPromise=self._perfilesFactory.getUsersCourses({userId:id});  
  coursesPromise.then((dataCourses)=>{  
    if(dataCourses){  
      this.listadoCursosUsuario=dataCourses;  
      this.datosMateria=this.listadoCursosUsuario;  
      console.log(this.listadoCursosUsuario);  
      angular.forEach(this.datosMateria,function(course) {  
        self.contenidoCurso(course.id);  
      });  
    }  
  });  
}
```

Figura 51. Fragmento de código del listado de materias
Elaborado por: Paola Olmedo y Mishell Rosero

La materia se envía dentro de un bucle para recorrer todas las que el usuario se encuentra matriculado, permitiendo cargar el contenido del curso enviando como parámetro el código de identificador del curso y cargando todos los contenidos, así muestra la figura No. 52.

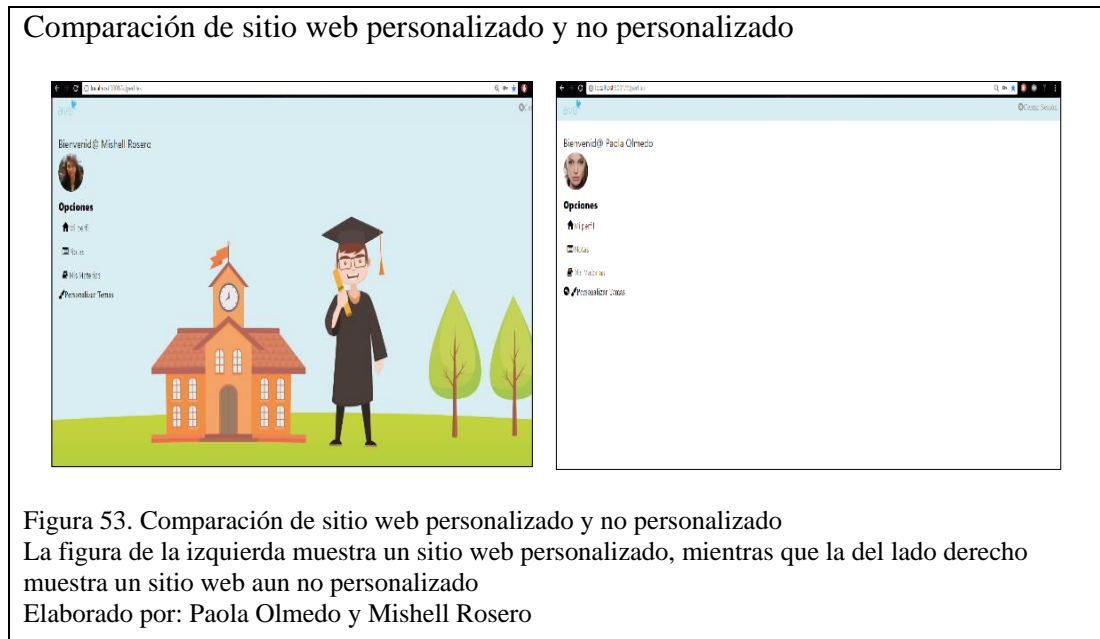
Fragmento de código que muestra el identificador de la materia

```
contenidoCurso(id) {  
  var contentPromise=self._perfilesFactory.getContentsCourse({courseId:id});  
  contentPromise.then((data)=>{  
    if(data){  
      self.contenidosCurso=data;  
      console.log(self.contenidosCurso);  
      this.modulosCurso.push(self.contenidosCurso);  
    }  
  });  
}
```

Figura 52. Fragmento de código que muestra el identificador de la materia
Elaborado por: Paola Olmedo y Mishell Rosero

4.1.2.6 Comparación de pantalla personalizada y no personalizada.

El listado de materias se encuentra en el menú al inicio de sesión del usuario, la figura No. 53 muestra el inicio de sesión de un usuario que ha realizado una personalización y junto a este presenta una imagen de un usuario que aún no la personaliza.



4.1.2.7 Asistente de personalización.

Se incorporó un asistente de personalización si el usuario aun no lo ha hecho, esta opción se presenta junto a la opción “Personalizar Temas” del menú vertical, se tiene un código incorporado que al dar click abrirá una ventana emergente informándole al usuario que deberá personalizar su sitio web. La figura No. 54 muestra la información de la ventana emergente.

Llamada a la ventana de Asistente de personalización

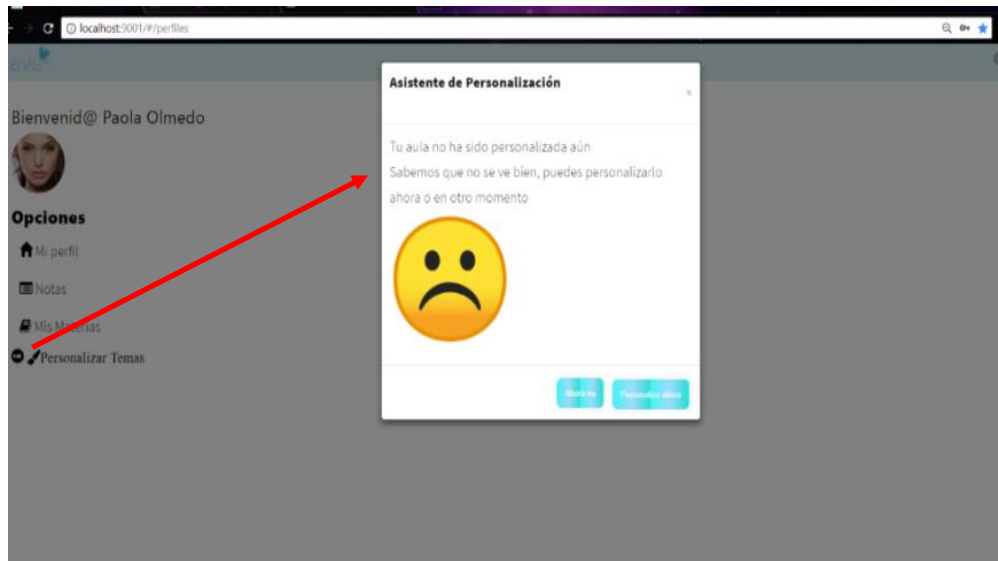


Figura 54. Llamada a la ventana de Asistente de personalización
Elaborado por: Paola Olmedo y Mishell Rosero

Si el usuario está de acuerdo en personalizar dará click en el botón "Personalizar ahora", caso contrario, podrá no hacerlo y realizarlo en otro momento.

4.1.2.8 Cambios guardados.

Así también se creó una ventana emergente, que al finalizar el usuario la personalización de la aplicación se despliega una ventana emergente indicando que sus cambios se han guardado exitosamente. La figura No. 55 muestra esta ventana.

Ventana de cambios realizados

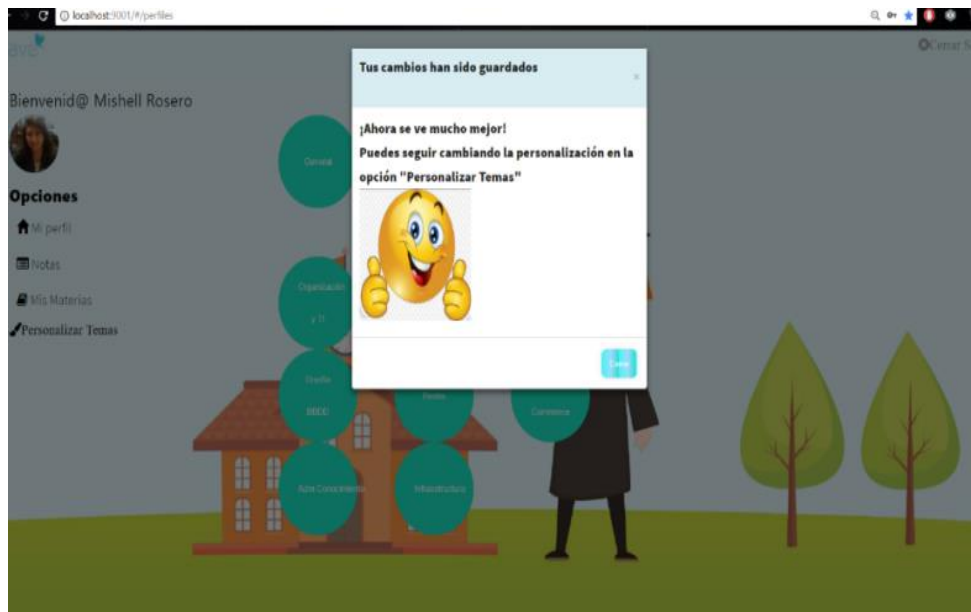


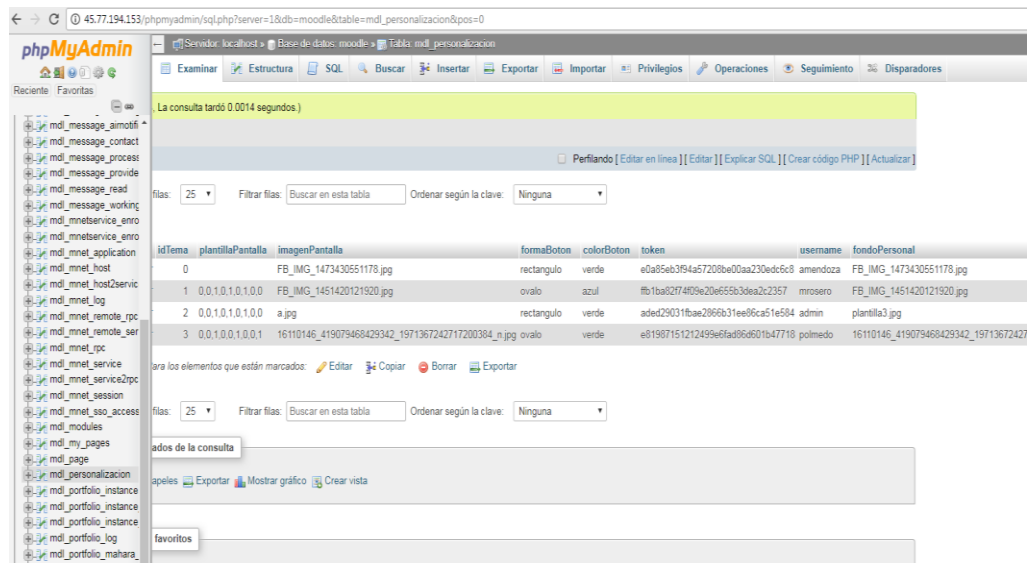
Figura 55. Ventana de cambios realizados
Elaborado por: Paola Olmedo y Mishell Rosero

Estos detalles se incorporaron para que al momento del ingreso del usuario se presente una pantalla más amigable y comprensible en la navegación y también se tomaron en cuenta los conceptos de usabilidad, mencionados en el capítulo uno.

4.1.3. Evaluación.

Se realizó la evaluación final ya con los cambios guardados en la base de datos de Moodle, la figura No. 56 muestra los registros almacenados por el nombre del usuario, la tabla en la que se almacenan los datos es "mdl_personalizacion".

Información de la tabla con los cambios guardados



idTema	plantillaPantalla	imagenPantalla	formaBoton	colorBoton	token	username	fondoPersonal
0		FB_IMG_1473430551178.jpg	rectangulo	verde	e0a85eb3f94a57208be00aa230edc6c8	amendoza	FB_IMG_1473430551178.jpg
1	0.0.1.0.1.0.1.0.0	FB_IMG_1451420121920.jpg	oval	azul	#b1ba2974f09e20e655b30ea2c2357	mrosero	FB_IMG_1451420121920.jpg
2	0.0.1.0.1.0.1.0.0	a.jpg	rectangulo	verde	aded29031bae2866b31ee86ca51e584	admin	plantilla3.jpg
3	0.0.1.0.1.0.0.1	16110146_419079468429342_1971367242717200384_n.jpg	oval	verde	e81987151212499e9fad96d01b47718	polmedo	16110146_419079468429342_19713672427

Figura 56. Información de la tabla con los cambios guardados
Elaborado por: Paola Olmedo y Mishell Rosero

El código del tema sigue siendo el mismo, se genera automáticamente. Para la plantilla, como se explicó anteriormente se está almacenando en un arreglo de ceros y unos, en el cual en los unos se ubicarán los componentes.

Para la forma y color del botón se mantiene la misma idea desde la iteración anterior de almacenar por el nombre. El token es único para cada usuario.

Para el fondo personal se creó un campo adicional en la tabla, permitiendo almacenar el fondo que el usuario seleccione desde su ordenador.

El método “appendTo()” se lo utilizó para el movimiento de botones, encargándose de insertar los elementos HTML al final de los elementos que se han seleccionado.

Para dar por finalizado a esta última iteración se hace referencia al proyecto de titulación mencionado anteriormente, ya que se mejoró su presentación visualmente, sin embargo, la lógica sigue siendo la misma, ya que los tokens se siguen almacenando.

4.1.3.1 Comparación visual de la presentación de las pantallas del proyecto de titulación en la que se basó con el producto final obtenido.

A continuación, se presenta una comparación en el menú principal del proyecto técnico del que se partió, con el que se tiene actualmente. Como se muestra en la figura No. 57 visualmente es diferente la forma de presentar el menú, el usuario puede ver en un menú de forma vertical. Además, el nombre del aula virtual cambió, el nombre anterior era **PAV** y ahora se cuenta con el de **AVE**, tal como se ha ido mencionando la lógica sigue siendo la misma. Esta iteración permitió mejorar visualmente y funcionalmente el proyecto de titulación anterior.

Comparación del menú principal



Figura 36. Menú Principal, inicia con una animación y muestra el menú con las opciones de perfil, notas y cursos.

Elaborado por: Tatiana Parreño y Jonathan Arana.



Figura 57. Comparación del menú principal

La figura de arriba es traída directamente del proyecto anterior, la figura de abajo es la que se muestra en este proyecto

Elaborado por: Paola Olmedo y Mishell Rosero

Capítulo 5

Pruebas

5.1. Plan de Pruebas

Buscando demostrar que el propósito del desarrollo de esta aplicación es lograr los objetivos iniciales y garantizar el funcionamiento correcto del sitio web, se realizaron pruebas que aportaron a encontrar falencias y así incrementar el nivel de calidad.

No se realizaron pruebas de usabilidad, debido a que se pretendió demostrar que es eficiente consumir los recursos del lado del cliente sin consumir ni uno solo del lado del servidor. Tampoco se realizaron pruebas funcionales, debido a que no se tuvo requisitos iniciales.

5.1.1. Pruebas de medición.

A continuación, se tienen las métricas que permitieron medir el consumo de recursos cliente/servidor.

5.1.1.1 Almacenamiento (*Storage*).

Esta métrica permitió guardar el tamaño de los cambios que el usuario realizó. El total de storage usado es: 12.8 Kb. En ambos casos, tanto del lado del cliente, como cliente/servidor, usa el mismo tamaño de capacidad de almacenamiento.

5.1.1.2 Consumo de memoria.

Para calcular el consumo de memoria se midió de acuerdo a los siguientes parámetros:

5.1.1.2.1. *Uso del servidor al momento de ingresar a la aplicación.*

Se realizó la medición al momento de ingresar con el usuario y la clave al sitio web.

Tabla 10. Medición de ingreso al sitio

Recurso	Mínimo	Llega hasta
CPU	0.7%	15.9%
Memoria	352 Mb	990 Mb

Nota: Muestra la medición de ingreso al sitio

5.1.1.2.2 *Uso del servidor al momento de cargar la personalización.*

Dentro del sitio web, el usuario tiene la posibilidad de personalizarlo, por lo que se desarrolló una tabla de medición.

Tabla 11. Medir consumo al cargar la ventana de personalización

Recurso	Mínimo	Llega hasta
CPU	0.7%	1.3%
Memoria	352 Mb	379 Mb

Nota: Tabla que permitió medir el consumo al cargar la ventana de personalización

5.1.1.2.3 *Uso del servidor en consulta de información del estudiante.*

Consultar datos tales como materias, contenidos, datos.

Tabla 12. Tabla 12. Medición de recursos a la carga de datos

Recurso	Mínimo	Llega hasta
CPU	0.7%	12.1%
Memoria	379 Mb	381 Mb

Nota: Tabla que muestra la medición de recursos a la carga de datos

5.1.1.3 *Tiempo de procesamiento.*

Para medir el tiempo de procesamiento se realizó las pruebas al guardar el fondo de pantalla, ya estando en la opción de personalización.

Tabla 13. Medición en el tiempo consumido

Recurso	Mínimo	Llega hasta
CPU	0.7%	30.5%
Memoria	379 Mb	386 Mb

Nota: Tabla que muestra la medición en el tiempo consumido

5.1.2 **Pruebas de simulador al servidor.**

A continuación, se presentan las pruebas que se realizaron consumiendo los recursos del lado del servidor. Estas pruebas simulan el ingreso de diferentes usuarios que cargan una imagen, seleccionan un color, una forma y una plantilla haciéndolo del lado del servidor.

En la aplicación web existe la opción de ingresar la cantidad de usuarios que van a realizar el envío y seleccionar color, forma, plantilla de un listado de opciones y además un botón que permite cargar la imagen y automáticamente medirá el nivel de rendimiento y tiempo de respuesta por parte del lado del servidor, así se puede observar en la figura No. 58.

Pruebas realizadas al servidor



Figura 58. Pruebas realizadas al servidor
Elaborado por: Paola Olmedo y Mishell Rosero

5.1.3 Tabla comparativa entre una aplicación convencional cliente/servidor y una aplicación solo de lado del cliente.

La tabla No. 14, muestra una comparación acerca de una aplicación convencional cliente/servidor frente a una aplicación solo del lado del cliente, siendo:

Escenario 1: Aplicación cliente/servidor

Escenario 2: Aplicación solo del cliente

Esta prueba se la realizó a 5 usuarios que usaron al mismo tiempo la aplicación del lado del cliente, y de acuerdo con el simulador se usaron 5 usuarios que se encuentren en la aplicación cliente/servidor.

Lo primero que se comparó es el tiempo de respuesta empleado por el cliente, tanto en el Escenario 1, como en el Escenario 2.

Después de hizo la comparación del enfoque interno del servidor, del consumo de Memoria RAM Y CPU, tanto en el Escenario 1, como en el Escenario 2.

El servidor para la aplicación convencional cliente/servidor, consta de un procesador de cuatro núcleos activos de 2.70 GHz, y la Memoria RAM es de 8 Gb.

El servidor en el que se encuentra alojado el sitio web de este proyecto técnico consta de un procesador de un núcleo activo de 2.50 GHz y 1 Gb de Memoria RAM.

Estas pruebas se las realizaron en la personalización de los cuatros enfoques, siendo estos: ubicación, cambio de forma, cambio de fondo de componentes y agregar un fondo de pantalla.

Tabla 14. Comparación de consumo de recursos

Criterios	Escenario 1	Escenario 2	Ahorro
Tiempo de respuesta empleado por el usuario	3.7 ms	0.1 ms	97.29%
Memoria RAM del servidor	41%	33%	19.51%
Uso de CPU del servidor	22%	1%	95.45%

Nota: Tabla que muestra el porcentaje de comparación al consumir recursos tanto del lado del cliente

De acuerdo con la tabla No. 14, se evidenció que, el tiempo de respuesta empleado por el usuario al utilizar una aplicación convencional es del 3.7 ms y al usar la aplicación solo del lado del cliente es de 0.1 ms, existiendo una diferencia del 2 % tendiendo, así como conclusión que el ahorro es del 97%.

Conclusiones

- Al investigar e implementar las funcionalidades de HTML5 y JavaScript que permiten brindar al usuario una experiencia de personalización se puede concluir que al realizar este proceso del lado del cliente hay una mejora de alrededor del 70.75% en el consumo de recursos del servidor, con lo cual la eficiencia de la solución es evidente.
- Al finalizar esta aplicación se comprobó que utilizando las tecnologías adecuadas es posible dotar de capacidades de personalización a un sitio web, consumiendo recursos del lado del cliente.
- Cuando se utilizan tecnologías del lado del cliente, el tiempo de respuesta en la carga de la personalización es mínima y cuando se usa tecnologías cliente/servidor, el tiempo se incrementa.

Recomendaciones

- Se recomienda explorar las funcionalidades que Moodle ofrece, para de esta manera en un futuro proyecto poder implementarlas basándose en este proyecto.
- En un futuro se puede demostrar que esta aplicación tiene un impacto netamente en la usabilidad, ya no solo teórico y que, mediante nociones básicas de diseño de sitios web, se pueden implementar diferentes tipos de colores o agregar aspectos visuales para un mayor agrado del usuario sin un significativo consumo del servidor.
- Al realizar un estudio más profundo de Moodle se debe aplicar una personalización más robusta, porque ya no solo se va a poder personalizar el usuario estudiante, sino también el usuario profesor y administrador.
- Para que esta aplicación sea explotada completamente es recomendable que se profundice más en el tema de usabilidad pedagógica y de esta manera demostrar todas las utilidades que las funcionalidades de esta aplicación poseen.

Glosario de Términos

Front End: Parte del desarrollo web, dedicado a la forma de presentación al usuario final, llegando a ser las tecnologías que se ejecutan del lado del cliente.

Token: Se lo llama también componente léxico, siendo una cadena de caracteres con significado congruente, se lo usa como identificador usado por el servidor de notificaciones.

Arquitectura de Información: Se refiere a los sistemas de estructuración y organización de los contenidos, sistemas de rotulado que provea el sitio web.

E-learning: Espacio virtual de aprendizaje inclinado a que la experiencia de la capacitación a distancia sea fácil, tanto en instituciones educativas como en empresas.

Scripts: Conjunto de ordenes almacenadas dentro de un archivo de texto, que suele tener un tamaño ligero, ejecutándose por bloques en tiempo real por un navegador.

LISTA DE REFERENCIAS

- Alejos, L. F. (18 de Diciembre de 2014). Hablemos de Angular. Obtenido de ANGULARJS: <https://frontendlabs.io/2152--hablemos-de-angularjs>
- aulainformativa(s.f). (13 de Septiembre de 2016). Aula Informativa Online Professional Training. Obtenido de Diferencias esenciales entre un wireframe, mockup y prototipo: <http://blog.aulaformativa.com/diferencias-esenciales-wireframe-mockup-prototipo/>
- Basalo, A., & Alvarez, M. (22 de Agosto de 2014). *desarrolloweb.com*. Obtenido de Por qué AngularJS: <https://desarrolloweb.com/articulos/por-que-angularjs.html>
- Benítez, C. P. (2012). *Los mejores diseñadores gráficos del mundo* (1era ed.). Madrid, España: Ilus Books.
- Bustamante, D., & Rodríguez, J. (2014). Metodología Actual Metodología XP. *Revista semana*, 20-23. Obtenido de <http://blogs.unellez.edu.ve/dsilva/files/2014/07/Metodologia-XP.pdf>
- Carol, G. (8 de Julio de 2016). *Diseño Web*. Obtenido de Que es el diseño web: <http://cv.uoc.edu/web/~cguirador/dise%C3%B1oweb/que.html>
- Casale, C. (24 de Marzo de 2015). *Libros blancos*. Obtenido de HTML5 vs. Flash: lo que necesita saber: <https://www.accusoft.com/whitepapers/html5-vs-flash-what-you-need-to-know/>
- ConexiónEsan. (4 de Marzo de 2013). *Web 3.0: diez características que te permitirán identificarla*. Obtenido de Web 3.0: diez características que te permitirán identificarla: <https://www.esan.edu.pe/apuntes-empresariales/2015/05/web-3-diez-caracteristicas-que-te-permitiran-identificarla/>

- Cortizo, J., Expósito, D., & Ruiz, M. (2003). *eXtreme Programming* (Vol. I). AINetSolutions Technical Report.
- Cruz, A. (4 de Enero de 2016). *Desarrollo Libre*. Obtenido de El Drag and Drop en HTML5 (JavaScript): <http://www.desarrollolibre.net/blog/tema/45/html/el-drag-and-drop-en-html5-javascript#.WjiAtFXibiIU>
- Diaz, I. A. (12 de Abril de 2014). *Ser Programadores*. Obtenido de ¿Que es FrontEnd Y Backend en la programación web?: <https://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>
- Diez, J. (26 de Agosto de 2016). *Avante Services*. Obtenido de Avante Services: <http://www.avanteservices.com/es/blog/?p=1265>
- ethertank, i. (6 de Mayo de 2015). *Developer.mozilla.org*. Obtenido de Arrastrar y soltar: https://developer.mozilla.org/es/docs/DragDrop/Drag_and_Drop
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascrip*. Caracas: Marcombo.
- Hassan, Y., Martín, F. J., & Iazza, G. (2004). Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. *eprints.org*, 13-17. Obtenido de http://eprints.rclis.org/8998/1/Dise%C3%B1o_Web_Centrado_en_el_Usuario_Usabilidad_y_Arquitectura_de_la_Informaci%C3%B3n.pdf
- Hurtado, F., Olmedo, P., & Rosero, M. (11 de Enero de 2017). *Blog personal de Franklin Hurtado*. Obtenido de LMS y LCMS: <http://franklinhurtado.info/2017/01/11/lms-y-lcms/>
- IBM Knowledge Center. (4 de Julio de 2014). *Modelo de cliente/servidor*. Obtenido de Modelo de cliente/servidor:

https://www.ibm.com/support/knowledgecenter/es/SSAL2T_8.2.0/com.ibm.cics.tx.doc/concepts/c_clnt_sevr_model.html

Joskowicz, J. (2008). *Reglas y prácticas en eXtreme Programming*. Universidad de Vigo, España. Obtenido de <https://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>

JustInMind. (12 de Noviembre de 2015). *JUSTINMIND*. Obtenido de JUSTINMIND: <https://www.justinmind.com/>

Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. San Vicente, Alicante: Editorial Club Universitario.

Martínez, C. (2013). ESTUDIO DEL ESTADO DEL ARTE SOBRE FRAMEWORKS MVC PARA EL DESARROLLO DE APLICACIONES WEB CLIENTE, CASO DE ESTUDIO EMBER.JS. *Ciencia y tecnología*, 78.

MDN web docs. (6 de Diciembre de 2017). *Introducción al lado del servidor*. Obtenido de *Introducción al lado del servidor*: https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction

Moodle. (8 de Marzo de 2014). *Moodle*. Obtenido de MariaDB: <https://docs.moodle.org/all/es/MariaDB>

Narváez, E. D. (2012). *Desarrollo de una aplicación para dispositivos móviles con sistema operativo android aplicando extreme programming*. Universidad de las Américas, Facultad de Ingenierías y Ciencias Agropecuarias, Quito.

Navarro, A., Fernández, J., & Morales, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *Prospectiva*, 11(2), 30-39. Recuperado el 4 de Junio de 2013

Nodejs. (2018). *NodeJS*. Obtenido de <https://nodejs.org/es/>

NPM. (2018). *NPM*. Obtenido de <https://www.npmjs.com/>

OkHusting. (5 de Febrero de 2012). *Ok Husting*. Obtenido de Metodologías del Desarrollo de Software: <https://okhosting.com/blog/metodologias-del-desarrollo-de-software/>

Parreño, T., & Arana, J. (2016). *Creación de un front end usando HTML5, para explotar las funcionalidades de la plataforma Moodle, además de una aplicación movil android para el manejo oportuno de notificaciones con sincronización a Google Calendar*. Tesis de Pregrado, Universidad Politécnica Salesiana Sede Quito, Ingeniería en Sistemas, Quito.

Peñalvo, F. J. (2005). Estado actual de los sistemas e-learning. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 6(2).

Pizarro, N. (7 de Febrero de 2011). *Estrategia digital*. Obtenido de Diferencias entre una aplicación web y un sitio web: <https://www.ida.cl/blog/estrategia-digital/diferencias-aplicacion-web-sitio-web/>

Robles, G., & Ferrer, J. (2002). *Programación eXtrema y Software Libre*. Universidad Politécnica de Madrid, Madrid.

Ruíz Lizama, E. (2014). Lenguajes de programación: conceptos y paradigmas. *Industrial Data*. Portal de las revistas de investigación de la Universidad Nacional Mayor de San Marcos, Lima, Perú.

- Sanvicente, D. S. (2017). *Desarrollo del portal web de gestión de licencias de RedBorder usando React y Node JS*. Tesis de Grado, Universidad de Sevilla, Departamento de Ingeniería Telemática, Sevilla.
- Suez, G. (2 de Julio de 2013). *Joomla! Community Magazine*TM. Obtenido de Estructura y Composición de un Sitio Web: <https://magazine.joomla.org/es/ediciones-anteriores/julio-2013/item/1398-estructura-y-composicion-de-un-sitio-web>
- Techlandia. (21 de Junio de 2014). *Diferencia entre el lado del cliente y programación del lado del servidor*. Obtenido de Diferencia entre el lado del cliente y programación del lado del servidor: https://techlandia.com/diferencia-lado-del-cliente-programacion-del-lado-del-servidor-sobre_151711/
- Tovar-Cardozo, D. A., Pimienta-Acero, L. M., & Ramírez-Saavedra, E. R. (2015). Internet: más allá de la superficie. *Revista científica*, 40- 42. doi:10.14483/udistrital.jour.RC.2015.21.a08
- Valbuena, A. (2014). *Guía comparativa de Frameworks para los lenguajes HTML 5, CSS y JavaScript para el desarrollo de aplicaciones Web*. Tesis Pregrado, Universidad Tecnológica de Pereira, Facultad de Ingeniería de Sistemas y Computación, Pereira. Obtenido de <http://hdl.handle.net/11059/4577>
- Vecino, I. (12 de Enero de 2017). *Revista PYMES*. Obtenido de Tendencias en desarrollo y diseño web para 2017: <https://revistapymes.es/tendencias-desarrollo-diseno-web-2017/>
- Vera, M. (2014). *Intelligence to Business*. Obtenido de Los beneficios de Implementar la Metodología Ágil: <http://www.i2btech.com/blog-i2b/tech-deployment/los-beneficios-de-implementar-la-metodologia-agil/>