



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título:

Personalización de la capa de presentación del sistema de seguridad Acaxia para su correcta visualización en dispositivos móviles.

Autores: Fredy Ulloa Morffi

Reinel Leonard García

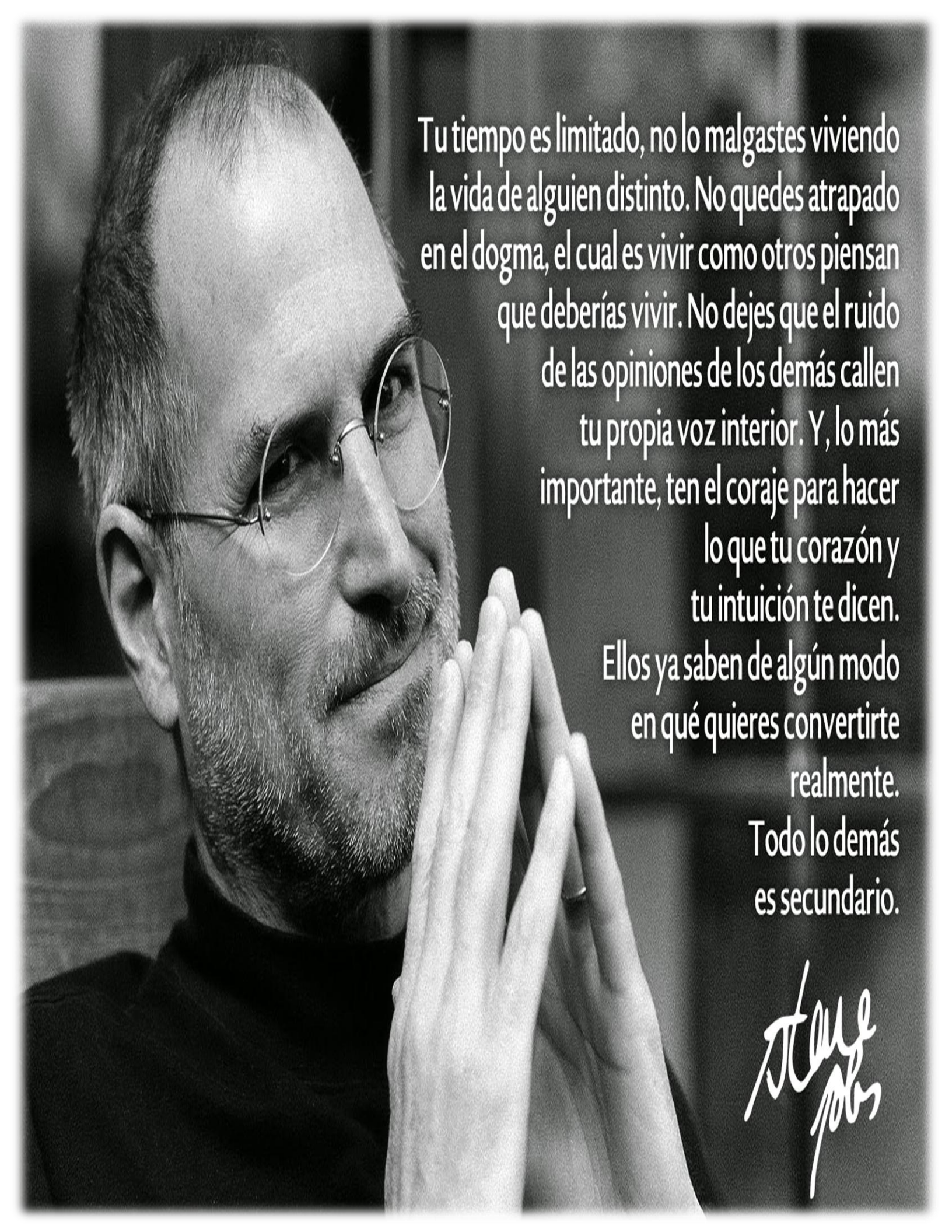
Tutores: Ing. Yoenry Vanega Hechavarría

Ing. Olga Yarisbel Rojas Grass

CIUDAD DE LA HABANA, 2015.

AÑO 57 DE LA REVOLUCIÓN.



A black and white portrait of Steve Jobs, looking slightly to the right with a thoughtful expression. His hands are clasped together in front of his chest. He is wearing his signature round glasses and a dark turtleneck. The background is blurred.

Tu tiempo es limitado, no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, el cual es vivir como otros piensan que deberías vivir. No dejes que el ruido de las opiniones de los demás callen tu propia voz interior. Y, lo más importante, ten el coraje para hacer lo que tu corazón y tu intuición te dicen. Ellos ya saben de algún modo en qué quieres convertirte realmente. Todo lo demás es secundario.

Steve Jobs

Declaramos que somos los autores de este trabajo y autorizamos a la Facultad 3 de la Universidad de las Ciencias Informáticas; así como al Centro de Informatización de Entidades (CEIGE) que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ___ días del mes _____ del año _____.

Fredy Ulloa Morffi

Firma de Autor

Reinel Leonard García

Firma de Autor

Ing. Olga Yarisbel Rojas Grass

Firma de Tutor

Ing. Yoenry Vanega Hechavarría

Firma de Tutor

Síntesis de tutor:

Ing. Olga Yarisbel Rojas Grass.

Ingeniera en Ciencias Informáticas, UCI 2009.

Email: yarisbel@uci.cu

Síntesis del co-tutor:

Ing. Yoenry Vanega Hechavarría.

Ingeniero en Ciencias Informáticas, UCI 2007.

Email: yvanega@uci.cu

Reinel Leonard García

La redacción de este acápite no me será más fácil que la escritura de la tesis, ya que muchas han sido las personas que han contribuido a que mi éxito en la carrera y en la vida se materialice.

- *En primer lugar agradecerle a mi familia, quienes me han brindado un apoyo incondicional a lo largo de estos años.*
- *A Fredy, mi compañero de labores este último año, a él las gracias por aguantarme.*
- *A los nuevos amigos que he tenido la oportunidad o la dicha de conocer y que siempre los guardaré en un lugar especial de mis recuerdos, en especial los que desde el principio de nuestro andar por estas aguas me han brindado su ayuda en cualquier situación. Estos amigos son Alejandro Miguel Saborit, Jose Javier Alemán, Roberto Alfonso Rivero, Pablo Antonio Moreno, Rigoberto Peña,*
- *No puedo dejar de mencionar y agradecer a los demás compañeros de grupo, no por gusto nos autoproclamamos la dinastía o el imperio.*
- *Por supuesto al personal docente que tuvo la tarea de formarme como ingeniero, ahora soy el fruto de lo que ellos también sembraron.*
- *A Elaidys Salcedo por ser tan paciente con nosotros y ayudarnos como nadie lo ha hecho.*
- *A mis tutores por la ayuda y paciencia.*
- *A nuestra consultante y amiga Rosalina Ibarra si es que puedo decirle así.*

Dejé para último los agradecimientos más importantes.

- *A mis hermanos de mil batallas Eddy Sosa y Camilo Martínez que siempre han estado ahí para mí.*
- *A mi abuela materna mi picotillo que está cargando conmigo como una madre y un padre desde que tengo uso de razón.*
- *A Javier mi padrastro que ha sido como un padre para mí desde que llegó a mi mundo.*
- *A mi madrina linda que me supo dar los consejos más sabios y más inteligentes que jamás he escuchado.*
- *A mi padrino que ambos son más que padrinos para mí*
- *Agradezco a mi esposa por ser tan incondicional y ayudarme tanto incluso teniendo la ardua tarea de criar a nuestro hermoso hijo.*
- *A mi papá que como siempre desde mis primeros pasos en la vida, está aquí a mi lado apoyándome y guiándome por el buen camino.*

*Y por último pero no menos importante quiero agradecer a **MI MAMÁ**, esa que es mi todo y que me lo ha dado todo, esa que es mi razón de ser, mi aliento, mi consuelo. Siempre agradeceré a Dios por ser su hijo.*

Gracias a todos.

Fredy Ulloa Morffi

Quiero agradecer primeramente la colaboración y el apoyo de los miembros del tribunal y los tutores, a estos últimos por ser tan comprensivos y pacientes.

A todos los profesores que han hecho posible mi éxito durante mi paso por la universidad.

A mis compañeros de cada una de las aulas por donde pasé en el transcurso de mi carrera y a los chuchumecos por todos los momentos de risas, fiestas y estudios que pasamos juntos.

Quiero agradecer especialmente a mi compañero de tesis Reinel y a mi profesora y amiga Rosalina por sus grandes contribuciones en el desarrollo de este trabajo.

Por último quiero expresar mi mayor agradecimiento a las personas que han hecho de mí alguien importante para sus vidas:

A los amigos del barrio por estar siempre presentes.

A mi hermano, mi suegrita, en general, a mi familia y la de mi novia por dedicarme todo su amor y su cariño.

A mi mamá por realizar la difícil tarea de encaminarme en la vida, brindándome su apoyo, ayuda y confianza en cada paso, por creer en mí, por hacerme sentir alguien realizado cada vez que cumplía una meta en mis estudios, por aguantar mis malcriadeces y comprenderme a pesar de que no siempre tenía la razón, a esa gran mujer que me lo dio todo, la amo.

A mi papá por ser mi guía, mi punto de partida, mi motor impulsor, mi ejemplo a seguir; porque a pesar de estar luchando contra su enfermedad y no poder estar presente en los últimos años de mi carrera, siempre confió en que yo podría superarme y llegar a ser ingeniero, siempre estuvo muy orgulloso de mí, a ese gran hombre lo recordaré toda la vida.

A mi novia “Elaidys” que lleva dentro la ilusión de mi vida, “mi bebecito”, por amarme tanto y tener presente que ellos son las personas que hoy ocupan cada uno de mis pensamientos, que dan sentido a mi vida, que me ayudan a tomar cada una de mis decisiones y que iluminan mi camino hacia un futuro brillante, que están siempre a mi lado cuando más los necesito y que más amo en esta vida.

Gracias a todos.

Reinel Leonard García

Le dedico mi trabajo, mi carrera y todos mis logros hasta el momento a los soles que iluminan mi vida. Mi hermana Rocio, mi hermano Bryan, mi hijo Reinier Alejandro, y a ese que desde el cielo me cuida, mi hermano Reinier.

Freddy Ulloa Morffi

Dedico este Trabajo de Diploma, mi carrera y mi formación profesional a mi mamá porque sin ella no hubiera logrado esta meta, a mi novia porque la amo, a mi hijo que está por nacer porque todo lo que haga lo voy hacer por él y para él, a mi papá que hoy no está con nosotros, pero donde quiera que esté, estoy seguro que está muy orgulloso de mi.

Resumen

Debido al desarrollo progresivo de la tecnología móvil, la información personal y las herramientas de trabajo deben ser adaptadas para ser consultadas desde los dispositivos móviles. Dicha necesidad está relacionada también con los sistemas de gestión y los marcos de trabajo. Por lo que en función de tratar tal situación, muchos autores han publicado diferentes conceptos sobre una nueva forma de diseñar y personalizar el contenido conocida como Diseño Web Responsivo. Por ello, esta investigación está enfocada a explicar el proceso de personalización de la capa de presentación del sistema de seguridad Acaxia, para proveer una visualización correcta en los dispositivos móviles.

Para cumplir este objetivo se estudió todo lo referente al diseño multidispositivo, haciendo énfasis en el Diseño Web Responsivo para dar respuesta a la propuesta de solución y tener los resultados requeridos en la investigación, este diseño permite que las páginas web se ajusten automáticamente a la pantalla de cualquier dispositivo y optimiza gran cantidad de recursos en la realización de sistemas Web. Además en este documento se analizan las características generales que deben cumplir las aplicaciones web para su correcta visualización en dispositivos móviles, se describen las tecnologías y herramientas utilizadas para el desarrollo del sistema propuesto y se presentan los prototipos y recursos necesarios para el diseño de la solución. Se implementa la propuesta de solución y como resultados fundamentales se constató que la personalización de la capa de presentación del sistema de seguridad Acaxia cumple con las pautas y buenas prácticas que definen que un sistema es responsivo. Finalmente se detallan elementos sobre la implementación de la propuesta y se validan los resultados obtenidos en el Trabajo de Diploma.

Palabras claves: Capa de presentación, diseño multidispositivo, Diseño Web Responsivo, dispositivos móviles, personalización.

Abstract

Due to the increasingly improvements of mobile technologies, one's information and work tools must be adapted to be accessed using mobile devices. That necessity is also related to management systems and frameworks. In order to handle such predicament, several authors have published different concepts on a new way of designing and customizing content known as Responsive Web Design. Therefore, this research is aimed to explain the process of customization of the Acaxia Security system layout, to provide a proper view load on mobile devices.

To accomplish such goal, a research about multi-devices design was made highlighting the Responsive Web Design to conceive a solution proposal and seek for good results. This type of design allows web pages to automatically adjust to any device screen size and optimizes a great deal of resources in web development. Besides what was previously said, on this paper are described the technologies and tools used to develop the proposal, as well as the prototypes and other critical resources for design.

The solution implementation comprehends the best practices and requirements of the Responsive Web Design and for the general validation of the solution several techniques are applied.

Keywords: Customization, Layout, Mobile technology, Multi-device design, Responsive Web Design.

Índice

Resumen.....	6
Abstract.....	7
Índice	8
Introducción.....	11
Capítulo 1: Fundamentación teórica.....	16
1.1 Introducción	16
1.2 Conceptos relacionados	16
1.3 Tecnologías para el diseño multidispositivo	17
1.3.1 Diseño Web Responsivo	17
1.3.2 Diseño web Adaptivo.....	18
1.3.3 Comparación de marcos de trabajo responsivos	18
1.4 Metodología de Diseño Web Responsivo	20
1.4.1 Técnica Mobile First	22
1.5 Lenguajes de Programación	23
1.6 Técnicas de prototipado.....	26
1.7 Herramienta de prototipado	26
1.8 IDE de desarrollo	26
1.9 Navegador	27
1.10 Servidor web	27
1.11 Marcos de trabajo	27
1.12 Conclusiones.....	28

Capítulo 2: Propuesta de solución	29
2.1 Introducción	29
2.2 Artefactos del portafolio de proyecto	29
2.3 Propuesta de diseño del sistema	35
2.3.1 Pautas de diseño para sitios web responsivos	35
2.3.2 Propuesta de diseño visual	37
2.4 Patrones de diseño e implementación	46
2.5 Conclusiones	47
Capítulo 3: Implementación y pruebas	48
3.1 Introducción	48
3.2 Técnica de codificación.....	48
3.3 Elementos a tener en cuenta en la implementación	48
3.3.1 Media Queries	48
3.3.2 Ancho de los div	49
3.3.3 Elementos de ancho variable	51
3.4 Estándares de codificación	51
3.5 Detalles de la implementación de la solución.....	55
3.5.1 Componentes visuales	55
3.6 Pruebas y validaciones	59
3.6.1 Técnica Guía de Observación	59
3.6.2 Resultado de la técnica Guía de Observación	62
3.6.3 Técnica de IADOV	62

3.6.4	Pruebas de usabilidad	65
3.6.5	Resultado de las pruebas de usabilidad	66
3.6.6	Pruebas con teléfonos móviles y emuladores	67
3.6.7	Resultado de pruebas con teléfonos móviles y emuladores.....	67
3.6.8	Acta de validación	67
3.7	Conclusiones	69
	Conclusiones generales	70
	Recomendaciones.....	71
	Referencias bibliográficas.....	72
	Análisis bibliométrico	76
	Glosario de términos	77
	Anexos	80

Introducción

En los últimos años se ha apreciado un creciente desarrollo de la tecnología móvil a nivel mundial, debido a que existe una tendencia a que todo gire a través de los servicios que estos han logrado incorporar en sus aplicaciones. El móvil es un instrumento receptor de valiosas cantidades de información y proporciona una multitud de herramientas para estar en contacto los unos con los otros. Ha sido también el responsable de cambios en los hábitos personales y profesionales, convirtiéndose en un factor indispensable en el desarrollo tecnológico.

El acceso por medio de la tecnología móvil a los sistemas de gestión de información, cuales quiera que estos sean, establece una nueva relación usuario-sistema, facilitando la gestión de la información y aproximándola al usuario, lo que permite una interacción continua, eficaz y rápida. Ante este panorama las empresas cubanas han decidido tener en cuenta la tecnología móvil para lograr mayor competitividad. Una parte esencial de la infraestructura necesaria es la creación de sistemas de gestión adaptables a dispositivos móviles.

La Universidad de las Ciencias Informáticas (UCI) es un centro de estudios universitarios cubano, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación. La UCI se encuentra hoy a la vanguardia en el desarrollo de aplicaciones web, incluyendo recientemente aquellas desarrolladas con tecnología que soporten diversidad de dispositivos móviles.

En la universidad, varios centros de desarrollo utilizan el marco de trabajo Sauxe desarrollado en el Centro de Informatización de Entidades (CEIGE) para la construcción de aplicaciones web de gestión. Sauxe es una plataforma que gestiona artefactos o módulos de software específicos, que permite organizar y desarrollar otros proyectos de software. Contiene además, un lenguaje interpretado (PHP) para desarrollar y unir los diferentes componentes de un proyecto, facilita la portabilidad entre arquitecturas e incluye además un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.

Uno de los componentes fundamentales del marco de trabajo Sauxe es Acaxia, un Sistema de Gestión Integral de Seguridad que brinda servicios de autenticación, autorización, auditoría, administración de perfiles y la administración de conexiones a todos los sistemas informáticos que se suscriban a él. Estos servicios solo pueden ser accedidos desde una computadora de escritorio

o una laptop, equipos complejos de ser trasladados o abiertos en muchos ámbitos sociales y que no siempre están disponibles en cualquier lugar donde se tenga acceso a una red inalámbrica.

El sistema Acaxia está implementado sobre el marco de trabajo Sauxe, el cual utiliza ExtJs en su versión 2.4 para la capa de presentación, dicha tecnología no cubre el diseño de aplicaciones destinadas a las pantallas pequeñas y la restringida navegabilidad de los dispositivos móviles. Además, al acceder al actual sistema desde estas tecnologías algunos de los componentes que presenta se distorsionan o cambian de lugar, por otra parte, el formato original del texto no es legible, y el movimiento horizontal y vertical dificulta la navegación por el mismo. Sin embargo, por el alto grado de aceptación que tienen los usuarios hacia las tecnologías móviles, es necesario acceder a los servicios que brinda Acaxia desde cualquier dispositivo. Esto generaría un valor agregado al sistema incrementando el interés en los desarrolladores que utilicen Sauxe, por la facilidad, disponibilidad y productividad que esto ofrece a la hora de acceder a la información.

A partir del análisis antes realizado se plantea el siguiente **problema a resolver**: ¿Cómo personalizar el sistema de seguridad Acaxia para obtener una correcta visualización en dispositivos móviles? Para dar respuesta a dicha interrogante se determina como **objeto de estudio**: el diseño web adaptable y responsivo. Tomando como **campo de acción**: el diseño web adaptable y responsivo para el sistema de seguridad Acaxia.

Analizado el problema, se ha trazado como **objetivo general**: personalizar la capa de presentación del sistema de seguridad Acaxia empleando tecnologías con soporte para los diferentes dispositivos para obtener una correcta visualización en dispositivos móviles. Se tendrán en cuenta objetivos específicos, cuyo cumplimiento, sinérgicamente contribuirá con la materialización del objetivo general.

Objetivos específicos:

1. Confeccionar el marco teórico de la investigación a partir de la búsqueda y revisión bibliográfica sobre las tecnologías web con soporte para los dispositivos móviles.
2. Definir el diseño de presentación de la información contenida en la capa de presentación del sistema de seguridad Acaxia.
3. Realizar el diseño de la propuesta de solución para la personalización de la capa de presentación del sistema de seguridad Acaxia empleando tecnologías con soporte para los dispositivos móviles.

4. Realizar la implementación de la propuesta de solución para la capa de presentación del sistema de seguridad Acaxia empleando tecnologías con soporte para los dispositivos móviles.
5. Validar la implementación de la solución mediante pruebas de usabilidad y pruebas con emuladores y dispositivos móviles.
6. Validar la investigación a través de una guía de observación, la técnica de ladov y un acta de validación.

Se plantea como **idea a defender**: si se personaliza la capa de presentación del sistema de seguridad Acaxia empleando tecnologías con soporte para los dispositivos móviles, se garantizará la correcta visualización del mismo en estos dispositivos.

Para cumplir los objetivos propuestos se definieron las siguientes **tareas de investigación**:

1. Estudio de las tecnologías web con soporte para dispositivos móviles.
2. Realización del marco teórico de la investigación.
3. Confección de la propuesta de solución.
4. Personalización de las funcionalidades en términos de componentes.
5. Validación de los resultados obtenidos.

Métodos científicos

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones (1). Para el desarrollo de la investigación se aplicaron los métodos teóricos y empíricos.

- **Métodos teóricos**

Permiten estudiar las características del objeto de investigación que no son observables directamente y posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado (1). De estos se utilizó:

Histórico – Lógico:

Permitió organizar cronológicamente los antecedentes histórico-lógicos sobre el objeto de estudio.

Analítico – Sintético: permitió la síntesis de la información obtenida en el estudio de tecnologías web con soporte para dispositivos móviles, facilitando su caracterización y descripción.

- **Métodos empíricos**

Los métodos empíricos son los llamados métodos de recolección de información en un proceso investigativo a través de la observación, la entrevista, el cuestionario y el análisis documental. De estos se utilizaron:

Análisis documental: la investigación documental utiliza la técnica bibliográfica o documental para recopilar información a través de diversos instrumentos: revistas, periódicos, internet, entre otros. Acorde a esto se realizó un análisis de todos los documentos que contienen el portafolio de proyecto acerca del Sistema de seguridad Acaxia y se efectuó una revisión de algunas investigaciones hechas sobre el tema.

Observación científica: se realizó con el objetivo de observar el comportamiento del sistema de seguridad Acaxia en diferentes dispositivos móviles.

Encuesta: se realizaron encuestas con el objetivo de evaluar la satisfacción de los usuarios hacia la personalización de la capa de presentación del sistema de seguridad Acaxia, así como para validar que la solución que se propone cumple con el objetivo general de la investigación.

Entrevista: Se realizaron entrevistas al cliente para determinar los requisitos funcionales y requisitos no funcionales a tener en cuenta para la realización de la personalización de la capa de presentación del sistema de seguridad Acaxia.

Para lograr una mejor organización y documentación del presente trabajo, se estructura en tres capítulos:

Capítulo 1: Fundamentación Teórica

En este capítulo se llevará a cabo el estudio del estado del arte realizándose un análisis del Diseño Web Responsivo y el adaptativo, así como de sistemas que utilizan un diseño u otro. Además se describen las tecnologías, herramientas y metodología que se utilizarán en el desarrollo de la solución.

Capítulo 2: Propuesta de solución

Se definirán los requisitos funcionales y no funcionales teniendo en cuenta los descritos en la documentación del análisis del sistema de seguridad Acaxia. Además se describirá la propuesta

de diseño visual de la personalización de la capa de presentación del sistema de seguridad Acaxia. Luego se realizará el diseño de la capa de presentación de Acaxia para su correcta visualización en dispositivos móviles y se reflejará el prototipado propuesto de dichas interfaces.

Capítulo 3: Implementación y pruebas

Se realizará la implementación de la solución propuesta en el capítulo anterior, reflejando brevemente los detalles de la misma. Finalmente se concluye la investigación con la verificación y validación de la solución propuesta a través de diferentes tipos de pruebas, así como el análisis de los resultados obtenidos mediante las validaciones.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En el presente capítulo se abordarán elementos teóricos relacionados con el diseño web adaptable y responsivo, así como conceptos que faciliten el entendimiento del presente trabajo. Como parte del estado del arte se lleva a cabo un estudio del diseño web en las modalidades antes mencionadas, sistemas que usan un diseño u otro y los marcos de trabajo empleados para el desarrollo del diseño responsivo. Además se definirán las tecnologías, herramientas y lenguajes que se utilizarán en el desarrollo de la personalización de la capa de presentación del sistema de seguridad Acaxia.

1.2 Conceptos relacionados

Dispositivos móviles: Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales (2).

Teléfono móvil: también conocido como *celular*, es aquel que carece de cables y que se puede trasladar sin que se registren inconvenientes en la comunicación. El funcionamiento de este tipo de teléfono está dado por ondas de radio que le permiten acceder a las antenas que conforman la red de la telefonía móvil (2).

Telefonía móvil: también llamada *telefonía celular*, básicamente está formada por dos grandes partes: una red de comunicaciones (o red de telefonía móvil) y los terminales (o teléfonos móviles) que permiten el acceso a dicha red (3).

Tecnología móvil: medio de comunicación que ha superado a la telefonía fija, esto se debe a que las redes de telefonía móvil son más fáciles y baratas de desplegar. El uso de las tecnologías móviles entre los habitantes de una población, ayuda a disminuir la brecha digital existente entre cada lugar, ya que muchos usuarios utilizan este medio tecnológico para el desarrollo de sus actividades sociales y económicas (4).

Personalización: adaptar una aplicación o programa a las necesidades y gustos del usuario (5).

Capa de presentación: esta capa, está formada por los formularios y los controles que se encuentran en los formularios. Es la capa con la que interactúa el usuario (6).

1.3 Tecnologías para el diseño multidispositivo

En la actualidad, para el desarrollo de una nueva web o para reformar una existente, hay que tener en cuenta los conceptos de diseño web adaptativo y del responsivo con el objetivo de lograr que la web funcione y se vea correctamente en cualquier dispositivo, para facilitar la navegación e interacción del usuario. A continuación se describen ambos conceptos estableciendo las diferencias existentes entre los mismos.

1.3.1 Diseño Web Responsivo

El Diseño Web Responsivo (RWD¹) se encarga de cambiar el aspecto de la interfaz automáticamente para adaptarse al dispositivo que el usuario está utilizando. Este permite de igual modo que las páginas de la web se ajusten automáticamente al sistema operativo y al diseño de cualquier navegador. En principio se mostrará el mismo contenido de la web sea cual sea el dispositivo, pero la distribución del mismo variará en función del tamaño de la ventana (7).

Ventajas

- Solo se utiliza una web única, la misma URL, es decir, se muestra el mismo contenido con la misma URL independientemente del dispositivo. Si se crea una web móvil, solo para móviles se requiere otras urls.
- La página web se ajusta automáticamente al tamaño de la pantalla.
- Se utiliza una única plantilla para todos los dispositivos, cualquier cambio que se desee realizar en una página solo se efectúa en esta página.

Inconvenientes

- La visualización de la página es más lenta que en el Adaptativo.
- Requiere mayor tecnología desde el punto de vista del desarrollador.
- Se muestra todo el contenido, es decir, puede ser que interese diferenciar el contenido en la versión escritorio del contenido en un móvil. Actualmente también se puede realizar, pero depende de la tecnología usada.

¹ Siglas de Responsive Web Design

1.3.2 Diseño web Adaptivo

En el diseño adaptivo se desarrollan y crean un conjunto de plantillas de diseño predefinidos basados en diferentes tamaños de pantalla de dispositivos, es decir, se muestra el diseño “adaptado” al dispositivo. A diferencia del responsivo que a través del navegador, es el servidor quien detecta el dispositivo del usuario y carga la versión adecuada del sitio según las dimensiones de su pantalla (7).

Ventajas

- Solo se utiliza una web única, la misma URL, es decir se muestra el mismo contenido con la misma url independientemente del dispositivo.
- Velocidad, ya que solo se carga el contenido específico para dicho dispositivo.
- Se puede diferenciar el contenido para cada dispositivo.

Inconvenientes

- Generalmente requiere mayor mantenimiento.
- La web es diferente en cada dispositivo, un cambio implica hacerlo en cada plantilla de cada dispositivo.

Justificación de la selección de la tecnología a utilizar

La personalización del sistema Acaxia no requiere grandes diferencias en el diseño de la capa de presentación para la correcta visualización de sus contenidos en cualquier dispositivo. El diseño responsivo adapta los componentes a cualquier resolución de pantalla, a diferencia del diseño adaptativo que no es tan flexible debido a que utiliza tamaños de pantalla preestablecidos. El Diseño Web Responsivo reestructura los elementos de la web para optimizar todo el espacio disponible y ofrecer una excelente experiencia al usuario, por lo que es más adecuado utilizar un **diseño responsivo** en la personalización de la capa de presentación de Acaxia.

1.3.3 Comparación de marcos de trabajo responsivos

Durante el estudio del Diseño Web Responsivo se identificaron varias tecnologías, las cuales son caracterizadas en la Tabla 1: Comparación de marcos de trabajo responsivos:

Tabla 1: Comparación de marcos de trabajo responsivos:

	Gumby	Foundation	Bootstrap
Definición	Marco de trabajo basado en Grid's que permite personalizar y moldear el diseño de una página adaptándose a las necesidades requeridas, como por ejemplo al tamaño de la pantalla o aplicación en la que se encuentra (8).	Marco de trabajo que permite realizar fácilmente prototipos, sistemas y aplicaciones web con una visualización correcta y adaptable a cualquier dispositivo (9).	Marco de trabajo de CSS que fue creado por Twitter para simplificar el proceso de maquetación web responsiva (10).
Utiliza	HTML5, CSS3 y PSD	JQuery, HTML5, Boilerplate y Normalizr.	HTML5, CSS3, jQuery y GitHub
Ventajas	Ofrece descarga personalizable, deja a un lado las imágenes de iconos con CSS, incluye el Font Icon que son óptimos para el rendimiento web, incluye una buena documentación y brinda soporte hasta Internet Explorer 7 y 8.	Ofrece un buen soporte para el diseño responsivo, siendo ligero y también más flexible para modificar.	Incluye gran cantidad de estilos por defecto y una colección de librerías de JavaScript. Ofrece una base bastante robusta para el diseño de interfaces.
Desventajas	Es un software propietario.	Más pequeño y menos completo que Bootstrap.	Manejo de varios archivos y variables, sumando tiempo al desarrollo.
Multi-Navegador	IE 7+, Firefox, Chrome, Safari y Opera.	IE 7+, Firefox, Chrome, Safari y Opera 11.	Firefox, Chrome, Safari, IE 7+ y Opera.

Diseño web adaptable	Si	Si	Si
Código abierto	No	Si	Si
Integración con otros marcos de trabajo	Limitada a algunos Marcos de trabajo.	Limitada a algunos Marcos de trabajo.	Si
Actualización de la versión	Si	Si	Problemas al actualizar la versión.
Comunidad	Si	No cuenta con una comunidad oficial.	Si
Documentación	Si	Muy poca documentación.	Si

Una vez finalizado el estudio de los principales marcos de trabajo para el desarrollo de sitios web responsivos se determinó que Bootstrap posee una agilidad y rapidez que facilita enormemente el diseño de interfaces para sitios responsivos. Además incluye por defecto plantillas predeterminadas y contiene una amplia documentación. El marco de trabajo seleccionado permite tener un mismo lenguaje para la comunicación entre diseñadores y el equipo de desarrollo. Es compatible con todo tipo de navegadores y dispositivos, y como elemento distintivo se conoce que Twitter liberó a Bootstrap como código abierto en agosto de 2011, por lo que cada día incluye más características gracias a la aparición de nuevos plugins² de terceros que lo sitúan como el marco de trabajo adecuado para dar solución al problema planteado.

1.4 Metodología de Diseño Web Responsivo

El objetivo de la investigación es permitir la correcta visualización del sistema Acaxia desde cualquier dispositivo, sin tener en cuenta la integración con la capa de negocio y la capa de acceso a datos. Las características comunes del RWD para la capa de presentación contrastan

² Anglismo usado en informática para describir programas que se proporcionan como añadidos o extensiones de otros programas para añadirles opciones y funciones que no traen de fábrica.

con las metodologías tradicionales de desarrollo de software, en las que el diseño es una de las fases o etapas que estas definen, por su parte en el RWD el diseño no es solo una etapa, es el proceso de conceptualización, creación, diseño y desarrollo. Se han realizado esfuerzos por hacer compatibles diferentes metodologías de desarrollo con los principios del RWD, resultando en propuestas interesantes pero muy específicas de la metodología en concreto, lo que hace que no puedan utilizarse, en la práctica, como referentes genéricos para la personalización de Acaxia.

Para la personalización de la capa de presentación de cualquier aplicación web con un diseño responsivo, es necesario seguir un conjunto de pasos y pautas que guíen el proceso de adaptación y transformación del diseño. Es por ello que para darle cumplimiento al objetivo general del presente trabajo, emplear una metodología orientada al Diseño Web Responsivo resulta de gran valor y constituye una forma más organizada y objetiva de alcanzar la meta deseada. Como el tópico de este trabajo es tan específico y no abundan los mecanismos, guías o metodologías orientados al Diseño Web Responsivo, después de un estudio de las diferentes bibliografías, se seleccionó la Metodología de Diseño Web Responsivo creada por Hernan Beati³. Esta metodología se basa en los contenidos y no en los dispositivos, proponiendo el enfoque de bocetar de mayor a menor dimensión y codificar de menor a mayor complejidad usando la técnica **Mobile First** (11).

1.4.1.1 Fases que propone la metodología

La metodología establece cinco fases donde se va concibiendo el diseño y finalmente se construye con la participación total del cliente. De esta manera todos los resultados quedan validados y se puede proceder a la fase siguiente, lo que implica que tiene un enfoque iterativo y escalonado, propiciando la concatenación de todos los elementos del diseño y la codificación del mismo. A continuación se describen brevemente las fases de la metodología en cuestión (11):

Wireframes: se realiza un esquema de página o plano de pantalla que representa el esqueleto o estructura visual de una aplicación web. Usualmente este esquema carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos. En otras palabras, se enfoca en qué hace la pantalla, no en cómo se ve.

³ Docente, Autor y Conferencista especializado en Estándares Web - Director de SaberWeb

Bocetado: se construyen las primeras vistas o prototipos de interfaz de usuario de baja fidelidad, que están diseñados en una hoja de papel a lápiz.

Prototipado: a partir del boceto a lápiz se realizan otros prototipos de interfaz de usuario de baja fidelidad en una herramienta de maquetado, de manera que el prototipo posea un diseño para diferentes resoluciones de pantalla. Después se discute con el cliente para su aprobación.

Diseño: una vez aceptadas las maquetas de interfaz de usuario, se procede al diseño de las vistas funcionales de alta fidelidad en HTML para luego validar por última vez con el cliente en esa iteración.

Código: se codifica el diseño uniendo los componentes visuales con las funciones, operaciones y métodos correspondientes, de manera que la aplicación sea totalmente funcional y pueda responder automáticamente a los cambios de dimensión del contenido de acuerdo a los dispositivos desde donde se ejecute la interfaz de usuario (11).



Figura 1: Fases de la metodología de Hernan Beati.

1.4.1 Técnica Mobile First

Para desarrollar la propuesta de solución de la investigación siguiendo las pautas de la metodología seleccionada, se aplicará la técnica Mobile First que fue ideada por Luke Wroblewski⁴. Esta técnica está concebida para trabajar con Diseño Web Responsivo que se incluye en el proceso de trabajo desde el principio, es decir, desde la fase de prototipado del diseño y transcurre por todas etapas de un RWD. La cuestión reside en tratar de comenzar un proyecto de diseño por la pantalla más pequeña e ir adaptándolo posteriormente a las más

⁴ Director de productos de Google

grandes (12). Es una técnica útil para definir la importancia de los elementos de una página teniendo en cuenta los principios del RWD, pero sobre todo, para preparar la personalización de Acaxia para futuras resoluciones o dispositivos.

1.5 Lenguajes de Programación

Lenguaje de Marcas de Hipertexto (HTML5)

HTML5 es la última evolución de la norma que define HTML⁵. El término representa un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance. Está diseñado para ser utilizable por todos los desarrolladores de la Web, permite describir con mayor precisión cuál es su contenido y proporciona una mayor optimización de la velocidad y un mejor uso del hardware (13). Se utilizó HTML5 para la personalización de Acaxia por la capacidad de incorporar una serie de etiquetas con valor semántico para la estructurar la web en bloques, dejando en un segundo plano el trabajo con div⁶ clásicos para este objetivo. Además permite la creación de aplicaciones modernas basadas en un navegador mediante la utilización de marcos de trabajo como Bootstrap y jQuery.

JavaScript

Es un lenguaje de programación interpretado basado en prototipos, que permite a los desarrolladores crear acciones en sus páginas web. Es un lenguaje independiente de la plataforma, es decir, se puede crear todo tipo de programas que pueden ser ejecutados en cualquier ordenador. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos (14).

JavaScript y XML asíncronos (Ajax).

AJAX⁷(JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla, aumentando la interactividad, velocidad y usabilidad en la misma.

⁵ Hyper Text Markup Language

⁶ Etiqueta que define una división o sección en un documento HTML.

⁷ Asynchronous JavaScript And XML

Es una combinación de tres tecnologías ya existentes: XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información; DOM accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y Script, para mostrar e interactuar dinámicamente con la información presentada y el objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor web. En algunos marcos de trabajo y en algunas situaciones concretas, se usa un objeto iframe⁸ en lugar del XMLHttpRequest para realizar dichos intercambios (15).

AJAX no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Lenguaje de Etiquetado Extensible (XML)

El Lenguaje de Etiquetado Extensible (XML⁹) es simple, pero estricto, pues juega un papel fundamental en el intercambio de una gran variedad de datos. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Dicha tecnología es un conjunto de módulos que ofrece servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. Su función principal es describir datos y no mostrarlos como es el caso de HTML (16).

Notación de Objetos de JavaScript (Json)

JSON¹⁰ es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

⁸ iframe es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

⁹ Extensible Markup Language

¹⁰ JavaScript Object Notation

JSON está constituido por dos estructuras: una colección de pares de nombre/valor. En varios lenguajes esto se conoce como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo y una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

JSON se ha convertido en un estándar en el desarrollo de aplicaciones web. Los servicios web proponen la utilización de JSON en vez de XML para permitir la integración de servicios en el navegador del usuario en vez de en el servidor. El formato JSON es seguramente la mejor opción para el intercambio de información entre el servidor y las funciones JavaScript (17).

Hojas de estilo en cascada (CSS3¹¹)

Hojas de Estilo en Cascada, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento (18). Para la personalización de Acaxia se utilizó CCS3, entre los beneficios concretos que incorpora este lenguaje se encuentran:

- Posee nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas.
- Define nuevas propiedades de estilo.
- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.).
- Incluye características avanzadas tanto para aplicar aspecto avanzado en elementos de una página como para realizar una maquetación más precisa.

¹¹ Cascading Style Sheets versión tres

1.6 Técnicas de prototipado

Técnica de prototipado de baja fidelidad

Su objetivo es verificar si los usuarios son capaces de realizar sus tareas con la interfaz propuesta. La utilización de esta técnica de prototipado no precisa incorporar avances tecnológicos; solo es necesario que capture la funcionalidad del sistema y que comunique la información y sus interacciones adecuadamente.

Esta técnica de prototipado consiste en dibujar en un papel, sin entrar en grandes detalles estéticos, las interfaces que se van a probar y valorar. Los diferentes estados de la interfaz se van dibujando en hojas separadas y mediante un proceso de ordenación que el diseñador determina, permite que el usuario final interactúe con este material simulando el funcionamiento del sistema (19).

Técnica de prototipado de alta fidelidad

Es una evolución del prototipo de baja fidelidad que resulta de la interacción con los usuarios. Se presenta en forma digital y tiene un aspecto semejante al sistema final. A este prototipo se le adicionan algunas de las funcionalidades básicas del sistema para comenzar las pruebas de usabilidad (19).

1.7 Herramienta de prototipado

Moqups 1.0

Moqups es un editor online para crear maquetas de diseños web. Es muy sencillo, rápido de utilizar, también es online y permite ser usado gratuitamente por todos los usuarios con acceso a internet. Se utiliza para definir el esquema básico de navegación y la estructura de la web, además sirven para fijar puntos de partida con tu cliente y aprobar ciertas partes y estructura (20).

1.8 IDE de desarrollo

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación o puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios (21).

NetBeans 8.0

NetBeans IDE es un producto libre y gratuito sin restricciones de uso, desarrollado por el proyecto NetBeans. Es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris, de código abierto, escrito completamente en Java. Es una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente aplicaciones del tipo web y empresarial, de escritorio y móviles utilizando la plataforma Java, así como un número importante de módulos para extenderlo a otros lenguajes como PHP (Zend y Symfony), JavaScript y Ajax, Groovy y Grails, y C / C + +. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación (22).

1.9 Navegador

Mozilla Firefox 35.0

Mozilla Firefox es uno de los navegadores web más populares del mundo. Es un navegador de código abierto. Firefox proviene de los laboratorios de Mozilla Aplicación Suite, se basa en el poderoso motor de búsqueda Gecko, el cual implementa estándares webs siempre actualizados. Por ser de software libre, este motor puede ser modificado a placer, lo que lo convierte en uno de los sistemas operativos favoritos por desarrolladores de aplicaciones (23). Se utilizará Mozilla Firefox a partir de la versión 35.0 para realizar las pruebas frecuentes al sistema.

1.10 Servidor web

Apache 2.4

Es un servidor web HTTP gratuito de código fuente abierto, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Se ejecuta en varios sistemas operativos y permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido (24).

1.11 Marcos de trabajo

Bootstraps 3.1

Bootstraps es un potente marco de trabajo de CSS que fue creado por Twitter para simplificar el proceso de maquetación web responsivo. Sin tener grandes conocimientos en CSS, con esta

herramienta se puede comenzar a maquetar un sistema web adaptable a todo tipo de dispositivos. Se utiliza cuando se trata de herramientas digitales para empresas de uso interno, donde el diseño no requiere más que pura funcionalidad. Además, para portales meramente informativos que disponen de un bajo presupuesto y que no necesitan ni captación de clientes, ni tener un valor diferencial con la competencia (25).

JQuery 2.0

Para la personalización de Acaxia se utilizó jQuery en su versión 2.0 que es un marco de trabajo para el lenguaje JavaScript, que implementa una serie de clases (de programación orientada a objetos) que permite programar sin preocuparse del navegador con el que está visitando el usuario, ya que funciona de forma exacta en todas las plataformas más habituales. Este marco de trabajo JavaScript, ofrece una infraestructura con la que se tiene mayor facilidad para la creación de aplicaciones complejas del lado del cliente. JQuery tiene licencia para uso en cualquier tipo de plataforma, personal o comercial (26). Es un producto con gran aceptación por parte de los programadores y un considerable grado de penetración en el mercado. Además, es un producto estable, bien documentado y con un amplio equipo de desarrolladores a cargo de la mejora y actualización del marco de trabajo.

1.12 Conclusiones

Se identificó y caracterizó en el estudio realizado las tecnologías para el diseño multidispositivo, se seleccionó el Diseño Web Responsivo por su capacidad de adaptarse automáticamente a cualquier resolución de pantalla y la técnica Mobile First que define cómo realizar la codificación. Se definió el marco de trabajo Bootstrap 3.1 para la personalización de la capa de presentación del sistema Acaxia por la agilidad y rapidez con que cuenta, facilitando el diseño de interfaces para sistemas responsivos y la metodología de Hernán Beati para guiar el proceso de personalización.

Capítulo 2: Propuesta de solución

2.1 Introducción

En el presente capítulo se abordarán las características generales de la personalización de la capa de presentación de Acaxia para su correcta visualización en dispositivos móviles. También se definirá la propuesta de solución, teniendo en cuenta las pautas de diseño a poner en práctica en el diseño de la personalización de la capa de presentación del sistema Acaxia, apoyadas en la documentación de la aplicación contenida en el portafolio de proyecto.

2.2 Artefactos del portafolio de proyecto

En el portafolio de proyecto se registraron 180 requisitos funcionales y 27 requisitos no funcionales. De acuerdo a las características de la solución que se propone y el objetivo general de la investigación, se emplearán 138 requisitos funcionales y 13 requisitos no funcionales relacionados con la capa de presentación. Por tanto, las funcionalidades que se modificarán se listan en la Tabla 2: Requisitos funcionales la capa de presentación del Sistema Acaxia:

Tabla 2: Requisitos funcionales la capa de presentación del Sistema Acaxia:

REQUISITOS FUNCIONALES	
RF 1	Gestionar servidor
RF 1.1	Adicionar servidor de autenticación
RF 1.2	Modificar servidor de autenticación
RF 1.3	Adicionar servidor de base de datos
RF 1.4	Modificar servidor de base de datos
RF 1.5	Eliminar servidor
RF 1.6	Listar servidores
RF 1.7	Probar conexión
RF 2	Gestionar gestor de base de datos
RF 2.1	Adicionar gestor de bases de datos a un servidor
RF 2.2	Eliminar gestor de bases de datos asociado a un servidor
RF 2.3	Listar gestores de bases de datos asociado a un servidor
RF 2.4	Listar servidores de base de datos
RF 3	Gestionar roles de bases de datos

RF 3.1	Listar servidores
RF 4	Gestionar nomenclador de gestores de bases de datos
RF 4.1	Adicionar nomenclador de gestor de bases de datos
RF 4.2	Modificar nomenclador de gestor de bases de datos
RF 4.3	Eliminar nomenclador de gestor de bases de datos
RF 4.4	Listar nomenclador de gestores de bases de datos
RF 4.5	Buscar nomenclador de gestor de bases de datos
RF 5	Gestionar sistemas
RF 5.1	Adicionar sistema
RF 5.2	Modificar sistema
RF 5.3	Eliminar sistema
RF 5.4	Importar sistema
RF 5.5	Exportar sistema
RF 5.6	Listar sistemas registrados
RF 5.7	Listar subsistemas
RF 6	Gestionar compartimentación de sistema
RF 6.1	Listar sistemas, funcionalidades y acciones compartimentadas
RF 6.2	Listar dominio
RF 6.3	Asociar sistemas, funcionalidades y acciones compartimentadas a dominio compartimentado
RF 7	Gestionar compartimentación de dominio
RF 7.1	Listar usuario compartimentado
RF 7.2	Listar dominio para compartimentar el acceso de usuarios
RF 7.3	Asociar usuario a dominio compartimentado
RF 7.4	Buscar usuarios compartimentados
RF 8	Gestionar compartimentación de rol
RF 8.1	Listar roles compartimentados
RF 8.2	Listar dominio para compartimentar el acceso de roles
RF 8.3	Asociar roles a dominio compartimentado
RF 8.4	Buscar roles compartimentados
RF 9	Gestionar nomenclador para objetos de base de datos

RF 9.1	Adicionar objeto de base de datos
RF 9.2	Modificar objeto de base de datos
RF 9.3	Eliminar objeto de base de datos
RF 9.4	Listar objeto de base de datos
RF 10	Gestionar usuario
RF 10.1	Adicionar usuario
RF 10.2	Modificar usuario
RF 10.3	Eliminar usuario
RF 10.4	Asignar roles
RF 10.5	Buscar rol
RF 10.6	Listar rol
RF 10.7	Listar entidades
RF 10.8	Activar usuario
RF 10.9	Desactivar usuario
RF 10.10	Listar sesiones activas en el sistema
RF 10.11	Cerrar sesión activa
RF 10.12	Cambiar contraseña
RF 10.13	Buscar usuarios
RF 11	Gestionar nomenclador de tipos de conexión
RF 11.1	Adicionar conexión
RF 11.2	Modificar conexión
RF 11.3	Eliminar conexión
RF 11.4	Listar tipos de conexión
RF 12	Gestionar perfiles de usuarios
RF 12.1	Buscar perfil de usuario
RF 12.2	Adicionar perfil de usuario
RF 12.3	Modificar perfil de usuario
RF 12.4	Eliminar perfil de usuario
RF 13	Gestionar autenticación de usuario
RF 13.1	Crear solicitud de autenticación
RF 13.2	Validar solicitud de autenticación

RF 13.3	Validación de los datos de logueo
RF 13.4	Crear respuesta
RF 13.5	Validar respuesta
RF 14	Gestionar nomenclador de dominio
RF 14.1	Adicionar nomenclador de dominio
RF 14.2	Modificar nomenclador de dominio
RF 14.3	Eliminar nomenclador de dominio
RF 14.4	Listar estructuras de dominio
RF 14.5	Listar nomenclador de dominio
RF 14.6	Buscar nomenclador de dominio
RF 15	Gestionar nomenclador de idioma
RF 15.1	Adicionar nomenclador de idioma
RF 15.2	Modificar nomenclador de idioma
RF 15.3	Eliminar nomenclador de idioma
RF 15.4	Listar nomenclador de idioma
RF 16	Gestionar nomenclador de expresiones
RF 16.1	Adicionar nomenclador de expresiones
RF 16.2	Modificar nomenclador de expresiones
RF 16.3	Eliminar nomenclador de expresiones
RF 16.4	Listar nomenclador de expresiones
RF 16.5	Buscar nomenclador de expresiones
RF 17	Gestionar nomenclador de temas
RF 17.1	Adicionar nomenclador de temas
RF 17.2	Modificar nomenclador de temas
RF 17.3	Eliminar nomenclador de temas
RF 17.4	Listar nomenclador de temas
RF 18	Gestionar nomenclador de escritorios
RF 18.1	Adicionar nomenclador de escritorios
RF 18.2	Modificar nomenclador de escritorios
RF 18.3	Eliminar nomenclador de escritorios
RF 18.4	Listar nomenclador de escritorios

RF 19	Gestionar nomenclador de claves
RF 19.1	Modificar nomenclador de claves
RF 19.2	Listar nomenclador de claves
RF 20	Gestionar funcionalidades
RF 20.1	Eliminar funcionalidad
RF 20.2	Buscar funcionalidad
RF 20.3	Listar funcionalidad
RF 20.4	Listar sistemas
RF 21	Gestionar acciones
RF 21.1	Adicionar acción
RF 21.2	Modificar acción
RF 21.3	Listar acción
RF 21.4	Buscar acción
RF 21.5	Eliminar acción
RF 21.6	Buscar acciones del controlador
RF 22	Gestionar servicios que presta
RF 22.1	Listar sistemas
RF 23	Gestionar servicios que consume
RF 23.1	Listar sistemas
RF 23.2	Listar servicios
RF 24	Gestionar funciones
RF 24.1	Listar sistemas
RF 24.2	Listar funciones
RF 25	Gestionar parámetros
RF 25.1	Listar sistemas
RF 25.2	Listar parámetros
RF 26	Gestionar acciones y reportes
RF 26.1	Listar sistemas
RF 26.2	Buscar acciones
RF 26.3	Listar acciones
RF 26.4	Listar reportes

RF 27	Gestionar roles
RF 27.1	Adicionar rol
RF 27.2	Modificar rol
RF 27.3	Eliminar rol
RF 27.4	Buscar roles
RF 27.5	Listar roles
RF 27.6	Listar sistemas
RF 27.7	Listar acciones autorizadas
RF 27.8	Listar acciones no autorizadas
RF 27.9	Autorizar acciones
RF 27.10	Desautorizar acciones

Entre los requisitos no funcionales, es preciso tener en cuenta aquellos que son consecuentes con el desarrollo web para dispositivos móviles. Por tanto la solución debe satisfacer las siguientes condiciones:

- **Usabilidad**

RNF 1: El sistema podrá ser accedido desde cualquier dispositivo y por todos los usuarios que posean conocimientos básicos en el manejo de estos.

RNF 2: Todas las páginas deben poseer un menú de navegación.

RNF 3: Los usuarios pueden ver la información sin la necesidad de utilizar desplazamiento horizontal o paginados excesivos.

- **Soporte**

RNF 4: La aplicación contará antes de su puesta en marcha con un período de pruebas.

- **Portabilidad**

RNF 5: El sistema debe ser compatible con todo tipo de navegadores, independientemente del sistema operativo que se utilice.

- **Restricciones de apariencia o interfaz externa**

RNF 6: Todos los textos y mensajes se muestran en idioma español.

RNF 7: El sistema debe contar con un identificador en la parte superior, ya sea el nombre o el logo del mismo.

RNF 8: La aplicación debe visualizarse correctamente en cualquier dispositivo, independientemente de las características y propiedades que estos posean.

RNF 9: El sistema utiliza la menor cantidad posible de imágenes.

- **Restricciones de interfaz de comunicación**

RNF 10: Los teléfonos móviles deben conectarse a la aplicación a través de la red Wi-Fi.

- **Confiabilidad**

RNF 11: El sistema debe utilizar contraseña de acceso para llevar a cabo la autenticación del usuario.

RNF 12: El sistema debe garantizar protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

RNF 13: El sistema debe realizar verificaciones sobre las acciones irreversibles (eliminaciones).

2.3 Propuesta de diseño del sistema

Para lograr un mayor entendimiento de lo que se propone se decide seleccionar tres funcionalidades para estructurar la interfaz, documentar las modificaciones y probar las adaptaciones inherentes a la solución. Dígase entonces, que los requisitos funcionales **Gestionar autenticación de usuario, Gestionar usuarios y Adicionar servidor** servirán de muestra para indicar el progreso de la investigación.

2.3.1 Pautas de diseño para sitios web responsivos

El Diseño Web Responsivo comprende una serie de técnicas y pautas de diseño que permiten adaptar sitios web al entorno de navegación del usuario, entendiendo como entorno de navegación la multiplicidad de dispositivos, móviles o no, por medio de los cuales los usuarios pueden acceder y navegar en Internet. La World Wide Web Consortium (W3C) utilizando el nombre de "One Web" para definir el camino al que debería apuntar el diseño web para múltiples dispositivos, define un conjunto de pautas, recomendaciones y buenas prácticas a tener en cuenta en el desarrollo web para móviles, entre las que se encuentran (27):

1. Diseñar para una web única.

- Asegurarse de que el contenido al que se accede desde una URL aporta la misma información esencial al usuario, independientemente del dispositivo que utilice.
 - Aprovechar las capacidades del dispositivo para proporcionar al usuario una mejor experiencia.
 - Realizar pruebas sobre dispositivos móviles y emuladores.
2. Emplear estándares web.
 - Crear documentos que sean válidos según las gramáticas formales disponibles.
 - Usar un formato que sea compatible con todos los dispositivos.
 - Asegurarse de que la codificación de caracteres del contenido es compatible con todos los dispositivos.
 - Intentar que tus mensajes de error sean informativos y facilitar la forma de regresar al contenido anterior.
 3. Evitar riesgos conocidos.
 - No usar tablas anidadas, ni marcos.
 4. Ser prudente con las limitaciones de los dispositivos.
 - Asegurar que la información transmitida mediante colores también puede ser comprendida sin color.
 5. Optimizar la navegación.
 - Concentrar la navegación en la parte superior de la página y reducirla al máximo.
 - Identificar claramente el destino de cada enlace.
 6. Comprobar gráficos y colores.
 - Facilitar un equivalente en forma de texto para cada elemento no textual.
 - Asegurar que entre el color del fondo y el del primer plano hay suficiente contraste.
 - Al usar imágenes de fondo, se debe asegurar que el contenido sigue siendo legible en cualquier dispositivo.

- No utilizar medidas en píxeles ni unidades absolutas en los valores de los atributos del lenguaje de etiquetado, ni en los valores de las propiedades de las hojas de estilo.
7. Construir en pequeño.
 - Limitar el desplazamiento a una sola dirección, a no ser que sea imposible evitar un desplazamiento secundario.
 8. Facilitar la entrada de datos.
 - Reducir el uso del teclado al mínimo.
 - Siempre que sea posible, establecer valores preseleccionados por defecto.
 - Crear un orden lógico mediante enlaces, controles de formulario y objetos.
 9. Pensar en los usuarios de la web móvil.
 - Poner un título a la página que sea corto pero descriptivo.
 - Utilizar un lenguaje sencillo y claro.
 - Asegurarse de que el contenido esencial de la página esté antes que el contenido que no lo es.
 - Asegurarse de que el contenido es adecuado para su uso en un contexto móvil.

De igual forma Google, el mayor motor de búsqueda del mundo, recomienda usar un diseño responsivo y define un conjunto de pautas, entre las que se encuentran (28):

1. Las páginas deben verse de manera legible en cualquier resolución de pantalla.
2. Concentrar la navegación en la parte superior de la página y marcar un conjunto de contenidos del sitio para que sean visibles en cualquier dispositivo.
3. Nunca mostrar una barra de desplazamiento horizontal, sea cual sea el tamaño de la ventana.

2.3.2 Propuesta de diseño visual

En el desarrollo de la personalización de la capa de presentación del sistema Acaxia con un Diseño Web Responsivo es necesario tener en cuenta las características del sistema de

seguridad Acaxia, además de poner en práctica las pautas de Diseño Web Responsivo definidas anteriormente. La personalización constará con las siguientes características:

2.3.2.1 Características de la interfaz visual

La interfaz visual debe tener un conjunto de peculiaridades para lograr una correcta visualización de sus contenidos en dispositivos móviles. A continuación se describen algunas de estas.

- **Optimizar la utilización de imágenes:** Las imágenes obtendrán el ancho porcentualmente permitiendo ajustarse a la pantalla del dispositivo.
- **Utilizar una estructura en forma de columnas:** Diagramación estructurada en un sistema de 12 columnas las cuales se reajustan según el tamaño de pantalla de dispositivo.
- **Utilizar la tipografía adecuada:** La tipografía debe ser legible en pantalla, el tamaño de letra *font-size* es 14px¹² y el interlineado *line-height* es 1.428. Se debe resaltar los subtítulos.
- **Mantener el menú de navegación en todas las páginas:** El menú de navegación se mantiene en la parte superior en todas las páginas del sistema.
- **Utilizar pautas cromáticas:** Las pautas cromáticas a utilizar son las mostradas en la Figura 2: Pautas cromáticas.



Figura 2: Pautas cromáticas.

2.3.2.2 Diseño centrado en contenidos

El diseño de la personalización se centra en los contenidos, tomando en consideración el orden de prioridad y la organización de estos en el sistema de la siguiente forma:

¹² Abreviatura de píxeles, anglicismo que define la menor unidad homogénea en color que forma parte de una imagen digital.

- Priorizar los contenidos como se muestra a continuación.

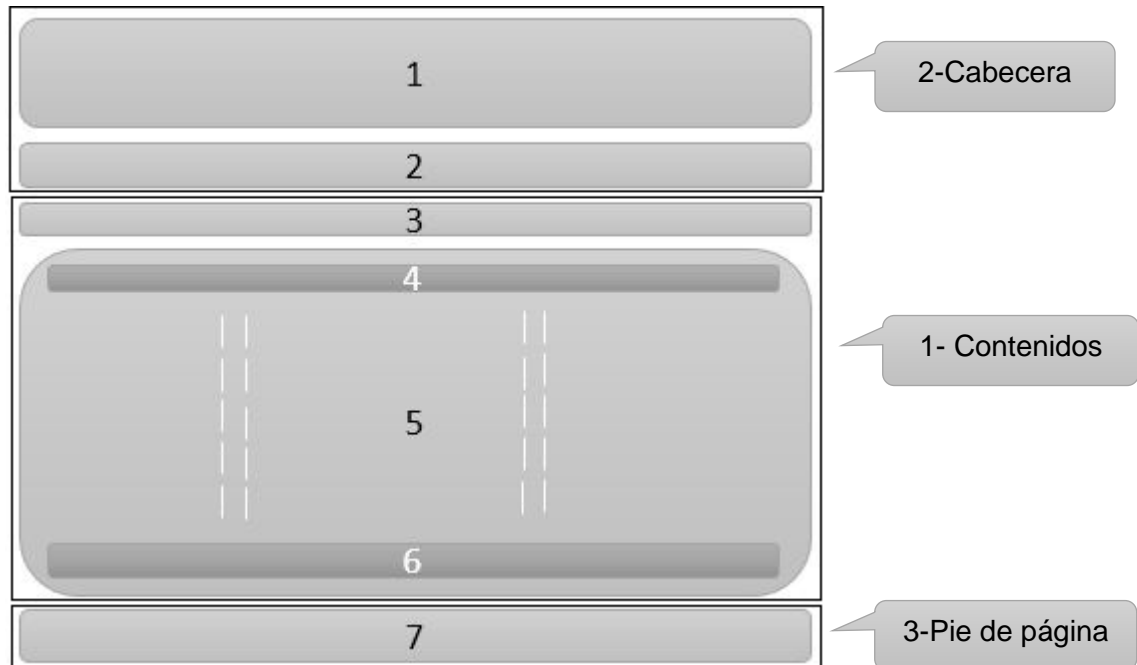


Figura 3: Vista del diseño centrado en contenidos.

1. Área de menú
2. Área de identificación
3. Área de opciones
4. Área de búsqueda
5. Área de contexto
6. Área de paginación
7. Área de pie de página

2.3.2.3 Bocetado de la interfaz

Los bocetos y dibujos en papel son una vía rápida y económica para crear prototipos de interfaz de usuario. Producto de la retroalimentación con usuarios y clientes o a partir del surgimiento de nuevas ideas, estos bocetos son modificados con facilidad y van evolucionando hasta convertirse en los prototipos de interfaz de usuario con valores agregados como los colores, el uso de componentes visuales específicos y los logos de identidad del negocio. A continuación puede

evidenciarse uno de los bocetos de la investigación en la Figura 4: Prototipo en papel de la vista general de procesos.

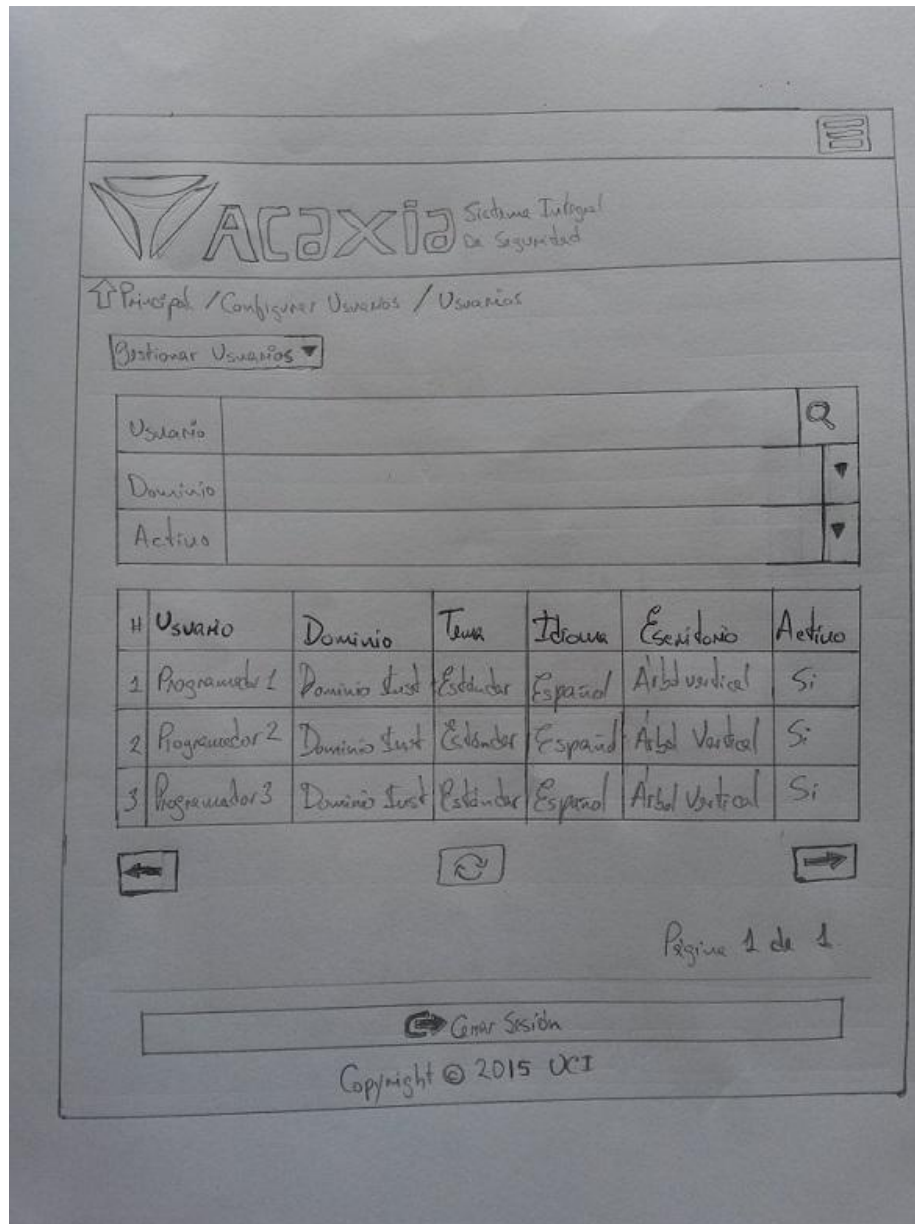


Figura 4: Prototipo en papel de la vista general de procesos.

2.3.2.4 Prototipado de la interfaz

Los prototipos anteriormente mencionados son creados a partir de la aprobación de los bocetos discutidos con el cliente. Para poderlos realizar es necesario emplear una herramienta de

maquetación, en este caso la seleccionada para este trabajo: Moqups. En la Figura 5: Prototipo realizado con Moqups 1.0, se ilustra un prototipo de baja fidelidad, con los valores agregados antes mencionados.



Figura 5: Prototipo realizado con Moqups 1.0.

2.3.2.5 Diseño de la interfaz visual

En esta fase, luego de haberse discutido y aprobado con el cliente tanto los bocetos como los prototipos de interfaz de usuario, se procede a diseñar en HTML5 los prototipos funcionales incorporándoles estilos web, efectos de transparencia, iconografía funcional y demás componentes visuales. A continuación se muestra en la Figura 6: Prototipo funcional de Interfaz de usuario, la interfaz del requisito funcional Gestionar usuario.



Figura 6: Prototipo funcional de Interfaz de usuario.

2.3.2.6 Acciones para reestructurar la interfaz

La interfaz debe ajustarse automáticamente a toda resolución de pantalla, para ello el sistema debe ser capaz de reestructurar los contenidos eliminando el desplazamiento horizontal, considerando las acciones siguientes.

- Dejar que se angoste el ancho en porcentaje (contenido celeste).
- Dejar de flotar y aumentar ancho en porcentaje (columnas grises).
- Ocultar contenido.
- Mostrar contenido.

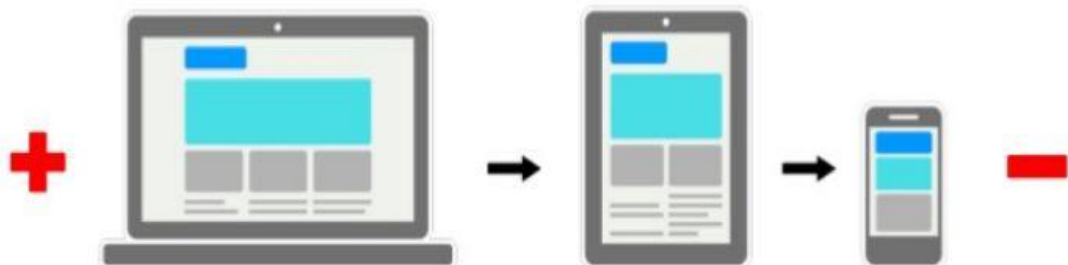


Figura 7: Vista de reestructuración de la interfaz.

2.3.2.7 Definición de los puntos de ruptura

El marco de trabajo Bootstrap implementa las acciones mencionadas en el punto anterior a través de los Media Queries, que por defecto definen puntos de ruptura como se muestra en la Tabla 3: Puntos de ruptura definidos por Bootstrap (29).

Tabla 3: Puntos de ruptura definidos por Bootstrap:

	Dispositivos extra pequeños (móvil < 768px)	Dispositivos pequeños (767px < tablet < 992px)	Dispositivos medianos (991px < escritorio < 1200px)	Dispositivos grandes (escritorio > 1199px)
Comportamiento del marco	Horizontal	Comienzo del colapsado por los puntos de ruptura anteriores		
Número de columnas	12	12	12	12
Ancho máximo del contenedor	Automático	750px	970px	1170px
Clase definida para implementar	.col-xs-	.col-sm-	.col-md-	.col-lg-
Ancho máximo de columna	Automático	60px	78px	95px

2.3.2.8 Estructura de la interfaz visual

La interfaz visual se estructurará empleando la técnica Mobile First. La estructura muestra en los prototipos de las vistas principales como cambia el diseño de la información contenida en la capa de presentación de Acaxia, en tres anchos de pantalla diferentes (320px, 768px y 1024px) como se muestra en la vista de formulario:

Vista de Formularios

En las aplicaciones web, el formulario es una parte de la interfaz, que es presentado al usuario en un navegador para ser rellenado a través de la red. Generalmente cuando se ingresan los datos, se envían a un servidor web para ser procesados (30). Los formularios de la personalización de la capa de presentación de Acaxia estarán compuestos por regiones en forma de barras de contenido horizontales, conformando una ventana que se irá reestructurando teniendo en cuenta el Diseño Web Responsivo.

Se estructurarán exactamente de la misma manera que están en el sistema Acaxia actual, en dependencia de la necesidad y de la complejidad del mismo, prestando fundamental atención a los Grids, los Botones, las Validaciones y la Definición de Mensajería.

1. Título del formulario

El título debe estar en correspondencia con el formulario definido, de manera que lo identifique.

2. Herramientas para manipulación del formulario

Se colocan botones con texto y/o iconos que definen acciones para la manipulación de los datos.

3. Controles del formulario

Estará conformado por etiquetas de tipo *Input*, *Select*, *TextArea*, *Table*, *List* y *Label*. Además contará con *GridPanels* y *TreePanels*. Una misma área horizontal se puede dividir según la necesidad de espacio siempre que no pierda la estructura principal de la pauta de diseño.

4. Botones

Se ubicarán de izquierda a derecha los botones de cancelación y luego de aceptación, alineados a la derecha.



Figura 8: Prototipos de interfaz de la Vista general de gestión de procesos para dispositivos móviles.

The image shows a web browser window with the address bar displaying 'http://PersonalizacionAcaxia/ConfigurarServidores/GestionarSevidor/#AdicionarServidor'. The main content is a form titled 'Adicionar servidor'. The form has several sections: 'Servidor *' with a text input containing 'ej: smtp.uci.cu'; 'Dirección IP *' with a text input containing 'ej: 10.0.0.1'; 'Tipo de Servidor *' with a dropdown menu set to 'Servidor de base de datos'; 'Puerto *' with a spinner box set to '389'; 'Datos de la conexión' with radio buttons for 'LDAP' (selected) and 'OpenLDAP'; 'Base DN *' with a text input; a section titled 'Datos de la conexión' with a dropdown arrow, containing 'DN de enlace*' and 'Contraseña *' text inputs; and 'Descripción' with a large text area. At the bottom are four buttons: 'Cancelar' (red), 'Probar' (green), 'Aplicar' (green), and 'Aceptar' (green).

Figura 9: Prototipo de interfaz de la Vista general de gestión de procesos para escritorio.

2.4 Patrones de diseño e implementación

Para la propuesta de solución se utiliza el Diseño Web Responsivo, el cual a su vez cuenta con sus propios patrones de diseño. El marco de trabajo Bootstrap implementa estos patrones implícitamente en sus funcionalidades. A continuación se presenta una breve descripción de los principales patrones empleados en la capa de presentación.

Patrones para el Diseño Web Responsivo:

Más fluido (Mostly Fluid): es uno de los más populares dentro del diseño responsivo, en este patrón, un primer bloque protagonista va seguido de otros dos con menos relevancia y en el momento de visualizarse en un dispositivo se adapta siguiendo el flujo según la prioridad de los bloques (31).

Caída de columna (Column Drop): este tipo de diseño se adapta hasta el punto en el que no se pueden manejar las tres columnas en una sola fila, entonces la última columna pasa a la parte inferior y luego el flujo se sigue desarrollando hasta llegar a una sola columna.

Movimiento de estructura (Layout shifter): este cambia ligeramente el diseño, esta estructura está orientada a dar una experiencia diferente en dispositivos de pantallas más reducidas. Muy usado para reacomodar menús y cabeceras (31).

Patrones GoF utilizados en la personalización del sistema Acaxia:

En el ambiente de desarrollo para la propuesta de solución se utiliza el marco de trabajo jQuery, el cual a su vez implementa independientemente sus propios patrones de diseño. A continuación se presenta una breve descripción de los principales patrones GoF¹³ empleados en la capa de presentación.

Fachada (Facade): Se utiliza para simplificar una interfaz compleja y reducirla a las funciones necesarias en nuestro sistema o a funciones mejor diseñadas y más simples que las originales. Fachada se puede encontrar con frecuencia a través de la librería jQuery y proporciona a los desarrolladores acceso fácil a las implementaciones para el manejo del DOM, de especial interés, en navegadores Ajax (32). JQuery proporciona ejemplos bien diseñados de este patrón, dentro de los que se encuentra el selector de elementos del DOM: **\$("#selector")**.

Decorador (Decorator): es un patrón de diseño estructural que tienen como objetivo promover la reutilización de código. Este permite añadir funcionalidades o responsabilidades a un objeto de forma dinámica y provee una alternativa muy flexible a la creación de subclases para extender funcionalidades. Esto admitirá, en tiempo de ejecución, añadir esta funcionalidad o responsabilidad extra que necesita nuestro objeto. El patrón Decorador puede ser implementado con jQuery (32). La funcionalidad **jQuery.extend()** permite extender o combinar dos o más objetos y sus propiedades juntos en un solo objeto, ya sea en tiempo de ejecución o dinámicamente en un momento posterior. En las funcionalidades **options-settings** e **InsertAfter** de jQuery es donde más se evidencian.

2.5 Conclusiones

Se definieron que 140 requisitos funcionales y 15 requisitos no funcionales se incluirían en la personalización de la capa de presentación. Se obtuvo el diseño visual de la solución a partir de las pautas definidas en el Diseño Web Responsivo. Se evidenció la utilización de patrones GoF y patrones para el Diseño Web Responsivo.

¹³ Gang of Four

Capítulo 3: Implementación y pruebas

3.1 Introducción

En el presente capítulo se abordará todo lo referente a la implementación de la personalización de la capa de presentación de Acaxia de forma que se aplique la técnica Mobile First, los estándares de codificación, aspectos a tener en cuenta en la implementación y los detalles de esta. Además se realizará la verificación y validación de la solución propuesta a través de diferentes tipos de pruebas.

3.2 Técnica de codificación

Para la codificación de la solución se utilizará la técnica Mobile First con el objetivo de seguir el principio de codificar de menor a mayor complejidad, utilizando CSS para ajustar los componentes en resoluciones de móviles más pequeños, para luego mediante estilos CSS ir flotando y estirando los componentes del sistema en pantallas más grandes como se muestra a continuación:

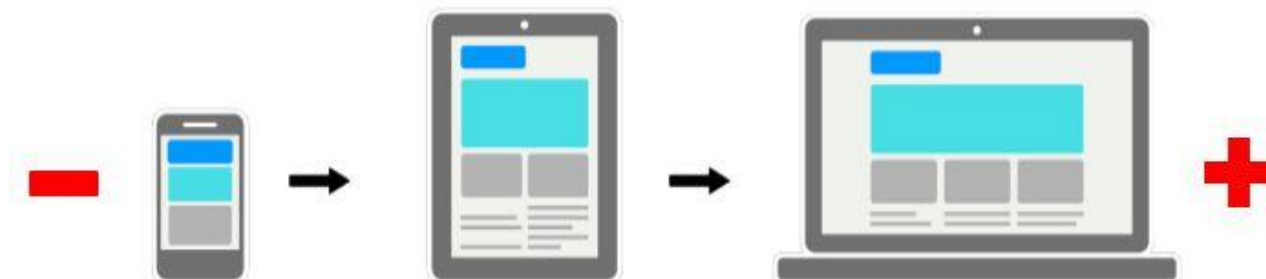


Figura 10: Codificación usando la técnica de Mobile First.

3.3 Elementos a tener en cuenta en la implementación

Para realizar la codificación de la capa de presentación de Acaxia para la correcta visualización de sus contenidos en cualquier dispositivo, es necesario tener en cuenta tres aspectos fundamentales que rigen un buen funcionamiento de un sistema web responsivo.

3.3.1 Media Queries

Una media query es una característica CSS que supone un mecanismo para identificar no sólo el tipo de medio en el que se visualiza la aplicación, si no también algunas características físicas del dispositivo, como el tamaño o la resolución de pantalla. Permiten construir una serie de reglas de

estilo que sólo se ejecutarán cuando el dispositivo cumpla las características especificadas. En el siguiente ejemplo se define la utilización de las media queries para dispositivos pequeños, utilizando los puntos de rupturas definidos por Bootstrap y descritos en el acápite [2.3.2.7](#), para la codificación del sistema:



Figura 11: Ejemplo de vista de dispositivos pequeños.

- Ejemplo de código

```
@media (max-width: 768px) {
```

```
/* reglas CSS*/
```

```
}
```

3.3.2 Ancho de los div

El diseño para la capa de presentación del sistema de seguridad Acaxia debe ser fluido y adaptable dejando atrás los pixeles y utilizando porcentajes para que la estructura cambie su aspecto automáticamente al visualizar el sistema a través de la pantalla pequeña de un móvil. Donde se mantendrán todos los elementos de la misma pero de un tamaño más pequeño, reorganizados y proporcionales entre ellos, con respecto al tamaño original. A continuación se

muestra como Bootstrap implementa el ancho de los div basados en el tamaño de las columnas, se muestra la clase **col-md-*** que es el tamaño para dispositivos medianos.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Figura 12: Disposición de las columnas según la clase.

```
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>
```

Figura 13: Utilización de la clase col-md-*

3.3.3 Elementos de ancho variable

Bootstrap implementa para este tipo de elementos el ancho en porcentajes, para conocer el ancho que debería tener un elemento se debe conocer el ancho inicial del mismo, y dividirlo entre el ancho del elemento original como se ilustra a continuación:

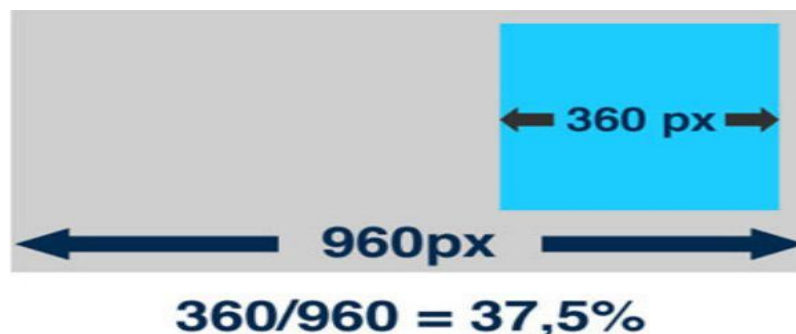


Figura 14: Propiedad de elementos de ancho variable.

3.4 Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. El código fuente debe reflejar un estilo armonioso con vistas de que un proyecto de software se convierta en un producto fácil de comprender y de mantener. A continuación se describen e ilustran los estándares utilizados en la implementación:

Todas las páginas HTML deben ser verificados contra el validador del W3C para garantizar que el margen de beneficio está bien formado. Esto en sí mismo no es directamente indicativo del buen código, pero ayuda a eliminar a los problemas que son capaces de ser probado a través de la automatización (33).

- **Atributos y Etiquetas:** Todas las etiquetas y atributos deben escribirse en minúsculas.
- **Comillas:** todos los atributos deben tener un valor, y deben utilizar comillas dobles o simples.
- **Sangría:** siempre debe reflejar la estructura lógica.

```
<!-- Inicio del Modal Eliminar Usuario-->
<div class="modal fade" id="modalElimUser" tabindex="-1" role="dialog" aria-labelledby="modalElimUser" aria-hidden="true">
  <div class="modal-dialog modal-sm">
    <div class="modal-content">
      <div class="modal-header" style="background-color: #5cb85c">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title" style="color: white; text-align: center">
          <strong>Eliminar Usuario</strong>
        </h4>
      </div>
      <div class="modal-body">
        &nbsp;&nbsp;&nbsp;&¿Est&aacute; seguro que desea eliminar el elemento?
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-danger" data-dismiss="modal">
          <span class="glyphicon glyphicon-remove-circle"></span> &nbsp;&nbsp;&Cancelar
        </button>
        <button type="submit" class="btn btn-success">
          <span class="glyphicon glyphicon-trash">
          </span> &nbsp;&nbsp;&Aceptar
        </button>
      </div>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
<!-- Final del Modal Eliminar Usuario-->
```

Figura 15: Ejemplo de código HTML5.

Dentro de las hojas de estilo básicas, a menudo se encuentran inconsistencias, para que esto no ocurra, se siguen las normas de codificación CSS de WordPress Core Handbook (34).

➤ **Estructura:**

1. Utilizar tabuladores, no espacios, para aplicar sangría a cada propiedad.
2. Añadir una línea en blanco entre las secciones y entre los bloques en una sección.
3. Cada selector debe estar en su propia línea, que termina en una coma o una llave de apertura.
4. Los pares propiedad-valor deben estar en su propia línea, con un tabular de hendidura y un punto y coma final.
5. La llave de cierre debe estar alineado a la izquierda, con el mismo nivel de sangrado como el selector de apertura.

➤ **Selectores:**

1. Comenzar con minúscula el selector y las palabras subsiguientes con mayúscula.

2. Utilizar selectores legibles que describen qué elementos de que estilo.
3. Los selectores de atributos deben utilizar comillas simples alrededor de los valores.

➤ **Propiedades:**

1. Las propiedades deben ser seguidas por dos puntos y un espacio.
2. Todas las propiedades y valores deben ser minúsculas, a excepción de los nombres de fuente.
3. Utilizar el código hexadecimal para los colores, o rgba () si se necesita opacidad.
4. Evitar el formato RGB y mayúsculas, y acortar los valores cuando sea posible: #fff lugar de #FFFFFF.

➤ **Valores:**

1. Poner después de los dos puntos espacio antes del valor.
2. Terminar siempre en un punto y coma.
3. Los valores separados por comas múltiples para una propiedad deben estar separados por un espacio o un salto de línea, incluso dentro de rgba().
4. Cada valor subsecuente después de la primera línea debe estar entonces en una nueva línea, con sangría hasta el mismo nivel que el selector y luego espaciados hacia la izquierda a alinear con el valor anterior.

```
#divIndexCompleto {  
    background: url(../Imagenes/logo.png) no-repeat center center fixed;  
    margin-left: 15%;  
    width: 80%;  
}  
  
#divIndexPrincipal {  
    margin-top: 18%;  
    margin-left: 20%;  
    margin-bottom: 18%;  
}
```

Figura 16: Ejemplo de código CSS.

Para utilizar la librería JavaScript/JQuery todos los archivos deberán seguir el estándar de codificación del jQuery Foundation, JavaScript Style Guide (35):

➤ **Espaciado:**

1. Sangría con tabuladores.
2. No usar espacios en blanco al final de la línea o líneas en blanco.
3. Las líneas no deben superar los 100 caracteres (contando los tabulados como 4 espacios).
4. Las sentencias **if / else / for / while / try**, siempre deben tener llaves e ir en varias líneas.
5. Los operadores caracteres especiales unarios (por ejemplo, **!**, **++**) No deben tener espacio al lado de su operando.
6. Ningún (,) y (;) debe tener espacio anterior.
7. Cualquier (;) que se utilice como un terminador de sentencia debe estar al final de la línea.
8. No usar espacios de relleno en construcciones vacías (por ejemplo, {}, [], fn ()).

➤ **Declaraciones de líneas múltiples:**

1. Cuando una declaración es demasiado larga para ocupar una sola línea, se debe realizar un salto de línea después de un operador.
2. Cuando una condicional es demasiado larga para ocupar una sola línea, las líneas sucesivas deben tener una sangría un nivel extra para distinguirlos del cuerpo.

➤ **Comentarios**

1. Los comentarios siempre son precedidos por una línea en blanco.
2. Los comentarios comienzan con una primera letra mayúscula, pero no requieren de un punto al final, a menos que se esté escribiendo oraciones completas.
3. Debe haber un espacio entre el símbolo de comentario y el texto del comentario.

➤ **Convenciones de nomenclatura**

1. La primera letra minúscula y la primera letra de cada palabra subsecuente debe comenzar con mayúscula, con el resto de las letras en minúscula.
2. Los nombres deben ser descriptivos, pero no en exceso.
3. Se permiten excepciones para los iteradores, tales como el uso de *i* para representar el índice en un bucle.

```
/* Selecciona la fila y se cambia el estilo cuando le das click con el ratón */  
function CambEstilCeldaTablaGestUser(x)  
{  
    $(document).ready(function() {  
        if (document.getElementsByClassName("trselected").length > 0) {  
            var element = document.getElementsByClassName("trselected");  
            if (parseInt(element[0].id) % 2 !== 0)  
                element[0].className = "impar";  
            else  
                element[0].className = "";  
        }  
  
        x.className = "trselected";  
        var usuarioSeleccionado = $(x).find('td').eq(1).html();  
        var entidadSeleccionada = $(x).find('td').eq(3).html();  
        var areaSeleccionada = $(x).find('td').eq(4).html();  
        var cargoSeleccionado = $(x).find('td').eq(5).html();  
  
        $('#inputModUser').val($.trim(usuarioSeleccionado));  
        $('#inputModEntidad').val($.trim(entidadSeleccionada));  
        $('#inputAddArea').val($.trim(areaSeleccionada));  
        $('#inputAddCargo').val($.trim(cargoSeleccionado));  
    });  
}
```

Figura 17: Ejemplo de código Javascript/JQuery.

3.5 Detalles de la implementación de la solución

3.5.1 Componentes visuales

Para lograr la correcta visualización de la capa de presentación del sistema de seguridad Acaxia y su correcto funcionamiento se decide utilizar los componentes propios de HTML5. A continuación se describen los componentes visuales utilizados.

3.5.2.1 Tablas

Una tabla es un contenedor que muestra la información concreta, breve y sustancial distribuida en filas y columnas. Esta característica conlleva a que en una pantalla pequeña mostrar los datos horizontalmente sería imposible sin desplazamiento horizontal en la página. Para solucionar este problema se definen dos técnicas descritas a continuación:

1. Tabla responsiva: Bootstrap 3.1 dentro de sus opciones para el trabajo con tablas contiene la clase **.table-responsive**, que convierte el contenedor donde se encuentra la tabla en una zona desplazable, confiriendo el desplazamiento horizontal de la página al contenedor de la tabla.

```
<div class="table-responsive">
  <table id="miTablaGestionarNomIdioma" class="table table-bordered ">
    <thead class="thead">
      <tr style="background-color: whitesmoke">
        <th>#</th>
        <th>Denominaci&ocute;n</th>
        <th>Abreviatura</th>
      </tr>
    </thead>
    <tbody id="tbody_idiomas">
    </tbody>
  </table>
</div>
```

Figura 18: Código de una table responsiva

2. Mostrar y ocultar columnas: se definen puntos de ruptura vía JavaScript en dependencia de la resolución del dispositivo y de la cantidad de columnas de la tabla. La Tabla 4 muestra las columnas que se ocultan en los requisitos que lo requieren:

Tabla 4: Requisitos de columnas a ocultar:

Requisito	Elemento	Columnas a ocultar
Gestionar roles	Tabla Principal	<ul style="list-style-type: none"> • Descripción • Abreviatura
Gestionar usuarios	Tabla principal	<ul style="list-style-type: none"> • Entidad • Área • Cargo • Tema • Idioma

		<ul style="list-style-type: none"> • Escritorio
	Tabla Asignar Roles	<ul style="list-style-type: none"> • Denominación
	Tabla Listar Sesiones	<ul style="list-style-type: none"> • Fecha • Hora • Rol
Gestionar campos del perfil de usuarios	Tabla principal	<ul style="list-style-type: none"> • Tipo de campo • Expresión • Visible • Longitud • Tipo de datos
Gestionar perfil de usuario	Tabla principal	<ul style="list-style-type: none"> • Email • Servidor • Años de experiencia • Fecha de activación • Puerto • Alias
Gestionar servidor	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar servidores de gestores de BD	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar nomenclador de gestor de base de datos	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar nomenclador de expresiones	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar nomenclador de escritorio	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar nomenclador de claves	Tabla principal	<ul style="list-style-type: none"> • Cantidad de históricos • Alfabético • Signos • Numérico
Gestionar nomenclador de tipo de conexión	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar acciones	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar servicios que consume	Tabla principal	<ul style="list-style-type: none"> • Protocolo • Proceso
Gestionar funciones	Tabla principal	<ul style="list-style-type: none"> • Descripción
Gestionar parámetros	Tabla principal	<ul style="list-style-type: none"> • Valor por defecto • Es null

3.5.2.2 Árboles

Los árboles son componentes que muestran listas anidadas de elementos, en el caso de la personalización de Acaxia se utiliza el plugin **jsTree** (36) basado en jQuery que permite construir, dar estilo, y realizar acciones sobre las listas anidadas de HTML5.

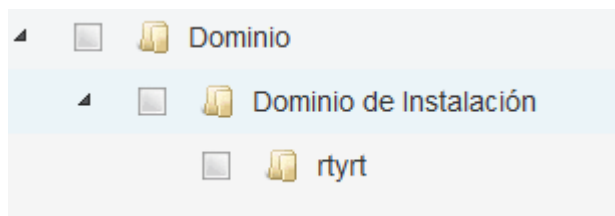


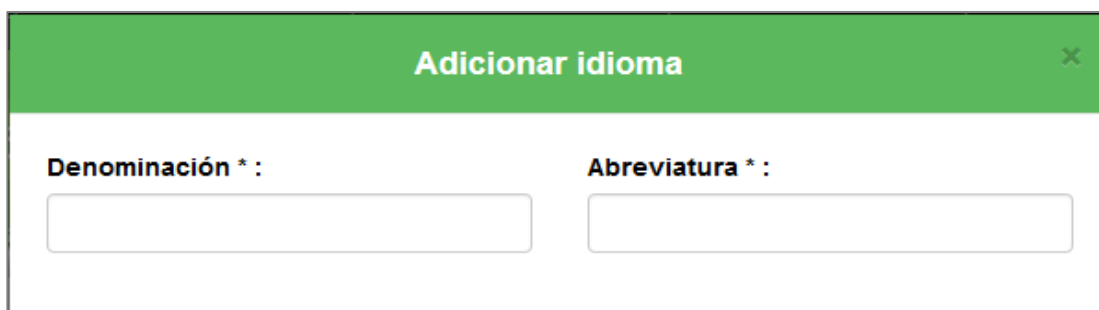
Figura 19: Ejemplo de árbol de dominios.

3.5.2.3 Campos de texto

Los campos de texto en HTML5 son construidos mediante la etiqueta `<input>`, dentro de sus atributos fundamentales se encuentra `type=""`, que especifica qué tipo de campo de texto es. Para lograr un mismo estilo y efecto en cualquier navegador y dispositivo, estos van acompañados de las diferentes clases que Bootstrap 3.1 ofrece. En la codificación se puede apreciar cómo se construye, estiliza y ubica un campo de texto dentro de un formulario.

```
<div class="row">
  <div class="col-sm-6" >
    <div class="form-group">
      <label for="inputAddNomIdiomDenom" style="color: black">Denominaci&ocirc;n * :</label>
      <input type="text" class="form-control" id="inputAddNomIdiomDenom" required>
      <span class="help-block with-errors"></span>
    </div>
  </div>
  <div class="col-sm-6">
    <div class="form-group" >
      <label for="inputAddNomIdiomAbbr" style="color: black">Abreviatura * :</label>
      <input type="text" class="form-control" id="inputAddNomIdiomAbbr" required>
      <span class="help-block with-errors"></span>
    </div>
    <br>
  </div>
</div>
```

Figura 20: Código de campos de texto.



The image shows a modal dialog box with a green header containing the title "Adicionar idioma" and a close button (X). Below the header, there are two text input fields. The first is labeled "Denominación *:" and the second is labeled "Abreviatura *:". Both fields are empty and have a light gray border.

Figura 21: Ejemplo de campos de texto.

3.5.2.4 Vínculos y Botones

Es necesario que por conceptos de visualización y estilo varios vínculos y los botones luzcan de igual manera, así como que se adapten al espacio que requiera la pantalla de un dispositivo. En la personalización de Acaxia se utiliza la clase `.btn` de Bootstrap 3.1 que proporciona un estilo único a los elementos de tipo vínculo (`<a>...`) y botón (`<button>...</button>`):

```
<button type="button" class="btn btn-danger" data-dismiss="modal">  
  <span class="glyphicon glyphicon-remove-circle"></span> &nbsp;&nbsp;&nbsp;Cancelar  
</button>
```

```
<a data-toggle="modal" href="#modalAddNomIdiomas" class="btn btn-success">  
  <span class="glyphicon glyphicon-plus-sign"></span>  
  &nbsp;&nbsp;&nbsp;Adicionar  
</a>
```

Figura 22: Código de botones

3.6 Pruebas y validaciones

Para la realización de las pruebas se siguieron un conjunto de técnicas que verifican y validan que la personalización del sistema cumple con el objetivo general de la investigación.

3.6.1 Técnica Guía de Observación

La guía de observación permitió listar una serie de eventos, procesos, hechos o situaciones a ser observados, su ocurrencia y características, ello es factible con base a un ejercicio de visión previo con miras a establecer los aspectos a observar. Se asocia generalmente con las interrogantes u objetivos específicos del estudio (37).

Se realizó una comparación íntegra entre el sistema de seguridad antes de la personalización y después de esta, evidenciando los cambios inherentes al Diseño Web Responsivo para la capa de presentación del sistema Acaxia.

3.6.1.1 Sistema Acaxia

Tabla 5: Comparación entre el sistema Acaxia y su personalización para dispositivos móviles:

Elemento	Acaxia	Personalización de Acaxia
Flexibilidad de las Imágenes	No flexible	Flexible
	Al observar el sistema desde un dispositivo móvil la imagen del logotipo no cambia de tamaño y en ocasiones se pierde parte de esta.	La imagen del logotipo cambia de tamaño al momento de minimizar el tamaño de la ventana o al visualizar el sistema desde distintos dispositivos móviles y cambia su aspecto en resoluciones más pequeñas.
Flexibilidad del marco	No flexible	Flexible
	Al momento de minimizar la ventana del navegador en una computadora de escritorio o laptop parte de la información y algunos elementos del sistema se cortan o se dejan de ver, además en ocasiones muestra una barra horizontal.	Al hacer más pequeña la ventana del navegador los elementos se van adaptando automáticamente al tamaño de esta.
Flexibilidad de la Imagen de background	No flexible	Flexible
	Se corta parte de la imagen.	Se redimensiona porcentualmente.
Resolución de imágenes al	Buena resolución	Buena resolución

cambiar el tamaño de la ventana del navegador	La imagen del Logotipo no cambia de tamaño por lo que siempre va a tener la misma resolución.	La imagen del logotipo cuenta con una buena resolución al cambiar de tamaño la ventana.
Se adapta el diseño a los diferentes tamaños de la ventana del navegador o dispositivos	No se adapta	Se adapta
	Al minimizar la ventana en una computadora de escritorio o laptop parte del contenido se va perdiendo, en el caso del acceso desde dispositivos sucede exactamente lo mismo.	Al ir cambiando el tamaño de la ventana el diseño se va adaptando al tamaño en el que está la ventana, los elementos se mueven de forma vertical sin dejar ningún elemento fuera del alcance del usuario.
Estructura del sitio web en los dispositivos	Estructura igual	Estructura distinta
	En resoluciones medianas (<i>Smartphone</i> y <i>Tablet</i>) algunos elementos se muestran recortados y se pierde parte del contenido y en resoluciones pequeñas (<i>IPad</i> y <i>Teléfonos móviles</i>) además de lo anterior se muestra una barra horizontal, es decir, se mantiene la misma estructura para todo tipo de dispositivos.	Al visualizar el sistema desde dispositivos móviles la estructura cambia el número de columnas.
Unidad de medidas	Pixeles	Porcentajes
	Las medidas que se colocaron en el CSS y el código que tiene el sistema se encuentra todo en pixeles.	La unidad de medida utilizada para este sistema son los porcentajes, que posibilita reajustar el tamaño de los

		componentes con respecto a su contenedor.
--	--	---

3.6.2 Resultado de la técnica Guía de Observación

Al analizar el Sistema Integral de Seguridad Acaxia en tres diferentes dispositivos se pudo observar que muestra una misma estructura para cada dispositivo, además al ir minimizando el tamaño del navegador buscando resoluciones menores algunos de los componentes que presenta desaparecen o cambian de lugar y muestra un desplazamiento horizontal que dificulta la navegación por el sistema.

Al analizar los cambios del sistema Acaxia se demostró que este cumple con las características de un Diseño Web Responsivo. Se puede ver el cambio de los elementos en cuanto a la jerarquía que se debe mantener, así como también se adaptan los contenedores de la información y los elementos al tamaño y capacidad del dispositivo en el cual se está observando el sistema.

3.6.3 Técnica de IADOV

Para evaluar la satisfacción de los especialistas del centro CEIGE con la personalización del sistema Acaxia se utilizó la Técnica de IADOV. La técnica facilitó determinar el nivel de satisfacción individual y grupal a partir de una encuesta elaborada según las exigencias de la solución propuesta (38), la encuesta fue aplicada a 10 especialistas del centro seleccionados como muestra para recoger su criterio, los mismos fueron elegidos teniendo en cuenta la experiencia que poseen en el trabajo con el sistema de seguridad Acaxia y en general con el marco de trabajo Sauxe. Los resultados individuales de la satisfacción se resumen en el gráfico que a continuación se presenta:

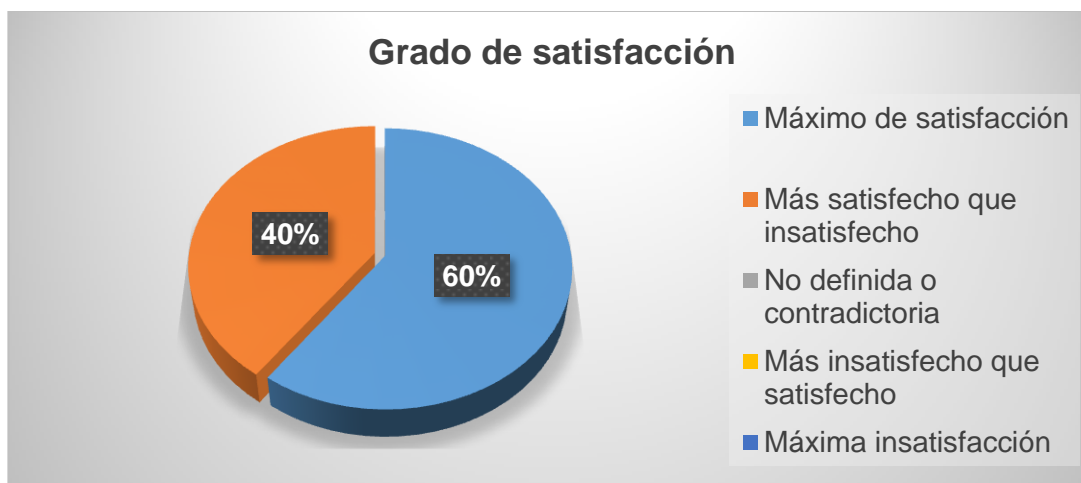


Figura 23: Satisfacción individual

Para obtener el Índice de Satisfacción Grupal (ISG) se parte de asociar los diferentes niveles de satisfacción de los encuestados, con una escala numérica que oscila entre +1 y - 1, como se muestra en la Tabla 6: Relación de la satisfacción individual con la escala de satisfacción.

Tabla 6: Relación de la satisfacción individual con la escala de satisfacción:

Relación de la satisfacción individual con la escala de satisfacción			
Escala	Significado	Satisfacción individual	%
+1	Máximo de satisfacción	6	60
0,5	Más satisfecho que insatisfecho	4	40
0	No definido y contradictorio	0	0
-0,5	Más insatisfecho que satisfecho	0	0
-1	Máxima insatisfacción	0	0

En esta investigación el Índice de Satisfacción Grupal fue: ISG = 0,8. Como se puede apreciar en la Figura 24, el valor del Índice es alto, lo que refleja aceptación de la personalización, en tanto los especialistas han emitido criterios donde evidencian su satisfacción por la construcción de la personalización del sistema Acaxia para su correcta visualización en dispositivos móviles.

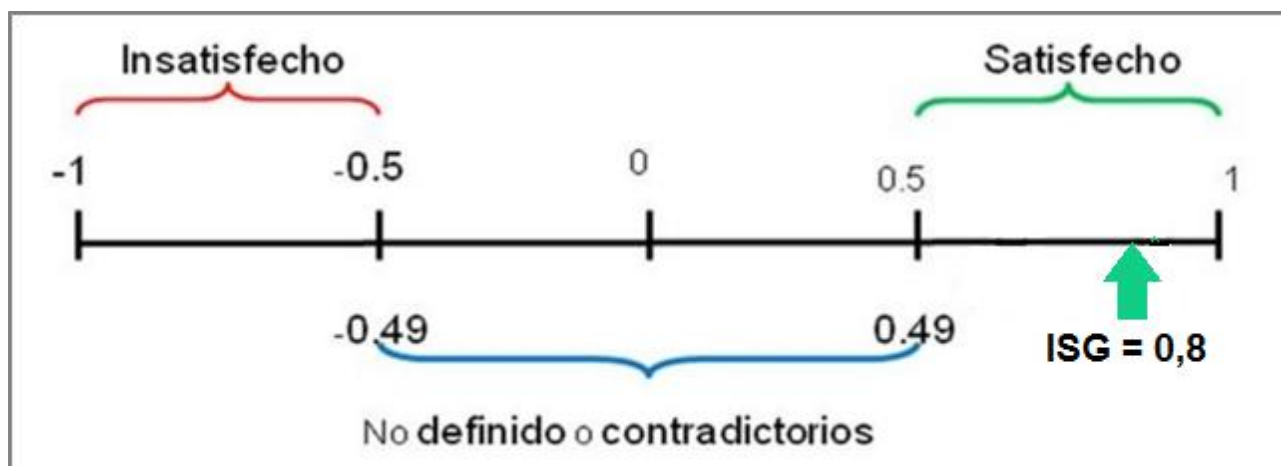


Figura 24: Intervalos de probabilidad de satisfacción grupal

La técnica de IADOV contempla además dos preguntas complementarias de carácter abierto. Estas son:

- Pregunta 4. A su consideración cómo cree que se incrementaría la utilización de la personalización del sistema Acaxia.
- Pregunta 6. ¿Qué opina usted acerca de que sería beneficioso para el centro CEIGE contar con la personalización del sistema de seguridad Acaxia para su correcta visualización en dispositivos móviles?

Con respecto a la utilización de la personalización del sistema de seguridad Acaxia algunas respuestas fueron:

- “Este sería el primer paso en dirección a la actualización de las tecnologías de los productos del centro para lograr su correcta visualización en dispositivos móviles. El empleo de estos productos, y por ende de Acaxia, aumentaría pues en la actualidad la tendencia es cada vez mayor al desarrollo de aplicaciones responsivas.”
- “Contaría de una facilidad palpable el hecho de poder interactuar con el sistema sin necesidad de estar sentado frente a una PC en el momento que sea necesario. Se gana en usabilidad y disponibilidad del sistema.”
- “Me gustaría que se extienda la solución a todos los sistemas informáticos que utilicen el componente de seguridad Acaxia, lo cual ayudaría a las entidades que tengan acceso a sus sistemas desde dispositivos móviles.”

- *“AL permitir su uso en dispositivos móviles habría diversidad de formas de utilización, permitiendo así un mayor uso de la aplicación.”*

Con relación al beneficio que la personalización traería para el centro CEIGE, algunas respuestas fueron:

- *“Acaxia y la base tecnológica en primer lugar y luego gradualmente sumar los productos del centro pues en la actualidad es cada vez más frecuente el diseño de este tipo de aplicaciones.”*
- *“Es un gran paso en el uso de las nuevas tecnologías. Se obtiene una aplicación mucho más accesible.”*
- *“Si sería beneficioso ya que el centro tendría posibilidades de acceso al sistema desde distintos dispositivos.”*
- *“Permitiría diversificar soluciones al involucrar tecnologías móviles debido al creciente uso de las mismas.”*

3.6.3.1 Resultado de la técnica de IADOV

El proceso de validación mediante la Técnica de IADOV del cuestionario a los especialistas que trabajan en el centro CEIGE, confirmó la factibilidad de uso de la personalización del sistema, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG = 0.8) y cualitativamente en los criterios emitidos donde evidencian su satisfacción por el sistema personalizado, lo que refleja aceptación de la solución y un reconocimiento a su utilidad.

3.6.4 Pruebas de usabilidad

Según la ISO 9241 la usabilidad es el grado en el cual un producto puede ser utilizado por usuarios específicos para alcanzar metas propuestas con eficiencia, eficacia, y satisfacción en un contexto específico de uso (39).

Evaluación heurística

Es un método de inspección de usabilidad para el software que ayuda a identificar problemas de usabilidad en la interfaz de usuario, de diseño específicamente. Se hace una inspección minuciosa a interfaces o sistemas con el fin de determinar si cada uno de sus elementos cumple con objetivos del sistema. Las evaluaciones heurísticas normalmente se llevan a cabo por un grupo pequeño de evaluadores (suelen ser de uno a tres). Se utilizaron para realizar las pruebas

de usabilidad a la personalización del sistema, la evaluación de expertos (40) y la heurística de Nielsen (41), tomando como guía la lista de chequeo que se muestra en el Anexo 6, para la aplicación de este método se realizaron dos iteraciones con el cliente. A continuación se muestra la Tabla 7 con el análisis de las No conformidades por iteración:

Tabla 7: Análisis de las no conformidades por iteración:

Iteración	Cantidad de No conformidades	Relacionadas con
Primera	4	<ul style="list-style-type: none">• Errores ortográficos en el contenido.• Se puede navegar en el sitio sin previa autenticación.• No existen puntos de salida que permita al usuario abandonar la tarea actual en la que se encuentre.• Algunos íconos no se corresponden con las tareas a realizar.
Segunda	0	

3.6.5 Resultado de las pruebas de usabilidad

El método de inspección de usabilidad para la personalización del sistema de seguridad Acaxia permitió identificar 4 problemas de usabilidad en la interfaz de usuario, de diseño específicamente, los cuales fueron erradicados inmediatamente dejando la solución lista para ser utilizada.

3.6.6 Pruebas con teléfonos móviles y emuladores

Las pruebas realizadas con teléfonos móviles reales comprueban que el sistema es accesible mediante conexión inalámbrica y que cumple con las pautas de diseño para sistemas web responsivos (descritas en el epígrafe [2.3.1](#)). Estas pruebas permiten analizar la correcta visualización de la personalización en cualquier dispositivo, comprobándose el funcionamiento y disposición de sus elementos en pantalla.

Para lograr la completitud de las pruebas con dispositivos móviles se utilizan emuladores, los cuales ofrecen la posibilidad de probar el sistema en diferentes entornos. Las pruebas son desarrolladas haciendo uso de siete emuladores:

- Android Emulator
- iPhone Emulator
- Ipad Emulator
- BlackBerry Emulator
- Tablet Emulator
- The Responsinator
- Responsive Web Design Testing Tool

3.6.7 Resultado de pruebas con teléfonos móviles y emuladores

La aplicación de las pruebas realizadas con teléfonos celulares reales y emuladores, validaron que la personalización de la capa de presentación del sistema de seguridad Acaxia reconoce y se adapta automáticamente al tipo de navegador con que se está accediendo, mostrando la interfaz correspondiente en cada caso. Por otra parte, en todos los celulares en que se probó el sistema el contenido se visualiza de forma correcta. Además el sistema permite la navegación sin tener que utilizar desplazamiento horizontal de la página debido a la capacidad de adaptarse a las dimensiones del navegador que posee la versión móvil.

3.6.8 Acta de validación

Luego de la revisión técnica del cliente a la personalización de la capa de presentación del sistema de seguridad Acaxia, este determina mediante un acta de validación mostrada en la Figura 25 que la solución es estable y está lista para su utilización.



UCI
Informáticas

CENTRO DE INFORMATIZACIÓN DE ENTIDADES
DEPARTAMENTO DE DESARROLLO DE COMPONENTES

FECHA 12/06/2015

A quien pueda interesar:

Por este medio se hace constar que la solución Personalización de la capa de presentación del sistema de seguridad Acaxia para su correcta visualización en dispositivos móviles de los autores Fredy Ulloa Morffy y Reinel Leonard Garcia fue sometida a una revisión técnica en la cual se detectaron 4 no conformidades que fueron resueltas quedando esta solución estable y lista para su posterior uso.

Como parte del desarrollo de la solución se elaboraron y entregaron los siguientes artefactos:

1. Descripción de la metodología
2. Bocetos de baja fidelidad (6)
3. Prototipos de interfaz de usuario de baja fidelidad (6)
4. Prototipos de interfaz funcional (6)
5. Documento oficial de la investigación
6. Código fuente

Para que así conste firman a continuación los miembros del equipo que realizó la revisión, el autor y los tutores del trabajo.

Dado a los 12 días del mes de Junio de 2015.

Nombre y apellidos	Firma
Revisores: <u>Inoelkis Delacruz Osorio</u>	
Autores: <u>Reinel Leonard Garcia</u> <u>Fredy Ulloa Morffy</u>	 
Tutores: <u>Olga Yarisbel Rojas Grass</u> <u>Yenny Vanega Hechanarria</u>	 

J' Departamento de Desarrollo de Componentes
René R. Bauta Camejo 

Figura 25: Acta de validación del cliente

3.7 Conclusiones

En el presente capítulo se describen los resultados de la implementación de la capa de presentación de Acaxia. Se muestran los estándares de codificación utilizados para la construcción del sistema, así como las características de los componentes visuales implementados. Se constató la validez de la propuesta que permitió obtener la personalización de la capa de presentación del sistema de seguridad Acaxia. Además se realizaron las pruebas y validaciones a la personalización de la capa de presentación del sistema Acaxia, donde se pudo verificar el correcto funcionamiento de la personalización.

Conclusiones generales

Se desarrollaron las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- La identificación y caracterización de las tecnologías para el diseño multidispositivo, permitieron seleccionar el Diseño Web Responsivo por su capacidad de adaptarse automáticamente a cualquier resolución de pantalla, y el marco de trabajo Bootstrap para la personalización de la capa de presentación del sistema Acaxia por la agilidad y rapidez con que cuenta, facilitando el diseño de interfaces para sistemas responsivos.
- El diseño visual para la personalización de la capa de presentación del sistema de seguridad Acaxia permitió realizar una reestructuración adecuada de toda la información a partir de las pautas definidas.
- Se obtuvo una correcta visualización de la capa de presentación del sistema de seguridad Acaxia en diferentes dispositivos a través del desarrollo de la solución.
- Se verificó la correcta visualización de la capa de presentación del sistema de seguridad Acaxia para diferentes dispositivos a partir de la aplicación de pruebas de usabilidad y aceptación. Se evidenció la satisfacción de los usuarios a partir del resultado obtenido en la técnica de ladov.

Recomendaciones

Realizar la integración de la personalización de la capa de presentación para su correcta visualización en dispositivos móviles con la capa de negocio y la capa de acceso a datos completando la personalización del sistema de seguridad Acaxia.

Referencias bibliográficas

1. Cabrera Campos, Alcides. *El diseño metodológico de la investigación científica*. La Habana : s.n., 2012.
2. Definicion.de. [En línea] 2008. [Citado el: 15 de enero de 2015.] <http://definicion.de/movil/#ixzz3VCFzLMVp>.
3. Telefonía Móvil. [En línea] 2012. [Citado el: 15 de enero de 2015.] <http://telefoniamoviltics.blogspot.com/p/definicion.html>.
4. Acosta Ramirez, Elvia Patricia. TICSMIELCA. *TECNOLOGÍAS MÓVILES*. [En línea] [Citado el: 12 de enero de 2015.] <https://sites.google.com/site/ticsmielca/home/tecnologias-moviles>.
5. Diccionarios.com. [En línea] 2011. [Citado el: 16 de enero de 2015.] [http://www.diccionarios.com/detalle.php?palabra=personalizacion&Buscar.x=0&Buscar.y=0&dicc_100=on&palabra2=.](http://www.diccionarios.com/detalle.php?palabra=personalizacion&Buscar.x=0&Buscar.y=0&dicc_100=on&palabra2=)
6. Ecured. *Arquitectura en capas*. [En línea] 2010. [Citado el: 20 de enero de 2015.] http://www.ecured.cu/index.php/Arquitectura_en_Capas.
7. Amo, Francesc. marketingOnline. *Diseño web Responsive o web Adaptive | Diferencias*. [En línea] 19 de noviembre de 2014. [Citado el: 7 de febrero de 2015.] http://blog.g4marketingonline.com/dise%C3%B1o-web-responsive-o-web-adaptive-diferencias?__.
8. WebDesign Cuba. *Gumby Framework*. [En línea] 31 de octubre de 2014. [Citado el: 10 de febrero de 2015.] <http://www.webdesigncuba.net/gumby-framework>.
9. Mendoza, Ivan E. ivanmendoza. *Foundation vs Bootstrap:Round 1*. [En línea] 26 de enero de 2012. [Citado el: 10 de febrero de 2015.] <http://ivanmendoza.net/disenio-web-2/foundation-vs-bootstrap-round-1>.
10. eSandra. eSandra online technology. *Bootstrap 3: ¿Vale La Pena Usar un Framework?* [En línea] 1 de octubre de 2014. [Citado el: 13 de febrero de 2015.] <http://www.esandra.com/search/Bootstrap+3%3A+%C2%BFVale+La+Pena+Usar+un+Frame+work%3F>.

11. Beati, Hernan. slideshare. *Hacia una Metodología de Diseño Web Responsive*, Hernán Beati. [En línea] 3 de diciembre de 2013. [Citado el: 5 de mayo de 2015.]
<http://es.slideshare.net/hbeati/hacia-una-metodologa-de-diseo-web-responsive>.
12. Wroblewski, Luke. Lukew ideation + design. *Mobile First*. [En línea] 03 de noviembre de 2009. [Citado el: 25 de febrero de 2015.] <http://www.lukew.com/ff/entry.asp?933>.
13. MDN, Mozilla Developer Network. HTML5. [En línea] 10 de abril de 2015. [Citado el: 25 de abril de 2015.] <https://developer.mozilla.org/es/docs/HTML/HTML5>.
14. Pérez Valdés, Damian. Maestro del web. *¿Qué es Javascript?* [En línea] 3 de julio de 2007. [Citado el: 17 de marzo de 2015.]
<http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>.
15. ProgramacionWeb.net. [En línea] 2009. [Citado el: 27 de enero de 2015.]
<http://www.programacionweb.net/cursos/curso.php?num=2..>
16. Reino Romero, Alfredo. Introducción a XML en castellano. [En línea] 26 de enero de 2000. [Citado el: 17 de febrero de 2015.] <http://www.asptutor.com/zip/xml.pdf>.
17. JSON. *Introducción a JSON*. [En línea] 2011. [Citado el: 7 de marzo de 2015.]
<http://www.json.org/json-es.html>.
18. W3C. World Wide Web Consortium. *Guía breve de CSS*. [En línea] [Citado el: 15 de enero de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>.
19. del Pino Moraguez, Abel Alejandro y Rivas Franco, Andy José. *Aplicación de los enfoques del Diseño Centrado en el Usuario (DCU) a la plataforma SO3*. La Habana : s.n., 2012. Tesis de grado.
20. Prieto, Ricardo. SILO CREATIVO. *Crear Mockups Online con Moqups*. [En línea] 02 de septiembre de 2013. [Citado el: 17 de mayo de 2015.]
<http://www.silocreativo.com/2013/09/herramienta-mockups-online-moqups/>.
21. E.U. de Ingeniería Técnica Informática de Oviedo. Sitio web de la E.U. de Ingeniería Técnica Informática de Oviedo. [En línea] [Citado el: 21 de marzo de 2014.]
<http://petra.euitio.uniovi.es/>.

22. Oracle Corporation. NetBeans. [En línea] [Citado el: 14 de enero de 2014.] <http://netbeans.org/community/releases/80/>.
23. Orozco, David. Concepto definición. *Definición de Firefox*. [En línea] 14 de septiembre de 2011. [Citado el: 20 de enero de 2015.] <http://conceptodefinicion.de/firefox/>.
24. Ciberaula. Ciberaula. *Una introduccion a APACHE*. [En línea] 2010. [Citado el: 24 de enero de 2015.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
25. Lessin, Jorge. jorgelessin. *Qué es Bootstrap y como funciona en el diseño web*. [En línea] 6 de noviembre de 2013. [Citado el: 15 de febrero de 2015.] <http://jorgelessin.com/que-es-bootstrap-y-como-funciona-en-el-diseno-web/>.
26. *Manual de JQuery*. Álvarez, Miguel Angel. 2009.
27. W3C. W3C Mobile Web Initiative. *Mobile Web Best Practices (MWBP) Flipcards*. [En línea] 2011. [Citado el: 19 de marzo de 2015.] http://www.w3.org/2007/02/mwbp_flip_cards.
28. Google. Google Webmaster Central Blog. [En línea] 30 de abril de 2012. [Citado el: 27 de marzo de 2015.] <http://googlewebmastercentral.blogspot.com/>.
29. Twitter. Bootstrapdocs. *CCS*. [En línea] 30 de enero de 2014. [Citado el: 14 de Abril de 2015.] <http://bootstrapdocs.com/v3.1.1/docs/css/>.
30. ALEGSA. Diccionario de Informática y Tecnología. *Definición de Formulario web, Leandro Alegsa*. [En línea] [Citado el: 10 de abril de 2015.] <http://www.alegsa.com.ar/Dic/formulario%20web.php>.
31. Wroblewski, Luke. Lukew ideation + design. *Multi-Device Layout Patterns*. [En línea] 14 de marzo de 2012. [Citado el: 10 de abril de 2015.] <http://www.lukew.com/ff/entry.asp?1514>.
32. Osmani, Addy. *Learning JavaScript Design Patterns*. [ed.] Mary Treserler. 1ra edición. Sebastopol, CA : O'REILLY MEDIA, Inc, 2012.
33. WordPress. WordPress Core Handbook. *HTML Coding Standards*. [En línea] 2013. [Citado el: 17 de mayo de 2015.] <https://make.wordpress.org/core/handbook/coding-standards/html/>.

34. —. WordPress Core Handbook. *CSS Coding Standards*. [En línea] 2012. [Citado el: 17 de mayo de 2015.] <https://make.wordpress.org/core/handbook/coding-standards/css/>.
35. jQuery. jQuery foundation. *JavaScript Style Guide*. [En línea] [Citado el: 17 de mayo de 2015.] <http://contribute.jquery.org/style-guide/js/>.
36. jQuery Foundation. jsTree jQuery tree plugin. *What is jsTree*. [En línea] [Citado el: 25 de mayo de 2015.] <http://www.jstree.com/>.
37. Becerra, Omar E. *Elaboración de instrumentos de investigación*. Caracas, Venezuela : s.n., 2012.
38. López Rodríguez, Alejandro; González Maura, Viviana. *La técnica de IADOV*. 47, Buenos Aires, Argentina : s.n., 2002.
39. Arosemena, Karla. Slideshare. *Conceptos de usabilidad*. [En línea] 26 de noviembre de 2009. [Citado el: 12 de mayo de 2015.] <http://es.slideshare.net/gikarl/usabilidad-conceptos-bsicos>.
40. Márquez Correa, Joaquín. *Guía para evaluación experta*. [En línea] 2003. [Citado el: 12 de mayo de 2015.] http://www.jmarquez.com/documentos/jm_checklist.pdf.
41. Nielsen, Jakob. Nielsen Norman Group. *10 Usability Heuristics for User Interface Design*. [En línea] 1 de enero de 1995. [Citado el: 12 de mayo de 2015.] <http://www.nngroup.com/articles/ten-usability-heuristics/>.

Análisis bibliométrico

La bibliografía debe ser la actualizada, representativa nacional e internacional, de fuentes de reconocida calidad; así como escritas y referenciadas utilizando alguna de las normas internacionalmente reconocidas. A continuación se muestra un análisis de la bibliografía consultada:

Tabla 8: Análisis bibliométrico:

Análisis bibliométrico					
Tipo de referencia bibliográfica	Cantidad por cada tipo	Idioma inglés	Idioma español	Últimos 5 años	Años anteriores
Libros	3	1	2	2	1
Tesis	1		1	1	
Sitios web	33	10	23	18	15
Artículos en revistas referenciadas en la Web	1		1		1
Documentos publicados en la web	2		2		2
Informes	1		1	1	

El análisis bibliométrico proporcionó como resultado que el trabajo de diploma cuenta con más de 35 referencias bibliográficas consultadas, más del 40% de las mismas son de los últimos cinco años y de ellas la mitad están en idioma inglés. Ninguna utiliza bibliografía de sitio no arbitrado, cumplen con las normas establecidas y existen referencias a libros, sitios web, tesis de grado, artículos, documentos e informes. Cumpliendo con el procedimiento guía para los trabajos de diploma en la Universidad de las Ciencias Informáticas.

Glosario de términos

Diseño: Proceso de disponer, estructurar y conformar un objeto o conjunto de información para que cumpla un cometido conforme a los medios disponibles para cumplirlo.

Multidispositivo: involucrar, o tener múltiples dispositivos.

PHP: Hypertext Preprocessor. Lenguaje de script diseñado para la creación de páginas web activas, multiplataforma (puede correr en Windows, Mac, Linux). Usualmente se usa en conjunto con la base de datos MySQL, pero puede usar cualquier otro tipo de base de datos como por ejemplo Oracle, SQL o Postgres.

ExtJs: es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX y DOM.

Resolución: número máximo de píxeles que se ven en una pantalla.

Sistema operativo: programa que administra los demás programas en una computadora.

Navegador: aplicación para visualizar todo tipo de información y navegar por internet. Programa que interpreta el código (HTML y más) en el que están escritas las páginas web y las muestra tal y como se ven en el monitor.

Navegabilidad: la navegabilidad o navegabilidad web es la facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sistema web.

Web: World Wide Web o WWW. Red mundial. Es la parte multimedia de internet, es decir, los recursos creados en HTML y sus derivados. Sistema de información global desarrollado en 1990 por Robert Cailliau y Tim Berners-Lee en el CERN (Consejo Europeo para la Investigación Nuclear). Con la incorporación de recursos gráficos e hipertextos, fue la base para la explosiva popularización de internet a partir de 1993.

Marcos de trabajo: plataforma, entorno, marco de trabajo. Desde el punto de vista del desarrollo de software, un marco de trabajo es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado.

URL: acrónimo de Uniform Resource Locator. Localizador Uniforme de Recurso. Es el sistema de direcciones en Internet. El modo estándar de escribir la dirección de un sitio específico o parte de una información en el Web.

Sistema web: permite cambiar información en el tiempo, haciendo y facilitando una comunicación más estrecha y dinámica con sus usuarios o clientes. Normalmente están apoyados con bases de datos que permiten almacenar información.

Aplicaciones: cualquier programa que corra en un sistema operativo y que haga una función específica para un usuario. Por ejemplo, procesadores de palabras, bases de datos, agendas electrónicas, entre otros.

XHTML: siglas del inglés eXtensible HyperText Markup Language. XHTML es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios y buscar errores, entre otras funciones.

DOM: siglas del inglés Document Object Model (Modelo de Objetos de Documento) es un API para documentos HTML y XML.

API: siglas del inglés Application Programming Interface. Interfaz de Programación de Aplicaciones. Una serie de rutinas usadas por una aplicación para gestionar generalmente servicios de bajo nivel, realizados por el sistema operativo de la computadora. Uno de los principales propósitos de un API consiste en proporcionar un conjunto de funciones de uso general, de esta forma los programadores se benefician de las ventajas del API, ahorrándose el trabajo de programar todo de nuevo.

HTTP: en inglés Hypertext Transfer Protocol. Protocolo de Transferencia de Hipertexto. HTTP es un protocolo con la ligereza y velocidad necesaria para distribuir y manejar sistemas de información hipermedia. HTTP ha sido usado por los servidores World Wide Web desde su inicio en 1993.

W3C: el World Wide Web Consortium (W3C) es un consorcio internacional en donde organizaciones, los usuarios y empleados de tiempo completo desarrollan estándares y especificaciones relacionados al WWW. Se creó en 1994 por Tim Berners-Lee, inventor del World Wide Web y otros.

One Web: hace referencia a la idea de construir una web para todos (Web for All) y accesible desde cualquier tipo de dispositivo (Web on Everything).

Interfaz: es el punto de conexión ya sea entre dos componentes de hardware, dos programas o entre un usuario y un programa.

GUI: Interfaz Gráfica de Usuario, en inglés Graphic User Interface. Componente de una aplicación informática que el usuario visualiza gráficamente, y a través de la cual opera con ella. Está formada por ventanas, botones, menús e iconos, entre otros elementos.

RGB: es un modelo de color (Red Green Blue) utilizado normalmente para presentar color en los sistemas de video, cámaras, y monitores de ordenadores. Representa todos los colores como combinaciones de rojo, verde y azul.

RGBA: se trata del mismo modelo RGB, pero con otra propiedad, canal alfa. Este canal se usa como un índice de la transparencia en un píxel. Esto nos sirve a la hora de mezclar varios colores designados para un solo píxel.

Anexos

Anexo 1: Prototipos de baja fidelidad de las vistas principales

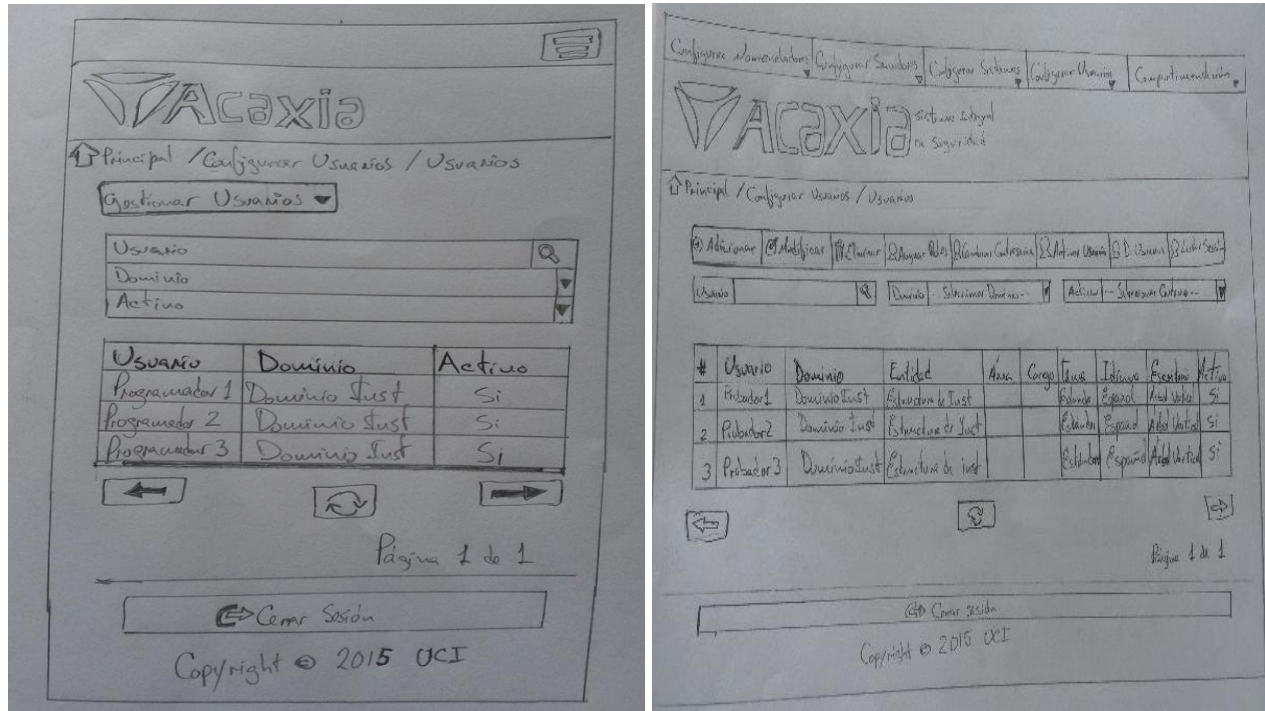


Figura 26: Bocetos de la vista general de procesos

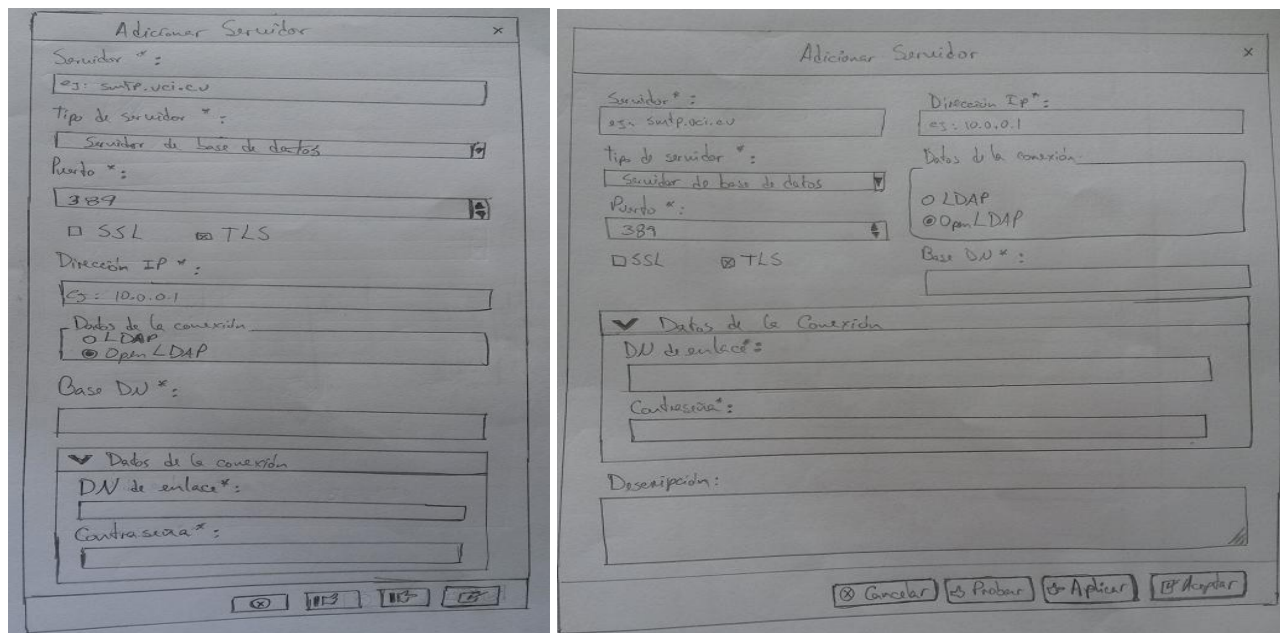


Figura 27: Bocetos de las vistas de formularios

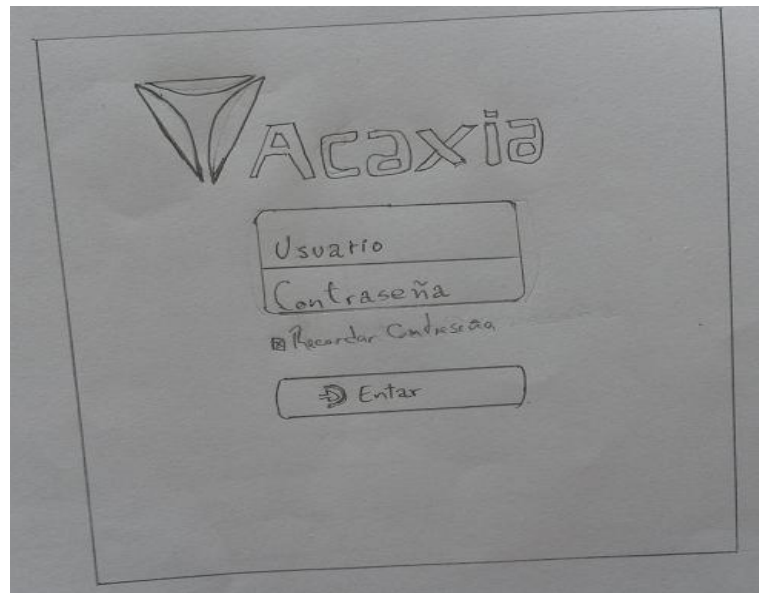


Figura 28: Boceto de la vista de autenticación

Anexo 2: Prototipos de interfaz de usuario de las vistas principales

Vista de autenticación



Figura 29: Prototipos de interfaz de la Vista de autenticación para dispositivos móviles.

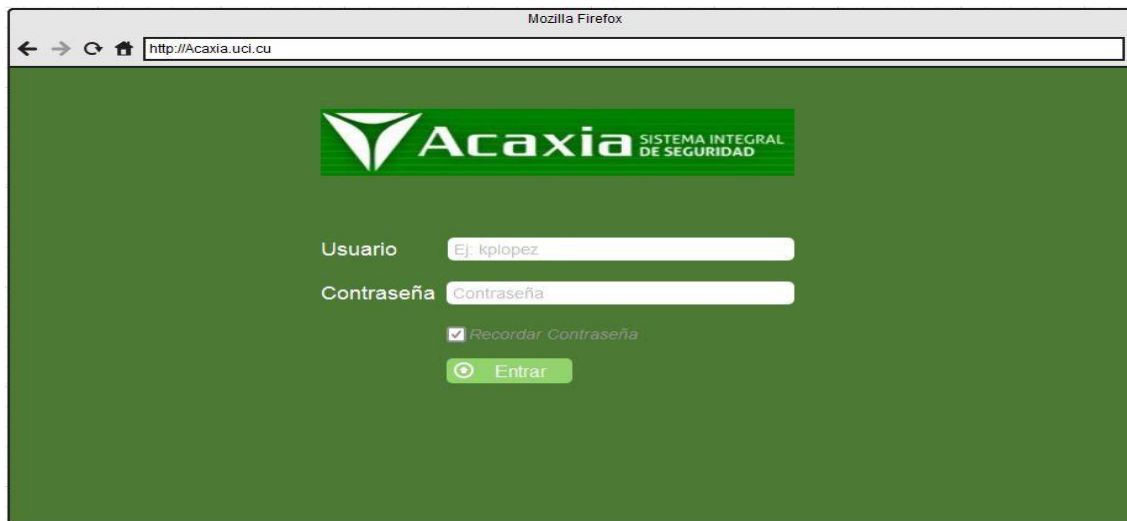


Figura 30: Prototipo de interfaz de la Vista de autenticación para escritorio.

Vista general de gestión de procesos



Figura 31: Prototipos de interfaz de la Vista general de gestión de procesos para dispositivos móviles.

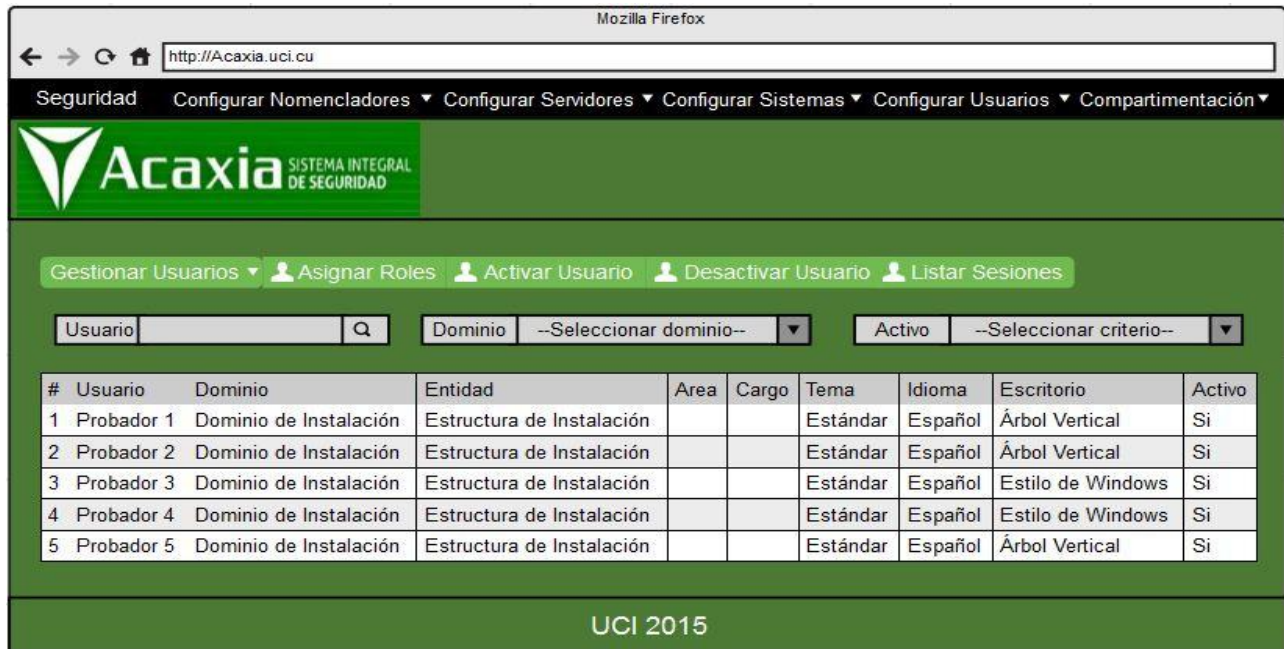


Figura 32: Prototipo de interfaz de la Vista general de gestión de procesos para escritorio.

Anexo 3: Prototipos funcionales de las vistas principales



Figura 33: Interfaz funcional de la vista de autenticación.

Figura 34: Interfaz funcional de la vista de formularios.

#	Usuario	Dominio	Entidad	Área	Cargo	Tema	Idioma	Escritorio	Activo
1	Probador 1	Dominio de instalación	Estructura de instalación			Estándar	Español	Árbol vertical	SI

Figura 35: Interfaz funcional de la vista general de procesos.



Figura 36: Interfaz funcional de la vista general de procesos en dispositivo móvil.

Anexo 4: Técnica de IADOV

Cuestionario de los especialistas que trabajan con Acaxia

Estimado especialista:

Lea con cuidado cada pregunta antes de responder. En este cuestionario NO DEBE PONER SU NOMBRE, marque en cada pregunta con una X en una sola opción y en el caso de las dos preguntas abiertas responda brevemente. Le agradecemos su participación y franqueza al responder honestamente lo que piensa en las siguientes preguntas:

1. ¿Cómo es el nivel de aceptación de usted hacia las tecnologías móviles?

Muy bajo__ Bajo__ Medio__ Alto__ Muy Alto__

2. ¿Usted considera que el sistema de seguridad Acaxia que utilizan actualmente garantiza la correcta visualización de sus contenidos en cualquier resolución de pantalla de dispositivos?

Si__ No__ No sé__

3. ¿Cree usted que sería factible contar con una solución para el sistema de seguridad Acaxia que implemente la correcta visualización de sus contenidos en dispositivos móviles?

Si__ No__ No sé__

4. A su consideración cómo cree que se incrementaría la utilización de la personalización del sistema Acaxia.

5. Luego de haber visto la personalización del sistema Acaxia refleje en qué medida le gusta la solución desarrollada.

__Me gusta mucho __Me disgusta más de lo que me gusta

__Me gusta más de lo que me disgusta __No me gusta nada

__Me da lo mismo __No sé decir

6. ¿Qué opina usted acerca de que sería beneficioso para el centro CEIGE contar con la personalización del sistema de seguridad Acaxia para su correcta visualización en dispositivos móviles?

Cuadro lógico de IADOV

Tabla 9: Cuadro lógico de IADOV:

5. Luego de haber visto la personalización del sistema Acaxia refleje en qué medida le gusta la solución desarrollada.	2. ¿Usted considera que el sistema de seguridad Acaxia que utilizan, garantiza la correcta visualización de sus contenidos en cualquier resolución de pantalla de dispositivos?		
	No	No Se	Si

	3. ¿Cree usted que sería factible contar con una solución para el sistema de seguridad Acaxia que implemente la correcta visualización de sus contenidos en dispositivos móviles?								
	Si	No se	No	Si	No se	No	Si	No se	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	6	6	5
No sé decir	2	3	6	3	3	3	6	6	4

Fórmula para calcular el índice de satisfacción general.

$$ISG = \frac{A(+1)+B(+0.5)+C(0)+D(-0.5)+E(-1)}{N}$$

En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 o 6; 4; 5 y donde N representa el número total de sujetos del grupo, que en el caso de la investigación se traduce de la siguiente manera:

$$ISG = \frac{6*(1)+4*(0.5)+0}{10} = 0.80$$

Anexo 5: Pruebas de usabilidad

Tabla 10: Descripción de la lista de chequeo:

Descripción	
Evidencia ¿Qué?	Es la descripción de qué se quiere verificar
Procedimiento	Especifica el procedimiento mediante el cual se pueden hacer las

¿Cómo?	verificaciones.
Respuesta	Se selecciona SI o NO en dependencia de si la respuesta a la evidencia es negativa o positiva.
Observaciones	Hallazgo encontrado

Tabla 11: Lista de chequeo utilizada para las pruebas de usabilidad:

Pruebas de Usabilidad			
Evidencia ¿Qué?	Procedimiento ¿Cómo?	Respuesta	Observaciones
Aspectos generales			
¿Cumple el sistema con sus objetivos?	Aplicación web		
¿Está diseñado para dar a los usuarios lo que ellos quieren?	Aplicación web		
¿Es eficiente e intuitivo?	Aplicación web		
¿Mantiene una consistencia tanto en su funcionamiento como en su apariencia?	Aplicación web		
¿Facilita que el usuario se sienta cómodo y con el control del sistema?	Aplicación web		
¿La presentación, composición y formularios del sistema en general se visualiza sin problemas en Opera, Firefox, Safari y Explorer?	Aplicación web		
¿Se visualiza correctamente el sistema tanto en dispositivos móviles como en computadoras de escritorio o laptop?	Aplicación web		

Marca			
¿Muestra la home page la naturaleza del sistema y se identifica bien el logotipo o marca?	Aplicación web		
¿Aparecen elementos del Logo o de la imagen que lo identifica en todas las páginas?	Aplicación web		
¿Aparece el logo en un lugar importante dentro de la página?	Aplicación web		
Navegación			
¿Aparece la navegación en un lugar prominente, donde se vea fácilmente?	Aplicación web		
¿Se puede navegar en el sistema sin autenticación?	Aplicación web		
¿Tiene el sistema buscadores para facilitar el acceso a los contenidos?	Aplicación web		
¿Se mantiene una navegación consistente y coherente en el sistema?	Aplicación web		
¿Existen elementos que permitan al usuario saber exactamente dónde se encuentra dentro del sistema y cómo volver atrás (breadcrumbs)?	Aplicación web		
¿Indican los enlaces claramente hacia dónde apuntan? ¿Está claro lo que el usuario encontrará detrás de cada uno?	Aplicación web		

¿Existe en cada página puntos de salida que permita al usuario abandonar la tarea actual en la que se encuentre?	Aplicación web		
¿El usuario tiene la opción de “Salir” para abandonar el sistema en cualquier momento, en caso de que este lo requiera?	Aplicación web		
Imágenes			
¿Se ha optimizado el uso de las imágenes para reducir el tamaño?	Aplicación web		
¿Poseen las páginas animaciones innecesarias?	Aplicación web		
Banner			
¿El banner se ubica en la parte superior del sistema?	Aplicación web		
Contenidos			
¿Es coherente el contenido con el contexto de la página del sistema?	Aplicación web		
¿La redacción es corta y precisa?	Aplicación web		
¿El mensaje de error es entendible e informa cuáles son las acciones correctoras?	Aplicación web		
¿Se dan ejemplos para facilitar la tarea?	Aplicación web		
Interfaz			
¿Tiene el sistema una interfaz	Aplicación web		

amigable, con colores que concuerden con los objetivos y propósitos del mismo?			
¿Hay espacios libres entre el contenido, para descansar la vista?	Aplicación web		
¿Es una interfaz limpia y sin ruido visual?	Aplicación web		
Otros			
¿La aplicación permite completar tareas introduciendo solamente los datos obligatorios?	Aplicación web		
¿Cuándo se selecciona un elemento se diferencia de los no seleccionados?	Aplicación web		
¿Los íconos son visualmente y conceptualmente distintos pero mantienen una armonía?	Aplicación web		
¿Se usa la misma fuente para todos los navegadores?	Aplicación web		
¿Los íconos se corresponden con las tareas a realizar?	Aplicación web		
¿En todos los casos el idioma definido por el sistema se empleará en todos los textos? ¿Si es español no se muestran palabras en inglés u otro, y de igual forma para los demás?	Aplicación web		
¿Existe suficiente espacio entre los	Aplicación web		

elementos de acción (links, botones, etc.) para prevenir que el usuario haga click en el elemento incorrecto?			
¿El nombre de los botones de un formulario es adecuado, aplicado a la acción, no general (Ej.: utilizar “Aceptar” en vez de “OK”...)?	Aplicación web		