



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO. Una herramienta de portabilización de aplicaciones Web 1.0 a Web Mobile

AUTORES. Mauricio Javier Witkin, Ramón Maximiliano Serrano

DIRECTOR. Dr. José Matías Rivero

CODIRECTOR. Dr. Matias Urbieto

ASESOR PROFESIONAL.

CARRERA. Licenciatura en Sistemas, plan 2015

Resumen

La masificación de los dispositivos móviles permite a los usuarios acceder a una gran variedad de aplicaciones desarrolladas sin haber tenido en cuenta que iban a ser accedidas desde dispositivos móviles. Como consecuencia el usuario enfrenta problemas de usabilidad al navegar sitios Web que no tuvieron en cuenta estas consideraciones. Las técnicas como Web Augmentation, para alterar los sitios Web existentes, con nuevas funcionalidades o contenidos, puede aportar una alternativa de solución que resulte beneficiosa en estos casos a un bajo costo. En esta tesina se definió un proceso de Portabilización de aplicaciones Web 1.0 en aplicaciones Web Mobile, con el fin de mejorar la usabilidad de las aplicaciones en dispositivos móviles. El proceso es asistido con una herramienta de Web Augmentation implementada en el contexto de esta investigación.

Palabras Clave

Web Augmentation, Diseño Web Adaptativo, Web Mobile, Patrones de Diseño Web Adaptativo, Portabilización, Encuestas, Goal Question Metric (GQM), Greasemonkey/Tampermonkey, Script de Portabilización.

Conclusiones

Con el fin de mejorar la experiencia del usuario al navegar sitios Web 1.0 con dispositivos móviles, se investigó la portabilización de aplicaciones Web mediante un proceso de portabilización asistido con una herramienta de Web Augmentation. La herramienta permite especificar las adaptaciones a realizar y generar en tiempo real la portabilización del sitio Web.

Se generaron casos de estudios y experimentos con el fin de evaluar la experiencia de navegación de los usuarios con las interfaces portabilizadas y sin portabilizar. Por medio de las encuestas realizadas y su análisis podemos concluir que el uso de la herramienta mejora la experiencia de navegación del usuario.

Trabajos Realizados

Investigación de las tecnologías de Diseño Web Adaptativo y Web Augmentation.

Desarrollo de una herramienta de Web Augmentation para la Portabilización de aplicaciones Web 1.0.

Planificación y ejecución de un experimento controlado utilizando la herramienta.

Realización de encuestas que permitieron evaluar la eficacia y/o eficiencia de la propuesta.

Trabajos Futuros

Complementar la herramienta con nuevos templates adaptativos y patrones de diseño.

Investigar métodos para la conservación de los estilos CSS, embebidos y externos, al portabilizar un sitio Web.

Analizar otras plataformas y medios para la distribución y ejecución de la herramienta.

Implementar estrategias de mejoras pasivas ante problemas de conectividad en entornos móviles, por medio de una caché client-side.

FACULTAD DE INFORMÁTICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Tesina de Licenciatura

Una herramienta de portabilización de aplicaciones Web 1.0 a Web Mobile

Alumnos:

Mauricio Javier Witkin,
Ramón Maximiliano Serrano

Director: Dr .José Matías Rivero

Co-director: Dr. Matias Urbieta

Índice

1. Introducción	4
1.1. Motivación	4
1.2. Objetivo y Alcance	8
1.3. Estructura de contenidos	9
2. Trabajos relacionados	10
2.1. A Platform for Web Augmentation Requirements Specification	10
2.2. A multi-screen refactoring system for video-centric Web applications	11
2.3. Sticklet: An End-User Client-Side Augmentation-Based Mashup Tool	12
3. Background	13
3.1. Tecnologías Web	13
3.2. Diseño Web Adaptativo de Interfaz de Usuario	16
3.2.1. Introducción	16
3.2.2. Píxel Lógico y Device Pixel Ratio	16
3.2.3. Bootstrap	19
3.2.4. Patrones de Diseño Web Adaptativo de Interfaz de Usuario	20
3.2.4.1. Layout	20
3.2.4.2. Navigation	21
3.2.4.3. Images	22
3.2.4.4. Data	23
3.2.4.5. Form	23
3.2.4.6. Text	24
3.2.4.7. Modules	24
4. Proceso de Portabilización	26
4.1. Identificar el sitio Web a portabilizar	28
4.2. Configuración del template adaptativo y de la URL	29
4.3. Selección de elementos y adaptaciones	30
4.4. Validación	32
4.5. Publicación	33
4.6. Instalación del script de portabilización	34
4.7. Navegación del sitio Web con UI adaptada	35
5. Otro caso de estudio: Pidgin	36
5.1. Portabilización de la página Home	36
5.2. Portabilización de la página Register	39
5.3. Portabilización de la página About	43
6. Arquitectura y detalles técnicos de la herramienta	46

6.1. Diseñador de Adaptaciones	46
6.1.1. Selección de template adaptativo	46
6.1.2. Configuración de URL	48
6.1.3. Selección de elementos, patrón adaptativo y ubicación	49
6.1.4. Previsualización	51
6.1.5. Exportación e importación de la configuración	52
6.1.6. Persistencia de datos	53
6.2. El Motor de Adaptaciones	53
6.2.1. Patrones de adaptación	54
6.2.1.1. Menú	54
6.2.1.2. Formulario	56
6.2.1.3. Títulos	57
6.2.1.4. Copiar	58
6.2.2. Importación de la portabilización	59
7. Encuestas y Evaluación	60
7.1. GQM	60
7.2. Planificación del Experimento	61
7.2.1. Metas	61
7.2.2. Participantes	61
7.2.3. Materiales del Experimento	63
7.2.4. Tareas	65
7.2.5. Hipótesis y Métricas	66
7.2.6. Diseño del Experimento	66
7.2.7. Procedimiento	69
7.2.8. Procedimiento de Análisis	69
7.3. Resultados	69
7.4. Análisis e Implicancias	76
8. Conclusiones y Trabajo Futuro	80
8.1. Conclusiones	80
8.2. Trabajo Futuro	81
Glosario	83
Referencias bibliográficas	85
Anexo A: Guía de instalación de la herramienta de portabilización	87
A.1. Instalación del Diseñador de Adaptaciones	87
A.2. Instalación del Motor de Adaptaciones	90
Anexo B: Resumen de la metodología GQM	98
Anexo C: Resultados de las encuestas	100

1. Introducción

1.1. Motivación

La masificación de los dispositivos móviles^{1,2} permite a los usuarios acceder a una gran variedad de aplicaciones desarrolladas sin haber tenido en cuenta que iban a ser accedidas desde dispositivos móviles tales como smartphones o tablets. Como consecuencia el usuario enfrenta problemas de usabilidad.

La decisión de mejorar la usabilidad de una aplicación Web convencional sobre dispositivos móviles, puede verse impedida en muchos casos debido al alto costo de implementación, que para la pequeña empresa podrían representar costos casi prohibitivos. Además de los costos monetarios, intervienen otros factores, tales como tiempo de desarrollo, contratación de recursos humanos calificados, capacitación del equipo de desarrollo. Una vez desarrollada la nueva aplicación se tienen los riesgos inherentes al lanzamiento de una nueva versión, como posibles bugs, problemas de compatibilidad, entre otros. Por estos motivos muchas veces se opta por no modificar la aplicación y como resultado el usuario mobile navega la página, enfrentando variados inconvenientes de usabilidad, tales como hacer zoom todo el tiempo y utilizar excesivamente el scroll.

Al navegar una página sin *diseño Web adaptativo* (Responsive Web Design, en adelante **RWD**) [1] con un dispositivo móvil nos encontraremos por lo general con una vista como muestra la figura 1.1.

¹ Ericsson Mobility Report, <https://www.ericsson.com/assets/local/mobility-report/documents/2015/emr-nov-2015-regional-report-latin-america-and-the-caribbean.pdf> - Visitado el 11/07/2016.

² Internet World Stats, <http://www.internetworldstats.com/stats2.htm> - Visitado el 11/07/2016.



Figura 1.1 - Página sin RWD en un dispositivo móvil.

Debido al reducido tamaño de las fuentes y las imágenes, el usuario está obligado a realizar zoom para poder leer o ver las imágenes. En la figura 1.2 podemos ver que se hizo zoom y gracias a eso se puede ver con claridad el contenido de la página.



Figura 1.2 - Zoom a una página sin RWD.

Podemos ver que recién en este momento es posible hacer click de manera segura sobre un link o imagen, dado que antes del zoom era muy complicado acertar con precisión al objeto deseado.

Ahora se puede ver claramente, pero el problema es que se debe hacer scroll para llegar a los distintos puntos de la página. Una vez realizados varios scrolls llegamos al menú de Registros, figura 1.3.



Figura 1.3 - Se necesitan realizar varios scrolls para llegar al menú de Registros.

Esto hace que el usuario se canse al usar la página y se sienta frustrado por no encontrar lo que está buscando. En el caso de esta página, el usuario no puede optar por no usarla ya que es una página oficial única, pero en el caso de poder optar por elegir otra alternativa que cuente con una versión adaptada a dispositivos móviles probablemente dejaría de utilizarla. Otro problema que ocasiona la poca usabilidad es que los usuarios no permanecerán en nuestro sitio más de lo justo y necesario, por lo que si nuestra sitio vende algún tipo de publicidad esto impactará de manera negativa en lo económico. Por último es probable que el usuario opte directamente por no usar la aplicación desde su móvil y la use siempre desde una PC, perdiendo la ventaja de la alta conectividad y masificación de móviles que tenemos hoy en día.

Por otro lado, el surgimiento de técnicas como Web Augmentation [8, 9], para alterar los sitios Web existentes, con nuevas funcionalidades o contenidos, puede aportar una alternativa de solución que resulte beneficiosa en estos casos.

A fin de evitar los problemas de usabilidad y costos, descritos anteriormente, se propone la implementación de una herramienta de *Web Augmentation*, que está basada en el trabajo de investigación “Técnicas avanzadas de aumentación Web: portabilización y Web semántica” [10]. Esta herramienta permite portabilizar una aplicación Web 1.0 tradicional a una aplicación Web para dispositivos móviles. El objetivo de la portabilización es presentar de manera adaptativa los contenidos de HTML generados por las aplicaciones Web sin necesidad de modificar el código fuente de la aplicación, sino con una estrategia de adaptación asistida desde el cliente, con un costo de implementación prácticamente nulo. La portabilización necesita una configuración básica por parte del diseñador, la cual es asistida a través de una herramienta que se desarrolló para este fin.

Para tener una métrica de la aceptación de los usuarios finales de dispositivos móviles se realizaron experimentos de uso para evaluar la utilidad y viabilidad del proyecto.

1.2. Objetivo y Alcance

El objetivo de esta tesina será investigar la portabilización de aplicaciones Web a dispositivos móviles mediante la utilización de una herramienta que facilite la introducción del RWD y un experimento de validación que permita evaluar la utilidad de la herramienta en aplicaciones del mundo real.

El alcance del presente trabajo incluye los siguientes puntos:

- Investigación, análisis y desarrollo de una solución que permita adaptar una página Web 1.0 tradicional a una que tenga RWD mediante el uso de un script.
- Mejora de la Herramienta de Portabilización que ya ha sido implementada como prueba de concepto [2]. Estas mejoras incluyen: 1) una interfaz gráfica que permitirá al diseñador realizar de manera intuitiva su trabajo, previsualizando en tiempo real los resultados de los cambios, 2) inclusión de Patrones de RWD, 3) opción de seleccionar el template adaptativo, 4) posibilidad de portabilizar un sitio Web completo y no sólo una página.
- Planificación y ejecución de un experimento controlado que permita evaluar la eficacia y/o eficiencia de la propuesta. Para la evaluación utilizamos encuestas a participantes, que realizaron unas tareas pautadas por el encuestador. Los datos obtenidos en estas encuestas fueron procesados utilizando métodos científicos de análisis de muestras.

1.3. Estructura de contenidos

El presente trabajo se divide en 8 capítulos y 3 anexos.

El primer capítulo contextualiza los temas a tratar, se establece la motivación, el objetivo y el alcance de la tesina, además nombra algunos conceptos importantes que en el resto de los capítulos se describirán con más detalle.

En el segundo capítulo se analizan algunos trabajos relacionados a los temas tratados en esta tesina que resultan de interés en este informe y cuyo aporte resulta relevante.

El tercer capítulo describe las tecnologías involucradas en las diferentes etapas del proceso de portabilización y profundiza en el concepto de diseño adaptativo.

El cuarto capítulo da inicio al desarrollo de la solución propuesta en este trabajo. En este sentido se describen las generalidades del proceso de portabilización de un sitio Web, a través de un ejemplo.

En el quinto capítulo se detalla el proceso de portabilización aplicado a un segundo caso de estudio, para abarcar nuevas situaciones con la herramienta.

El sexto capítulo describe la arquitectura de software y los detalles técnicos de la implementación de la herramienta. Además, se muestran las distintas funcionalidades y las opciones que nos provee.

En el séptimo capítulo se tratan las encuestas y el método *GQM*. Se detalla el procedimiento realizado para capturar los datos, la evaluación de resultados y las conclusiones obtenidas.

En el octavo capítulo se desarrollan las conclusiones finales sobre todo lo expuesto, como así también algunas ideas planteadas en propuestas de trabajos futuros.

Por último se brindan tres anexos, uno de ellos con las instrucciones de instalación de la herramienta, otro con los resultados de las encuestas y el último con un breve resumen de la metodología *GQM*.

2. Trabajos relacionados

Diversos autores ya han explorado temas tratados en este trabajo. El uso de Web Augmentation para mejorar la experiencia del usuario al navegar un sitio Web. Los procesos de transformaciones de aplicaciones, para hacer uso de nuevos recursos provistos por los cambios tecnológicos.

En este capítulo se describirán diferentes trabajos relacionados con nuestra propuesta. En particular, se detallarán las siguientes investigaciones: “A Platform for Web Augmentation Requirements Specification” [9] , “Sticklet: An End-User Client-Side Augmentation-Based Mashup Tool” [11] y “A multi-screen refactoring system for video-centric Web applications” [12].

También, como trabajos relacionados podemos mencionar que esta tesina se encuentra enmarcada en los proyectos de investigación de la Facultad de Informática de la UNLP [2, 10] donde se expusieron parte de las ideas presentadas en este trabajo.

2.1. A Platform for Web Augmentation Requirements Specification

Web Augmentation es una de las técnicas más populares para adaptar las aplicaciones Web existentes de terceros y cubrir nuevos requerimientos de usuarios. Esta técnica modifica la interfaz de usuario original, generalmente utilizando scripts que se ejecutan en el lado del cliente (client-side), por ejemplo el Navegador Web. En este contexto se distinguen dos tipos de usuarios: 1) scripters, capaces de crear un nuevo artefacto de augmentation, y 2) usuarios finales, que no poseen conocimientos de programación, pero consumen los artefactos para cubrir sus necesidades. El único lugar de contacto que tienen generalmente estos dos tipos de usuarios es a través de los repositorios de scripts. Los usuarios finales solicitan nuevos artefactos para cubrir sus necesidades a través de requerimientos de Web Augmentation utilizando descripciones textuales, que suelen ser difíciles de interpretar por los usuarios scripters. CrowdMock propone una metodología para definir y gestionar los requerimientos de Web Augmentation utilizando prototipos visuales enriquecidos y descripciones textuales que pueden asignarse automáticamente a artefactos de software ejecutables. Para respaldar este enfoque, los autores implementaron las herramientas UserRequirements and MockPlug. MockPlug es una herramienta client-side que funciona como una extensión del navegador Web que permite a los usuarios finales definir requerimientos de augmentation. UserRequirements es un repositorio, con funcionalidades del tipo red social, que administra los requerimientos CrowdMock.

El objetivo principal de CrowdMock es mejorar cómo funcionan las comunidades de Web Augmentation. Para ello, se trabajó principalmente en la

comunicación entre los solicitantes y creadores. Estas comunidades poseen una composición similar, existen creadores de artefactos, usuarios de los artefactos y solicitantes de artefactos. Esta metodología permite a los usuarios colaborar en el proceso de refinamiento y elegir la mejor versión para implementar, administrando de esta manera el esfuerzo realizado por los creadores. Además, proporciona una forma de generar automáticamente el código fuente inicial para implementar la augmentation que satisfaga sus requisitos.

Al observar CrowdMock y nuestra propuesta se observan similitudes, ambas proveen herramientas implementadas como una extensión del navegador Web, y permiten acercar a los usuarios finales que no poseen conocimiento de programación a la técnica de Web Augmentation. No obstante CrowdMock, está orientado a la definición y gestión de requerimientos de Web Augmentation en contextos generales, no es de gestión específica de dispositivos móviles y los problemas de visualización o navegación que estos presentan.

2.2. A multi-screen refactoring system for video-centric Web applications

Las aplicaciones Web permiten a los usuarios acceder al mismo contenido desde varios dispositivos en diferentes lugares. Como consecuencia, el consumo del contenido ha cambiado debido a esta ubicuidad y al creciente número de dispositivos conectados disponibles para un mismo usuario, provocando un incremento de la demanda de aplicaciones multi-pantalla (Multi-screen applications, en adelante MSAs). Este tipo de aplicaciones consta de múltiples componentes. Cada componente se ejecuta en un dispositivo distinto y posee un contenido particular que cumple una función específica. Estos componentes están siempre en comunicación entre sí para proporcionar un uso complementario. Las aplicaciones que pueden funcionar en el sistema propuesto deben cumplir dos requisitos principales: 1) deben ser de una sola pantalla y dedicados para un dispositivo de escritorio específico, y 2) deben ser aplicaciones que contengan al menos un elemento de contenido multimedia, por ejemplo, audio o video.

Entre los diversos desafíos relacionados con la creación de MSAs, el sistema se centralizó en la caracterización del entorno, en la división y distribución gráfica de la interfaz de usuario, en la adaptación del diseño a los dispositivos y el contenido, y finalmente en la funcionalidad de la aplicación en todos los dispositivos. Este enfoque analiza la estructura de la aplicación y sus aspectos visuales para segmentar la interfaz de usuario en bloques. Los bloques se caracterizan por las funciones que se pueden asignar a los dispositivos. Luego, se separan los bloques con diferentes funciones para crear dos aplicaciones, una maestra y una esclava. Ambas aplicaciones se amplían con una lógica que les permiten las acciones de descubrir y comunicarse entre sí en la red, para sincronizar su vista y sus datos continuamente durante el tiempo de ejecución. La disposición del componente maestro también se refactoriza dinámicamente para

cubrir toda la ventana del dispositivo y para eliminar el desplazamiento horizontal del componente esclavo.

El enfoque multi-screen comparte elementos en común con nuestra propuesta: 1) implementa una solución client-side, 2) utiliza una extensión del navegador web, 3) hace uso de tecnologías de diseño adaptativo (RWD) para presentar el contenido. Sin embargo, las características de las aplicaciones a transformar y los temas centrales de análisis son diferentes. La aplicación Web es de contenido multimedia en forma de video o audio, y la investigación se basa en cómo distribuir y sincronizar el contenido en las múltiples pantallas disponibles.

2.3. Sticklet: An End-User Client-Side Augmentation-Based Mashup Tool

Sticklet es una herramienta de Web Augmentation que concibe los sitios Web como muros donde puede agregarse sticky notes que contienen fragmentos de HTML de otras páginas o servicios Web. Un editor dedicado facilita a los usuarios finales realizar las definiciones a través de expresiones en un Domain Specific Language (DSL) propio. Se distribuye como un complemento que requiere la instalación previa de Greasemonkey, como extensión para el navegador Web Firefox.

Al igual que en nuestra propuesta, la ejecución de las augmentations se realizan en el navegador Web del usuario final a través de un administrador de scripts y se proveen características para facilitar el acceso a usuarios sin conocimiento de programación en Javascript, pero la funcionalidad principal que provee tiene un objetivo diferente, dar soporte para enriquecer un sitio Web a través de información de otros sitios Web, por medio de un DSL, con ayuda de un editor.

3. Background

En este capítulo, se da un panorama del contexto técnico utilizado para el diseño e implementación de las mejoras en la Herramienta de Portabilización mencionada en la subsección 1.2. Esta herramienta se compone de dos módulos principales, el Diseñador y el Motor de Adaptaciones. Además, se describen las tecnologías usadas para el desarrollo de ambos módulos. Luego se detalla el RWD, poniendo especial atención en los patrones de diseño adaptativo y el framework Bootstrap³.

3.1. Tecnologías Web

En el Diseñador de Adaptaciones las tecnologías utilizadas fueron CSS, JavaScript, JQuery⁴, Greasemonkey⁵ y HTML 5 Storage⁶. Para la utilización del script de portabilización en la etapa de diseño se utiliza el Motor de Adaptaciones, sumado a las herramientas Dolphin⁷ Browser y Tampermonkey⁸, a fin de ejecutar el script generado en el dispositivo móvil del usuario final. Además de estas tecnologías se utilizó Bootstrap, el cual merece una explicación más detallada que encontraremos en la subsección 3.2.4.

JavaScript

Es un lenguaje interpretado, orientado a objetos con funciones de primera clase, utilizado principalmente en páginas Web. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientado a objetos e imperativo.

CSS (Cascading Style Sheets)

Es el lenguaje utilizado para describir la presentación de documentos HTML o XML, posee una especificación estandarizada por parte del W3C⁹ (World Wide Web Consortium). CSS describe como debe ser renderizado el elemento estructurado en pantalla o en otros medios.

jQuery

El DOM (Document Object Model) es el Modelo de Objetos para representar Documentos, es una interfaz de programación de aplicaciones que permite acceder, agregar, quitar y modificar de manera dinámica el contenido en documentos HTML y XML.

³ <https://getbootstrap.com/docs/3.3/> - Visitado 09/11/2017

⁴ <https://jquery.com/> - Visitado el 28/11/2017

⁵ <https://www.greasepot.net> - Visitado el 15/01/2018

⁶ <http://www.w3.org/TR/webstorage/> - Visitado el 11/07/2016

⁷ <https://dolphin.com/> - Visitado el 09/11/2017

⁸ <https://tampermonkey.net> - Visitado el 09/11/2017

⁹ <https://www.w3.org/> - Visitado el 28/11/2017

jQuery [13] es una biblioteca liviana de JavaScript, pensada para interactuar con los elementos de una página Web por medio del DOM. Permite la selección de elementos DOM, la interactividad y modificaciones del árbol DOM, incluyendo soporte y manipulación de CSS. Manejo de eventos, generación de efectos y animaciones.

Lo que la hace tan especial es su sencillez y su reducido tamaño. Esta biblioteca permite enriquecer estéticamente una página Web, a la vez que simplifica la manera de manipular los documentos.

jQuery es un software libre, de código abierto y con una gran cantidad de funciones, que se ejecuta del lado del cliente. Ofrece una serie de funcionalidades basadas en JavaScript, que de otra manera requerirían de mucho más código, reduciendo los tiempos de desarrollo y los tamaños de los programas. Es compatible con la mayoría de los navegadores (Chrome, Edge, Firefox, Dolphin, IE, Safari entre otros) y posee soporte de operaciones Ajax a través de una API cross browser.

Greasemonkey y Tampermonkey

Greasemonkey es una extensión (add-on) para el navegador Web Firefox (ver Figura 3.1) que permite personalizar la apariencia y el funcionamiento de las páginas web. Funciona como un administrador de *user scripts* (pequeñas porciones de código que añaden funcionalidad, eliminan molestias y en general mejoran las páginas Web que el usuario navega). Los scripts están escritos en lenguaje Javascript y poseen acceso a funcionalidades privilegiadas. La ejecución se realiza en el lado del cliente (client-side) a través del navegador Web, por lo que se pueden inyectar en páginas Web arbitrarias.

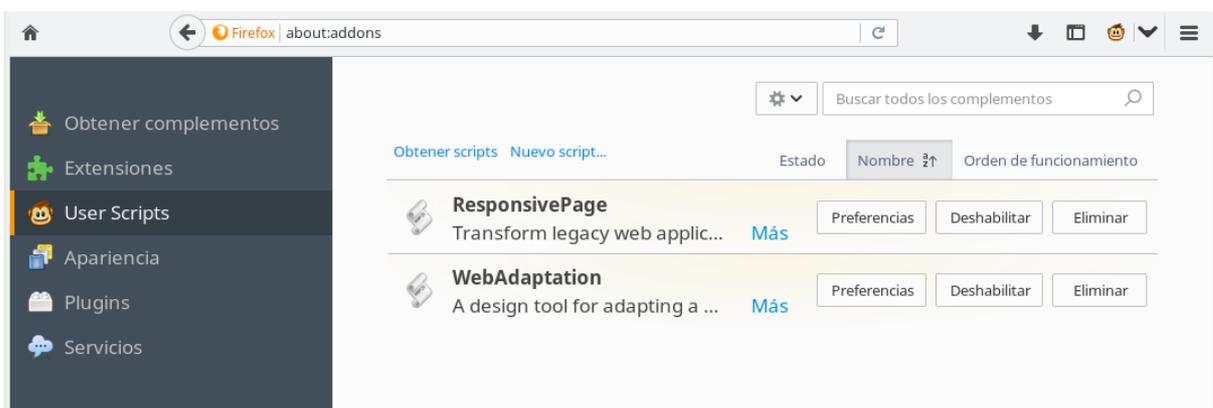


Figura 3.1 - Extensión Greasemonkey en Mozilla Firefox.

Los *user scripts* permiten realizar distintas acciones [14]:

- Mejorar la legibilidad y usabilidad de un sitio Web.
- Corregir problemas en sitios Web donde los propietarios no dan soporte.

- Recuperar datos automáticamente de otros sitios Web para obtener sitios más interconectados.
- Modificar las páginas Web para que funcionen mejor con las tecnologías de asistencia, como la conversión de texto a voz.
- Agregar, modificar o eliminar contenido de la página Web modificando el HTML DOM (Document Object Model).

Luego de Greasemonkey, surgieron en otros navegadores Web diferentes extensiones con funciones similares que permiten la ejecución de *user scripts*, en la siguiente figura se detallan los principales navegadores Web y sus extensiones.

Navegador Web	Extensiones
Google Chrome	Tampermonkey, Violentmonkey
Mozilla Firefox	Greasemonkey, Tampermonkey, Violentmonkey, Scriptish
Apple Safari	Tampermonkey
Microsoft Edge	Tampermonkey
Opera	Tampermonkey, Violentmonkey
Dolphin	Tampermonkey

Figura 3.2 - Extensiones de navegadores Web para administrar los users scripts.

Los scripts son compartidos mediante repositorios públicos, donde están disponibles para que otros usuarios los descarguen. Algunos ejemplos de repositorios son Greasy Fork¹⁰ y OpenUserJS¹¹.

La extensión Tampermonkey¹² es el administrador de *user scripts* más popular. Se encuentra disponible en Google Chrome (ver Figura 3.3) y otros navegadores Web (ver Figura 3.2), además posee compatibilidad con Greasemonkey hasta la versión 3.

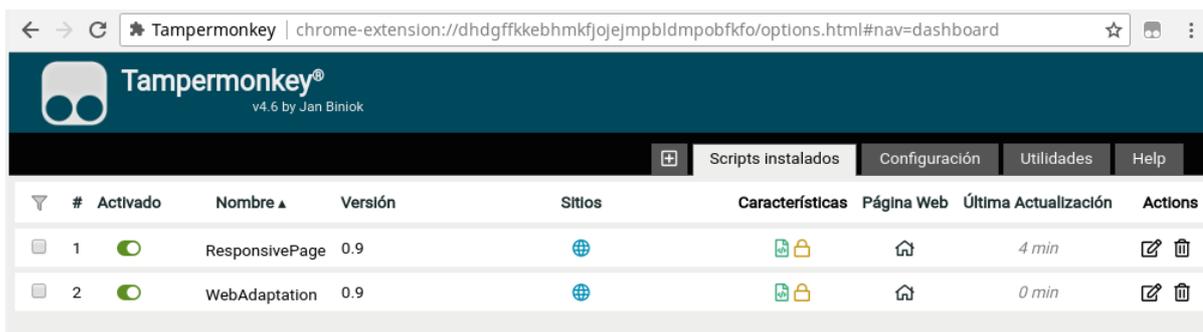


Figura 3.3 - Extensión Tampermonkey en Google Chrome.

¹⁰ <https://greasyfork.org> - Visitado el 15/01/2018

¹¹ <https://openuserjs.org> - Visitado el 15/01/2018

¹² <https://addons.mozilla.org/es/firefox/addon/tampermonkey> - Visitado el 15/01/2018

En resumen, las características principales que provee Tampermonkey son:

- Funciones para agregar, modificar, eliminar, habilitar o deshabilitar fácilmente los user scripts.
- Editor integrado de scripts, con validación de sintaxis.
- Importación y exportación basada en ZIP o almacenamiento en la nube (Google Drive, Dropbox, OneDrive).
- Actualizaciones automáticas de los scripts.
- Permite definir una blacklist con los sitios web a los que un user script nunca debería acceder.

3.2. Diseño Web Adaptativo de Interfaz de Usuario

3.2.1. Introducción

En la actualidad una gran variedad de dispositivos (computadoras de escritorio, netbooks, celulares, tablets, smart TV, etc.) tienen acceso a Internet y esto puede convertirse en un problema cuando diseñamos una página Web, ya que debemos considerar el acceso al sitio Web desde múltiples plataformas, con diferentes tamaños y resoluciones de pantalla.

El *Diseño Web Adaptativo* (RWD) de *Interfaz de Usuario* (User Interface, en adelante UI) presenta una solución a este problema, permitiéndonos utilizar el mismo HTML en todos los dispositivos, mientras que la hoja de estilos CSS cumple la tarea de cambiar el aspecto de la página Web de acuerdo con el tamaño de la pantalla, reorganizando los elementos según la pantalla se amplía o reduce.

En este capítulo se presentarán diferentes conceptos relacionados al RWD de UI que permiten la implementación de un sitio Web adaptable a diferentes dispositivos.

3.2.2. Píxel Lógico y Device Pixel Ratio

El píxel lógico está determinado por el Device Pixel Ratio^{13,14} (DPR) y los píxels físicos del dispositivo. El píxel físico es cada uno de los puntos que forman la pantalla de un dispositivo, es el punto del hardware. Por ejemplo, si el DPR es 1 entonces un píxel lógico tendrá un píxel físico. Si tiene un DPR de 2 entonces un píxel lógico ocupará un área de la pantalla de 2×2 píxels físicos.

En diseño Web el DPR se representa a través del CSS Pixel Ratio (DPR = CSS Pixel Ratio), que nos dará la resolución de la pantalla de un dispositivo interpretada por el CSS. O sea, que todas las propiedades CSS relacionadas con las dimensiones de los elementos responderá a la resolución lógica y no a la resolución física.

¹³ <https://developer.mozilla.org/en-US/docs/Web/API/Window/devicePixelRatio> - Visitado el 28/11/2017

¹⁴ <https://docs.imgix.com/tutorials/responsive-images-srcset-imgix> - Visitado el 28/11/2017

La razón por la que se creó el DPR o CSS Pixel Ratio fue el aumento de la resolución en los dispositivos móviles. Con el aumento de la densidad de píxeles físicos, si todos los dispositivos tuvieran un DPR de 1 las páginas se renderizarían demasiado pequeñas y por consiguiente el usuario tendría dificultades para apreciar los detalles de las imágenes o leer las letras. Existen diferentes situaciones que pueden presentarse de acuerdo a las características del dispositivo, veamos algunos ejemplos:

Ejemplo 1, cuando visualizamos una Web con un teléfono celular Samsung Galaxy S4 con las siguientes características:

Resolución física: 1080 x 1920 pixels

DPR: 3

Resolución lógica: $(1080/3) \times (1920/3) = 360 \times 640$

el CSS interpretará que el dispositivo tiene una resolución de 360×640 y las @media queries responderán a dicha resolución. Pero los elementos renderizados en la pantalla tendrán el triple de definición que una pantalla con una resolución física de 360×640 .

Ejemplo 2, en la *Figura 3.4* podemos ver como se visualiza una letra 'a' en un iPhone 3G y en un iPhone 6. En ambos casos el tamaño real es el mismo, a fin de que la letra sea legible. La mayor resolución en el caso del iPhone 6 es utilizada para darle mayor nivel de detalle a la letra.



Figura 3.4 - Pixel físico, pixel lógico y DPR

Ejemplo 3, si tenemos un monitor de escritorio de 24" con una resolución de 1920×1080, y encogemos ese monitor a un tamaño inferior a las 5" manteniendo la misma resolución, los elementos mostrados se verían demasiado pequeños. Mediante la utilización del pixel lógico o del DPR un dispositivo con una densidad

de píxeles muy alta no necesariamente significará ver el contenido más pequeño, porque el renderizado de la página estará siempre sujeto al DPR del dispositivo.

Ejemplo 4, el iPhone 4 y el iPhone 4S reportan un DPR de 2, debido a que la resolución física es el doble que la resolución lógica.

iPhone 4 y iPhone 4S:

Resolución física: 960 x 640

Resolución lógica: 480 x 320

Otros dispositivos informan de diferentes DPR, incluyendo valores no enteros. Todo esto nos da la idea que nunca se debe diseñar para un dispositivo específico. Esto tiene muchas implicaciones cuando se trata de diseño de páginas Web, tales como la preparación de los recursos Web, como las imágenes de alta definición y considerar el acceso a la aplicación desde dispositivos con diferentes Device Pixel Ratios. No se quiere forzar a un dispositivo de gama baja para descargar una imagen de muy alta resolución, sólo para reducir la escala a nivel local. También es posible que no desee que los dispositivos de gama alta usen imágenes de baja resolución para una experiencia de usuario borrosa.

Para evitar estos problemas con imágenes de mapa de bits y dar cabida a muchos Device Pixel Ratios se puede utilizar CSS Media Queries¹⁵ para proporcionar diferentes conjuntos de recursos para los diferentes grupos de dispositivos.

Ejemplo, configuración CSS para una imagen de fondo:

```
#element { background-image: url('lores.png'); }

@media only screen and (min-device-pixel-ratio: 2) {
  #element { background-image: url('hires.png'); }
}

@media only screen and (min-device-pixel-ratio: 3) {
  #element { background-image: url('superhires.png'); }
}
```

De esta manera, cada tipo de dispositivo sólo cargará la imagen debida. También hay que tener en cuenta que la unidad de píxeles en CSS opera siempre en píxeles lógicos.

A medida que aparecen más y más tipos de dispositivos, se vuelve más difícil proporcionarle a todos ellos los recursos adecuados de mapas de bits. Las *media queries* en CSS, es actualmente la única forma. Además en HTML5, el elemento `picture` permite utilizar diferentes fuentes para diferentes media queries.

¹⁵ <https://www.w3.org/TR/css3-mediaqueries/> - Visitado el 09/11/2017

3.2.3. Bootstrap

Bootstrap fue desarrollado en sus inicios por dos empleados de Twitter, Mark Otto y Jacob Thornton, con el propósito de estandarizar el conjunto de herramientas frontend de los ingenieros de toda la empresa. Desde que Bootstrap [3, 4] se lanzó en agosto de 2011, ha ganado popularidad, siendo ampliamente adoptado por los desarrolladores Web.

Bootstrap es un framework HTML, CSS, JavaScript para el desarrollo adaptativo de proyectos mobile first en la Web. Se distribuye con CSS standard y en su código fuente utiliza los preprocesadores CSS más populares, Less y Sass. Permite escalar de forma fácil y eficiente sitios Web o aplicaciones con un único código fuente de base, desde teléfonos, tablets y computadoras de escritorio a través de CSS media queries. Además, posee una documentación extensa para elementos HTML comunes, componentes personalizados de HTML y CSS, y complementos jQuery. Bootstrap es de código abierto y se encuentra alojado, desarrollado y mantenido en GitHub.

Bootstrap es un marco prometedor para el diseño Web que posee las siguientes características [5]:

- **Reusabilidad**

Bootstrap tiene componentes listos, estilos CSS y complementos que se pueden incluir directamente en el código. Este aspecto ahorra una considerable cantidad de tiempo y esfuerzo, lo que resulta en un rápido desarrollo. Además, esto permite un fácil mantenimiento del código y ayuda a organizar el código de manera eficiente.

- **Consistencia**

Como Bootstrap usa fragmentos de código listos para usar y es compatible con diferentes navegadores, existe un alto grado de uniformidad en el proceso de diseño. Esto también reduce la curva de aprendizaje para los nuevos diseñadores que desean construir sobre el mismo proyecto o implementar una funcionalidad similar en diferentes proyectos.

- **Flexible Grid Layout**

Bootstrap tiene un sistema de grilla predeterminada que puede escalar hasta 12 columnas con el aumento relativo en el tamaño de la pantalla y con la flexibilidad de optar por una grilla de respuesta fija o fluida. Además de esto, Bootstrap permite agregar cualquier número de columnas personalizadas que pueda necesitar fila por fila. Al utilizar las medias queries CSS, también puede manipular ciertos bloques de contenido haciéndolos aparecer u ocultar en función del tamaño de la pantalla.

- **Personalización**

Bootstrap es personalizable, permite elegir las funciones que se quieren usar y cuales no. Se puede usar una hoja de CSS personalizada para anular los estilos predeterminados de Bootstrap y usar los archivos LESS

para el preprocesamiento de CSS. Usando LESS se puede alterar casi todos los atributos predeterminados definidos. Además, puede personalizar la manera en que los complementos como Modals y Alerts funcionan usando JavaScript avanzado.

- **Comunidad con múltiples iniciativas de terceros**

Bootstrap tiene una comunidad activa de desarrolladores, así como un importante soporte de terceros que alojan una amplia gama de plantillas, editores y constructores, y fragmentos a medida, que ayudan a optimizar el diseño Web con Bootstrap.

- **Perspectiva a futuro y desarrollo abierto**

El desarrollo de Bootstrap se lleva a cabo en GitHub. Se pueden seguir todos los cambios implementados y ver los registros de problemas pendientes con la facilidad de informar errores relacionados con Bootstrap, además de contribuir al desarrollo futuro de Bootstrap. La hoja de ruta del proyecto también se puede encontrar en el sitio Web oficial junto con aspectos tales como la compatibilidad con versiones anteriores. Bootstrap es un framework que incorpora HTML5 y CSS3, además de una gran cantidad de conjuntos de herramientas y utilidades, por lo que se está convirtiendo en un punto de referencia para el diseño y desarrollo de las nuevas aplicaciones.

3.2.4. Patrones de Diseño Web Adaptativo de Interfaz de Usuario

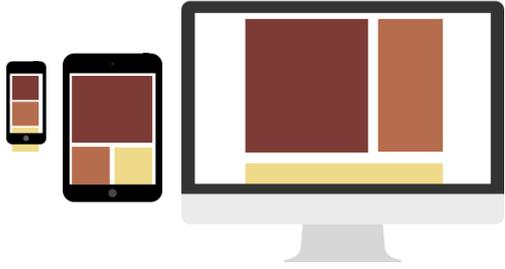
Los patrones^{16,17} de RWD de UI proveen soluciones a problemas comunes con la UI cuando son accedidas desde dispositivos con diferentes tamaños de pantalla, para las cuales no fueron desarrolladas. De acuerdo al problema que tratan, podemos agruparlos en diferentes tipos de patrones: layout, navigation, images, data, form, text, modules. A continuación se mencionan algunos ejemplos de patrones de cada tipo.

3.2.4.1. Layout

El esquema de distribución de los elementos de la página no se adapta correctamente al *viewport*, provocando que ciertos elementos sean ilegibles y de difícil acceso, como consecuencia el usuario debe realizar zooming o scrolling para poder verlos o interactuar con ellos.

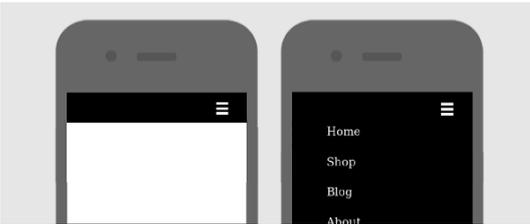
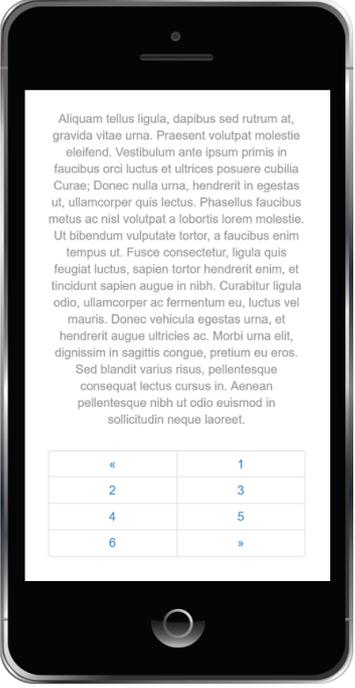
¹⁶ <https://bradfrost.github.io/this-is-responsive/patterns.html> - Visitado 10/11/2017

¹⁷ <https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns> - Visitado 10/11/2017

<p>Mostly Fluid</p>		<p>Adapta contenido de la página alterando la distribución de los elementos, el orden y la cantidad de columnas para obtener una visualización acorde al tamaño de pantalla del dispositivo. Los elementos que no entran en una fila se desplazan hacia abajo.</p>
<p>Off Canvas</p>		<p>En pantallas pequeñas algunos elementos como por ej. los menús se contraen automáticamente para liberar espacio. El usuario puede expandir el contenido mediante la interacción de un widget. Al expandirse el contenido se realiza un desplazamiento de los otros elementos en la pantalla.</p>
<p>List with Thumbnails</p>		<p>Modifica la visualización de una lista de elementos para ajustarla al tamaño de pantalla, donde cada elemento se compone de una imagen y una descripción. En pantallas pequeñas visualiza menos elementos por fila para mantener a los elementos legibles y facilitar la interacción. Los elementos que no caben en una la fila se desplazan a la siguiente.</p>

3.2.4.2. Navigation

La navegación presenta dificultades cuando se visita un sitio Web no diseñado para dispositivos móviles, el reducido tamaño de los widgets de navegación y las fuentes provocan dificultades en la navegabilidad del sitio Web. Esto conduce al usuario a la utilización del zoom para mejorar la visualización y precisión al seleccionar un widget de navegación en el sitio Web.

<p>Toggle Menu</p>		<p>En pantallas pequeñas, aparece un botón de menú que, al hacer click, expande un navegador de estilo acordeón, aprovechando el espacio disponible para mostrar las opciones con mayor claridad.</p>
<p>Paginación</p>		<p>La paginación de contenidos en pantallas pequeñas se ajusta para mostrar menos elementos y utilizar widgets de navegación de mayor tamaño que ayudan a realizar la selección de la página deseada con mayor precisión.</p>

3.2.4.3. Images

Ver una imagen a tamaño real requiere de un dispositivo con un alto y ancho en píxeles capaz de contener los píxeles de la misma, una resolución inferior requeriría de scroll vertical u horizontal.

<p>Basic Fluid Image</p>		<p>Redimensiona la imagen con respecto al contenedor, para así hacerla adaptable a las distintas resoluciones posibles. Se reemplaza las unidades en píxeles por unidades relativas en la definición de estilos CSS del sitio Web.</p>
--------------------------	---	--

3.2.4.4. Data

Las tablas con datos que fueron diseñadas para ser visualizadas en pantallas de computadoras de escritorio, presentan un tamaño que excede el máximo disponible en pantallas pequeñas, haciendo necesario el uso de scroll para acceder a las columnas y filas más alejadas, además si las fuentes son ilegibles el usuario va a hacer uso del zoom.

<p>Respon ve Table</p>	 <p>A photograph of an iPhone displaying a table with two columns and 15 rows of data. The data is as follows:</p> <table border="1"><thead><tr><th>Field</th><th>Value</th></tr></thead><tbody><tr><td>Date of Birth</td><td>January 13, 1979</td></tr><tr><td>Dream Vacation City</td><td>Gotham City</td></tr><tr><td>GPA</td><td>3.1</td></tr><tr><td>Arbitrary Data</td><td>RBX-12</td></tr><tr><td>First Name</td><td>The</td></tr><tr><td>Last Name</td><td>Tick</td></tr><tr><td>Job Title</td><td>Crimefighter Sorta</td></tr><tr><td>Favorite Color</td><td>Blue</td></tr><tr><td>Wars of Trek?</td><td>Wars</td></tr><tr><td>Porn Name</td><td>John Smith</td></tr><tr><td>Date of Birth</td><td>July 19, 1968</td></tr><tr><td>Dream Vacation City</td><td>Athens</td></tr><tr><td>GPA</td><td>3.1</td></tr><tr><td>Arbitrary Data</td><td>RBX-12</td></tr></tbody></table>	Field	Value	Date of Birth	January 13, 1979	Dream Vacation City	Gotham City	GPA	3.1	Arbitrary Data	RBX-12	First Name	The	Last Name	Tick	Job Title	Crimefighter Sorta	Favorite Color	Blue	Wars of Trek?	Wars	Porn Name	John Smith	Date of Birth	July 19, 1968	Dream Vacation City	Athens	GPA	3.1	Arbitrary Data	RBX-12	<p>Transforma una tabla de varias columnas en una nueva tabla de sólo 2 columnas donde la primer columna repite la cabecera de la tabla para cada fila y la segunda columna contiene los valores correspondientes a una fila. La exploración de los datos se realiza hacia abajo mediante scroll vertical.</p>
Field	Value																															
Date of Birth	January 13, 1979																															
Dream Vacation City	Gotham City																															
GPA	3.1																															
Arbitrary Data	RBX-12																															
First Name	The																															
Last Name	Tick																															
Job Title	Crimefighter Sorta																															
Favorite Color	Blue																															
Wars of Trek?	Wars																															
Porn Name	John Smith																															
Date of Birth	July 19, 1968																															
Dream Vacation City	Athens																															
GPA	3.1																															
Arbitrary Data	RBX-12																															

3.2.4.5. Form

Ingresar datos en un formulario utilizando un dispositivo móvil resulta una tarea no tan sencilla debido al tamaño pequeño de las fuentes y de los campos de ingreso de datos, que hacen necesario el uso del zoom y el desplazamiento de la pantalla.

<p>Top to left labels</p>		<p>En pantallas pequeñas, los campos de ingreso de datos del formularios se expanden para ocupar el ancho de la pantalla disponible y desplaza las etiquetas de cada campo a una fila arriba.</p>
---------------------------	---	---

3.2.4.6. Text

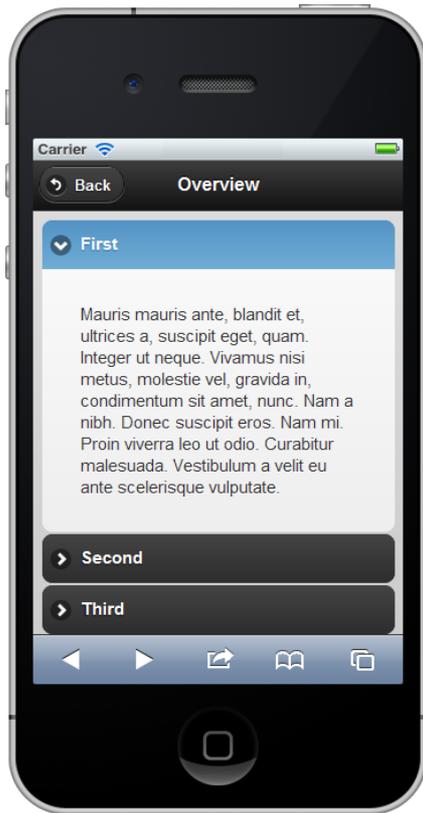
La legibilidad de los textos en un sitio Web se ve reducida en dispositivos con pantalla pequeña, por lo que el usuario debe realizar zoom para mejorar la visibilidad.

<p>Fit text</p>		<p>Escala el tamaño del texto para ajustarse al área del tamaño de la pantalla. Dependiendo del viewport disponible se aplican diferentes tamaños a las fuentes a través de definiciones en los estilos CSS.</p>
-----------------	---	--

3.2.4.7. Modules

La visualización y navegación de páginas compuestas por diferentes módulos o secciones que aprovechan el ancho de las pantallas de las computadoras de escritorio, no resulta cómoda en pantallas pequeñas. El usuario debe aplicar zoom y desplazar la pantalla hacia un lado para acceder a la sección que necesita.

Tabs to accordion



Transforma los tabs en un nuevo widget acordeón cuando el tamaño de pantalla es reducido. Este nuevo widget se visualiza aprovechando el tamaño de pantalla disponible y mejora la visualización y la navegación.

Slide-Down Notification Bar



Los mensajes se muestran en una barra de notificación, que ante un nuevo mensaje se desliza hacia abajo y se adapta al tamaño de pantalla. Una vez que el usuario confirma la lectura del mensaje o después de un tiempo preestablecido se oculta la barra de notificación.

4. Proceso de Portabilización

El proceso de portabilización de aplicaciones Web a dispositivos móviles (ver Figura 4.1) permite mejorar la experiencia de usabilidad de un sitio Web, que no fue desarrollado para ser navegado con dispositivos móviles. El proceso comienza seleccionando un sitio Web que presenta problemas de usabilidad en un dispositivo móvil. Luego el diseñador trabaja con la herramienta sobre el conjunto de páginas seleccionadas realizando adaptaciones y validando en el previsualizador. Finalmente exporta un script con las adaptaciones de la UI del sitio Web. El usuario de un dispositivo móvil importa el script utilizando un administrador de scripts para navegadores Web, como por ejemplo Tampermonkey y luego mientras navega en el sitio Web portabilizado, el script se encarga de ejecutar las adaptaciones en tiempo real.

En el siguiente diagrama se pueden apreciar los puntos principales de este proceso, las actividades numeradas desde el 2) hasta 4) del proceso, se realizarán en cada una de las páginas del sitio que el diseñador decida portabilizar. Los pasos 1 al 7 serán explicados en detalle en las subsecciones 4.1 hasta la 4.7 respectivamente.

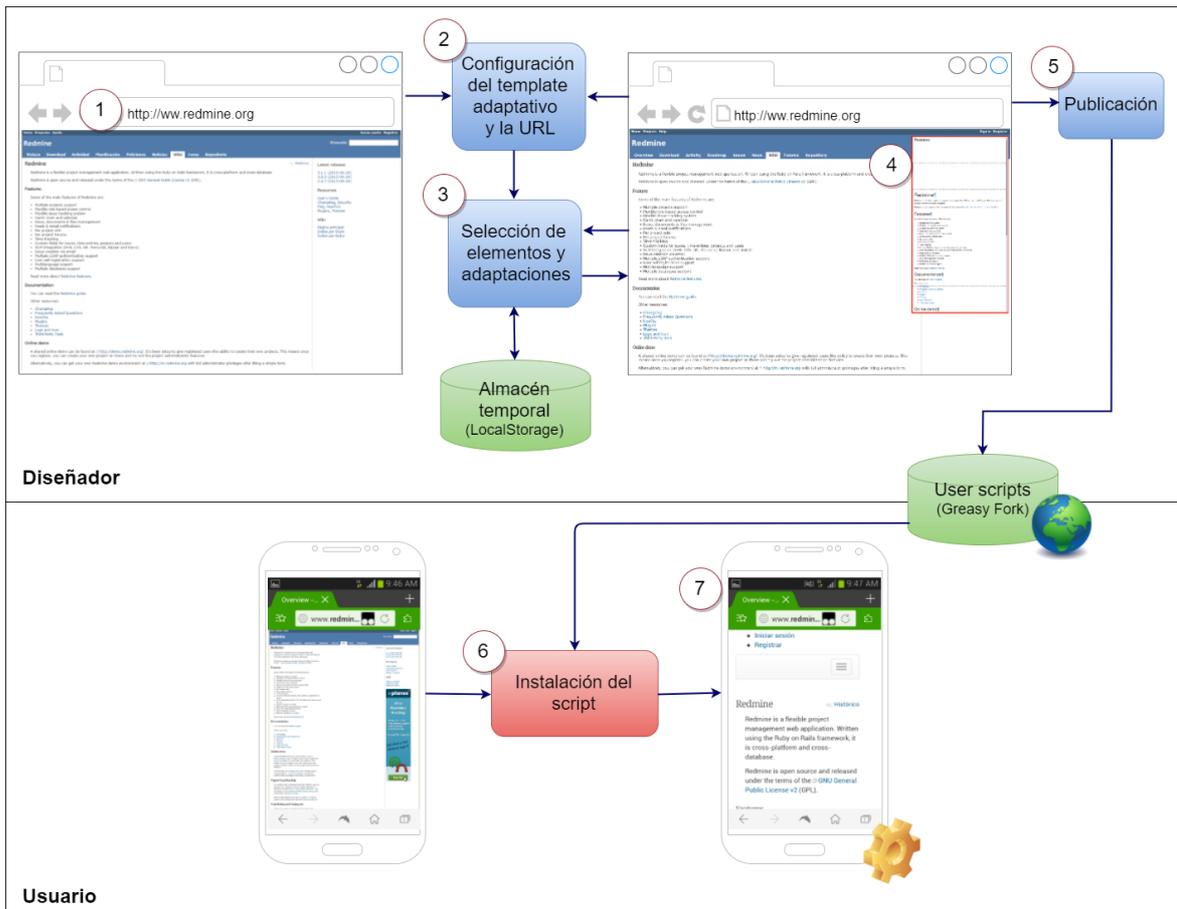


Figura 4.1 - Esquema general del proceso de portabilización.

Como caso de estudio para describir el proceso de portabilización se utilizó el sitio Redmine¹⁸ por sus características de sitio Web 1.0 y porque no cuenta con versión para dispositivos móviles. Se utilizó como dispositivo móvil de referencia un celular Moto G (3ª generación) con una resolución de 1280x720 píxeles y CSS pixel ratio 2.

¹⁸ <http://www.redmine.org/> - Visitado el 31/10/2017

4.1. Identificar el sitio Web a portabilizar

En general los sitios a portabilizar serán Web 1.0 (ver Figura 4.2), que al ser navegados en un dispositivo móvil, con un viewport de tamaño reducido, muestran un exceso de información para el tamaño de pantalla, provocando que el usuario realice continuamente eventos de zoom para ver correctamente el contenido, y eventos de scroll para desplazarse por la página Web.

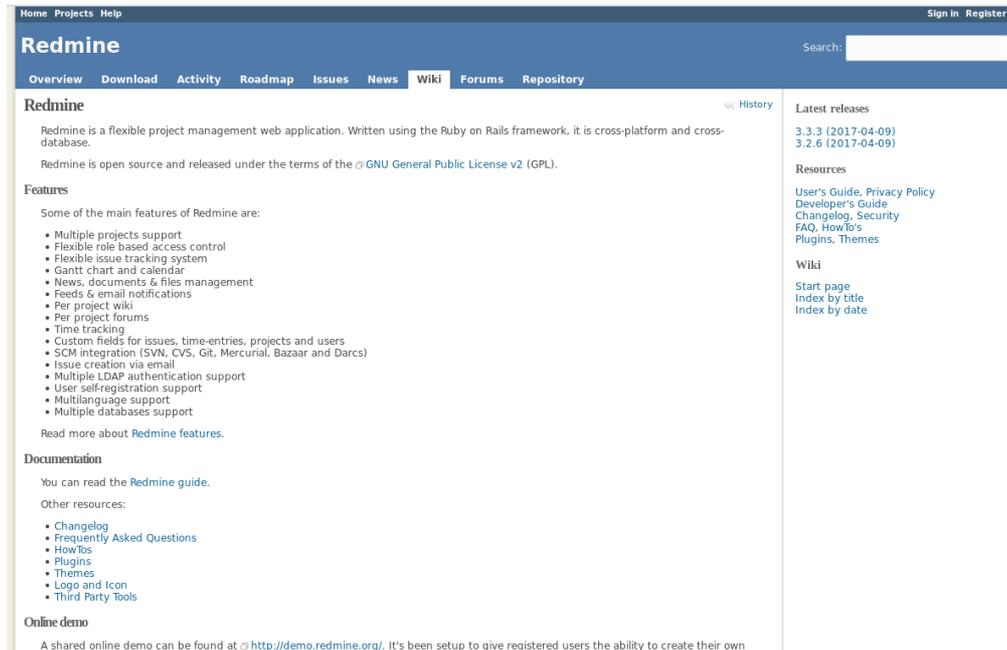


Figura 4.2 - Sitio visto desde una computadora de escritorio.

Analizando el sitio Web Redmine (ver Figura 4.3) podemos identificar varios problemas que afectan la usabilidad del sitio, como el tamaño de las fuentes y el menú:

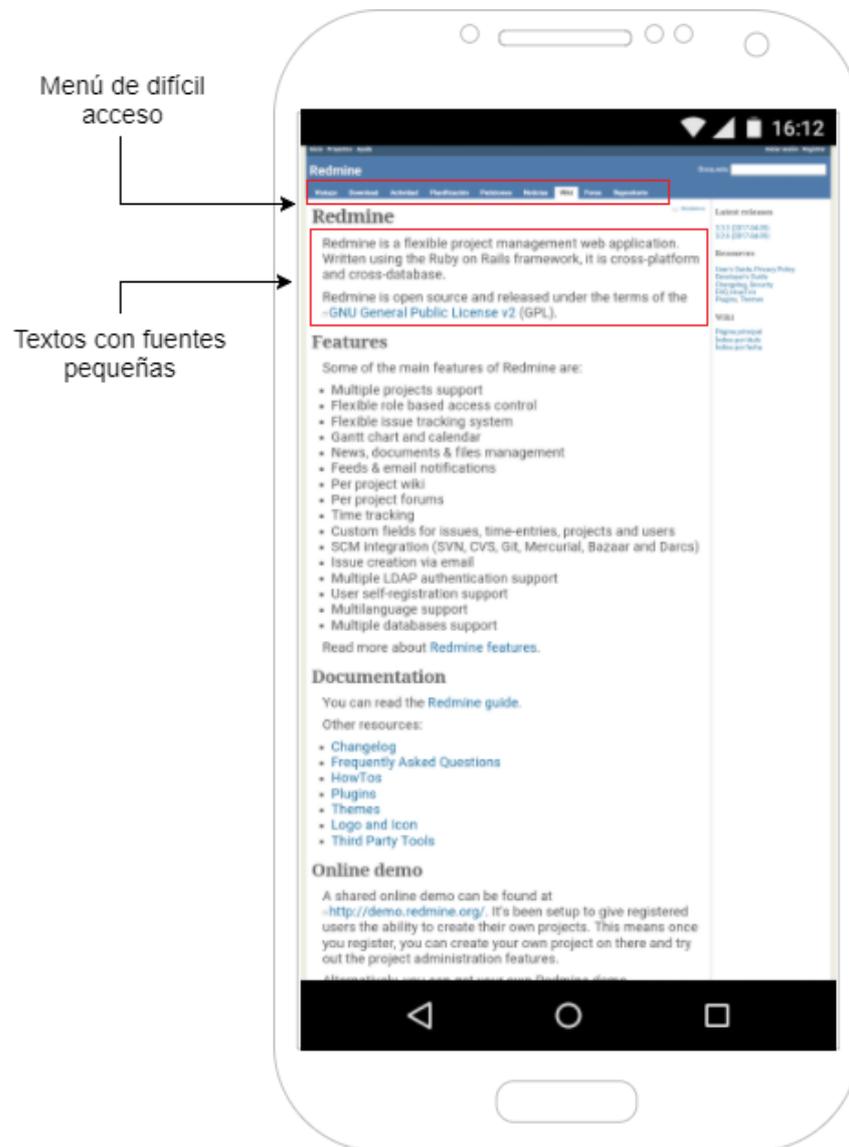


Figura 4.3 - Sitio visto desde celular de 1280x720 píxeles y CSS pixel ratio 2.

4.2. Configuración del template adaptativo y de la URL

Cada página posee la configuración de un template adaptativo y la URL. Para la URL además se configura el tipo de comparación. La página Web portabilizada tendrá asociada un template adaptativo, que será seleccionado con el objetivo de mejorar la presentación de la información o la navegabilidad. Varios factores influyen en la elección: 1) las necesidades de diseño del sitio móvil, 2) qué información es importante resaltar, 3) qué información se va a mostrar en la versión para dispositivos móviles. La configuración de la URL y el tipo de comparación permite definir si el template seleccionado será aplicado a únicamente a la página cuya URL coincida a la URL configurada o a un conjunto de páginas donde la URL

incluya el valor configurado. La herramienta completa por defecto el valor con la URL de la página actual.

Para adaptar la página principal del sitio Redmine se seleccionó el template **celular**, ya que es el template que se adapta mejor al caso de estudio. Luego se configura la URL asociada a la adaptación con el valor **http://www.redmine.org** y el tipo de comparación **igualdad**, como se muestra en la Figura 4.4.

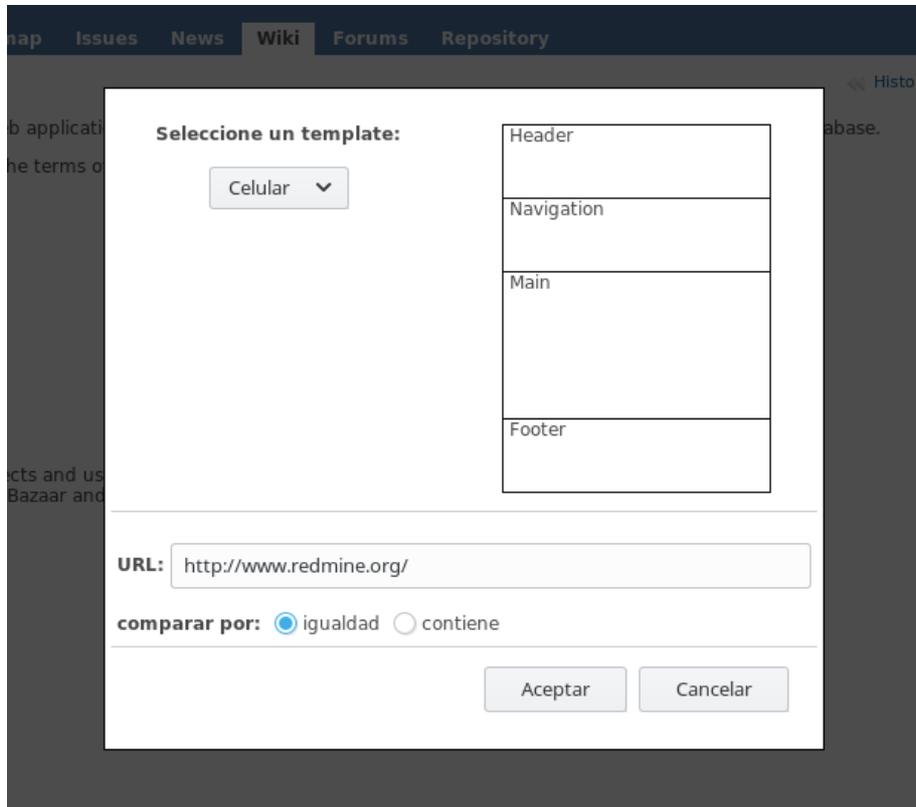


Figura 4.4 - Configuración del template adaptativo y la URL de comparación.

A medida que se avanza con la portabilización, la herramienta va almacenando la información de las adaptaciones en un objeto con formato JSON.

4.3. Selección de elementos y adaptaciones

En esta etapa se van a seleccionar, a través de la herramienta, los elementos de la UI existente para la versión móvil, definir los patrones adaptativos a ejecutar y elegir la posición destino en el template adaptativo. Cada elemento seleccionado de la página es identificado a través expresiones XPath¹⁹, para luego poder acceder a él desde la herramienta para su manipulación. Toda la

¹⁹ <https://www.w3.org/standards/history/xpath-30> - Visitado el Visitado el 01/02/2017

información de la operación se almacena para que posteriormente el módulo de adaptación pueda realizar las acciones definidas.

Con el objetivo de definir la adaptación del menú del sitio Redmine se selecciona a través del mouse el menú principal (ver Figura 4.5).

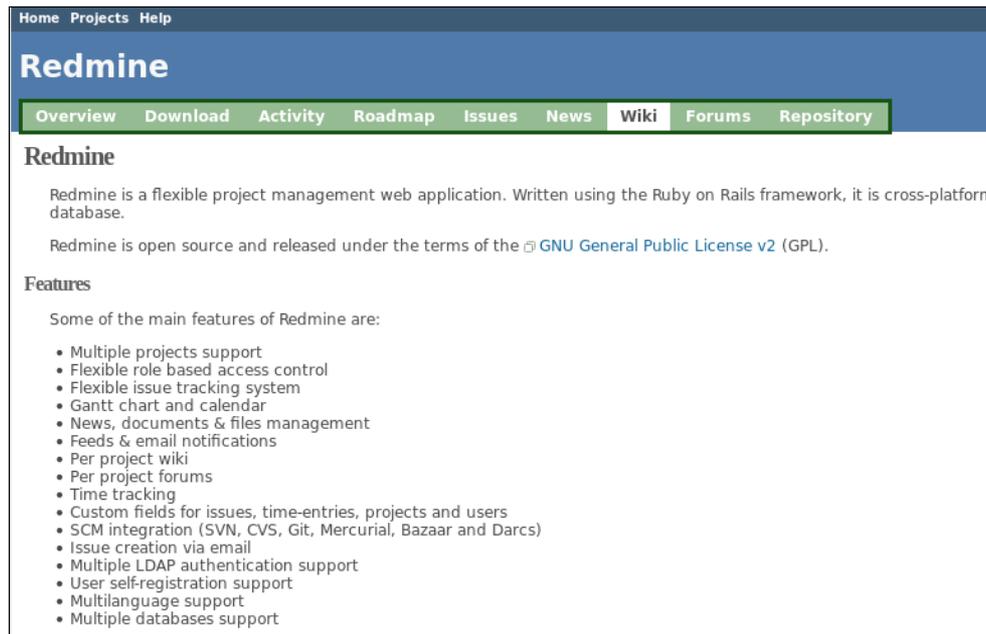


Figura 4.5 - Selección del menú principal de Redmine con la herramienta.

luego se selecciona la opción **Ubicar elemento** (ver Figura 4.6):

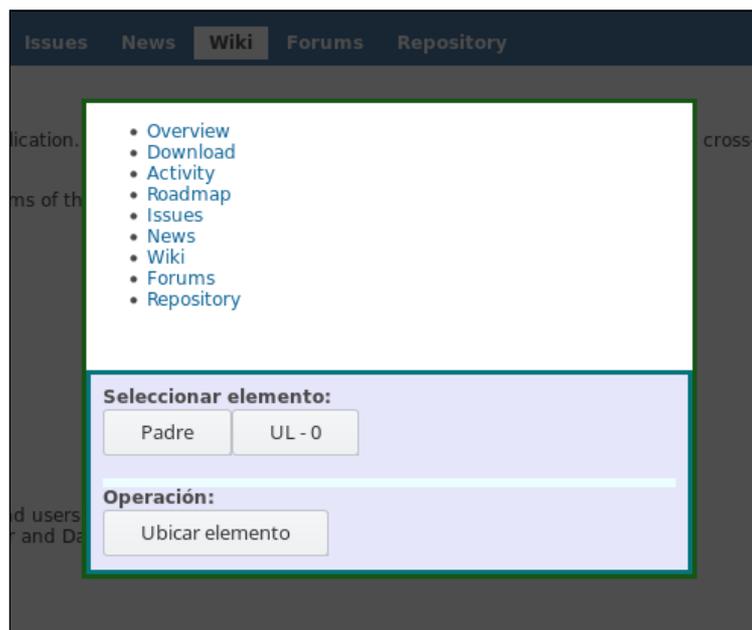


Figura 4.6 - Pantalla con las opciones disponibles al seleccionar un elemento.

finalmente se selecciona **Menú** como patrón adaptativo y la ubicación en el template adaptativo (ver Figura 4.7).

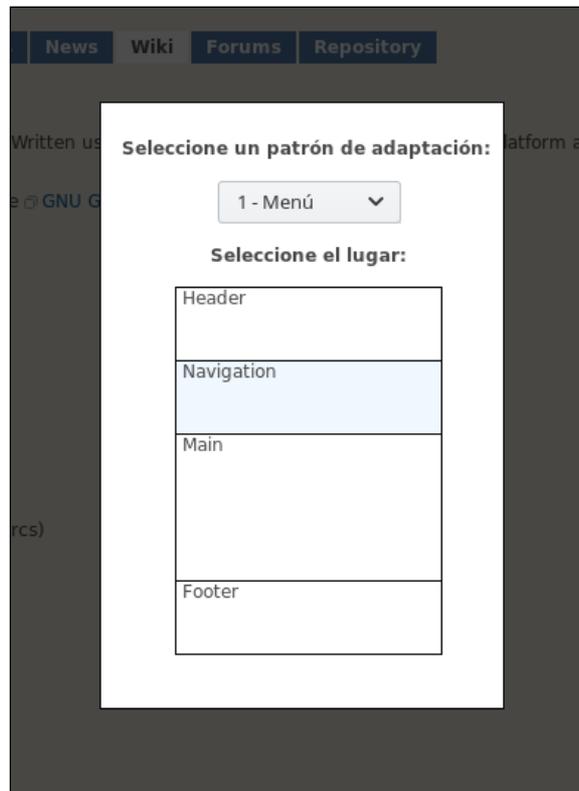


Figura 4.7 - Selección del patrón de adaptación.

Este procedimiento se repite para cada uno de los elementos de la página que van a ser portabilizados a la versión móvil.

4.4. Validación

A medida que se avanza con la portabilización, el previsualizador permite ver en tiempo real cómo se va conformando la página Web adaptada para un dispositivo móvil. Cada vez que se agrega o modifica una adaptación el cambio es visible en el previsualizador que interpreta las adaptaciones definidas por el diseñador. Esto permite validar visualmente el resultado y tomar decisiones respecto al diseño de una manera ágil.

En la siguiente imagen (Figura 4.8) se puede observar el previsualizador y dentro de él la adaptación del menú principal del sitio Redmine:

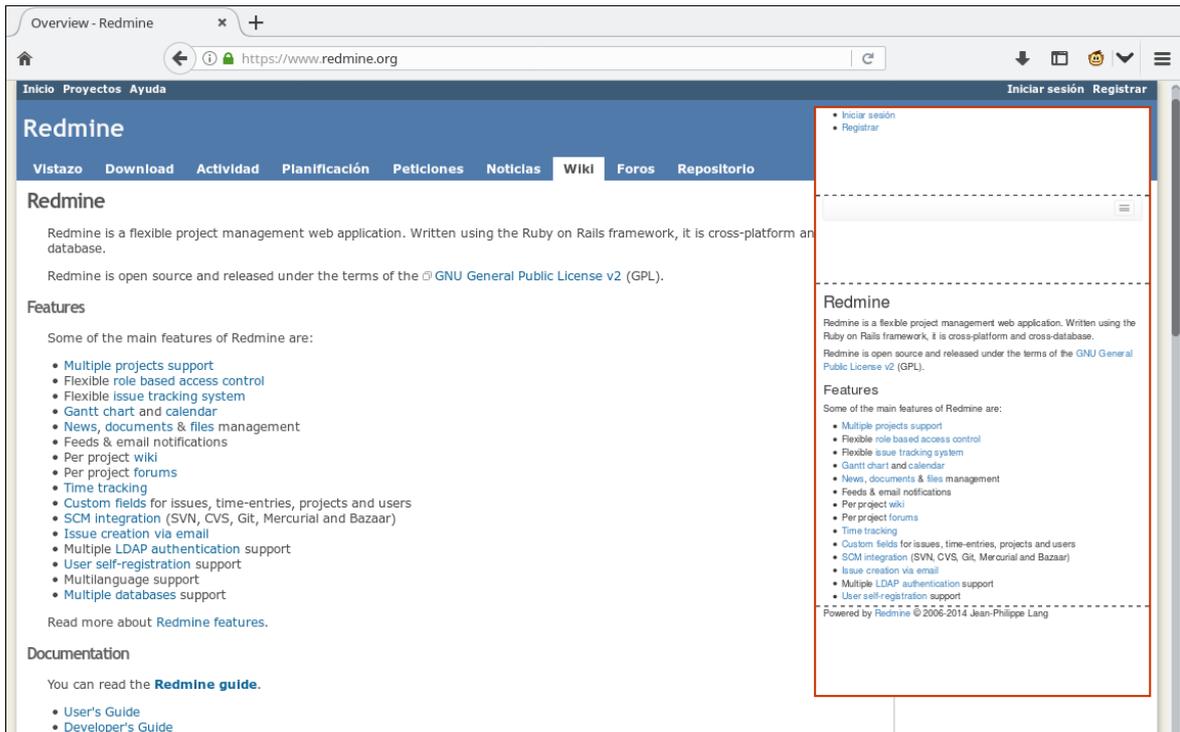


Figura 4.8 - Previsualización en vivo.

4.5. Publicación

Una vez finalizado el diseño del sitio, mediante la función de exportación de la herramienta, se obtiene como resultado un script que realiza la portabilización del sitio en el dispositivo móvil. La siguiente acción es publicar el script en un repositorio. Se puede utilizar un repositorio privado o un repositorio público. Greasy Fork (ver Figura 4.9), es un repositorio público que ofrece un lugar donde compartir scripts y dejar opiniones, además dan un marco de seguridad al usuario del script.

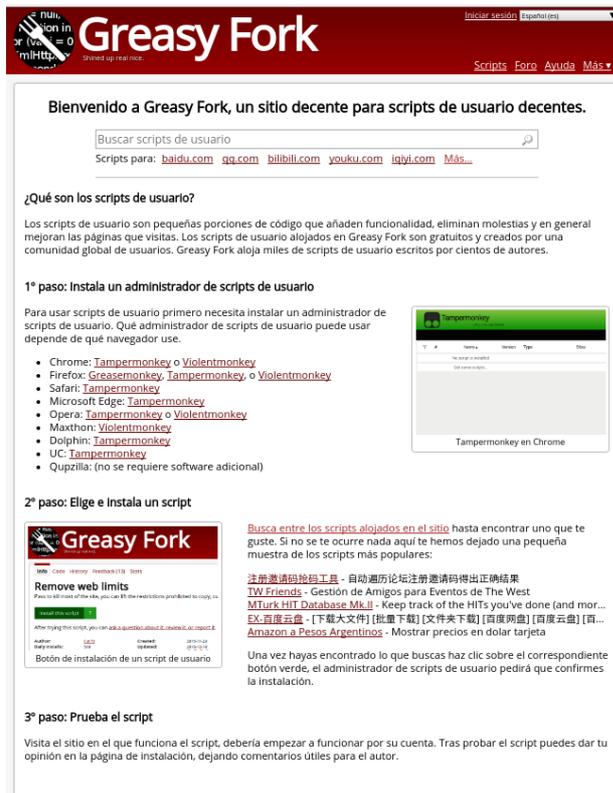


Figura 4.9 - Imagen principal del repositorio de scripts Greasy Fork.

4.6. Instalación del script de portabilización

Los programas administradores de scripts para navegadores Web como Tampermonkey o Greasemonkey permiten al usuario instalar scripts en los dispositivos móviles. Para ver los resultados finales de la portabilización del sitio Redmine se utilizó el navegador Web Dolphin Browser, que soporta el uso de Tampermonkey (ver Figura 4.10).

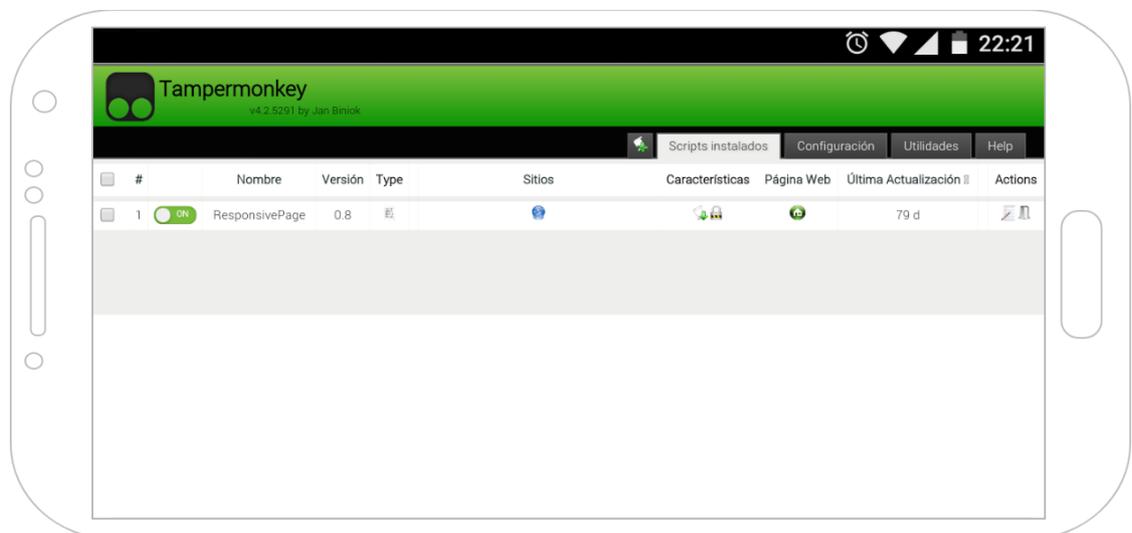


Figura 4.10 - Vista de Tampermonkey en un dispositivo móvil.

4.7. Navegación del sitio Web con UI adaptada

Mientras el usuario del dispositivo móvil navega por un sitio Web portabilizado, el script aplicará automáticamente modificaciones en la UI. Estas modificaciones, reflejarán las configuraciones del template y patrones adaptativos seleccionadas por el diseñador (ver Figura 4.11).

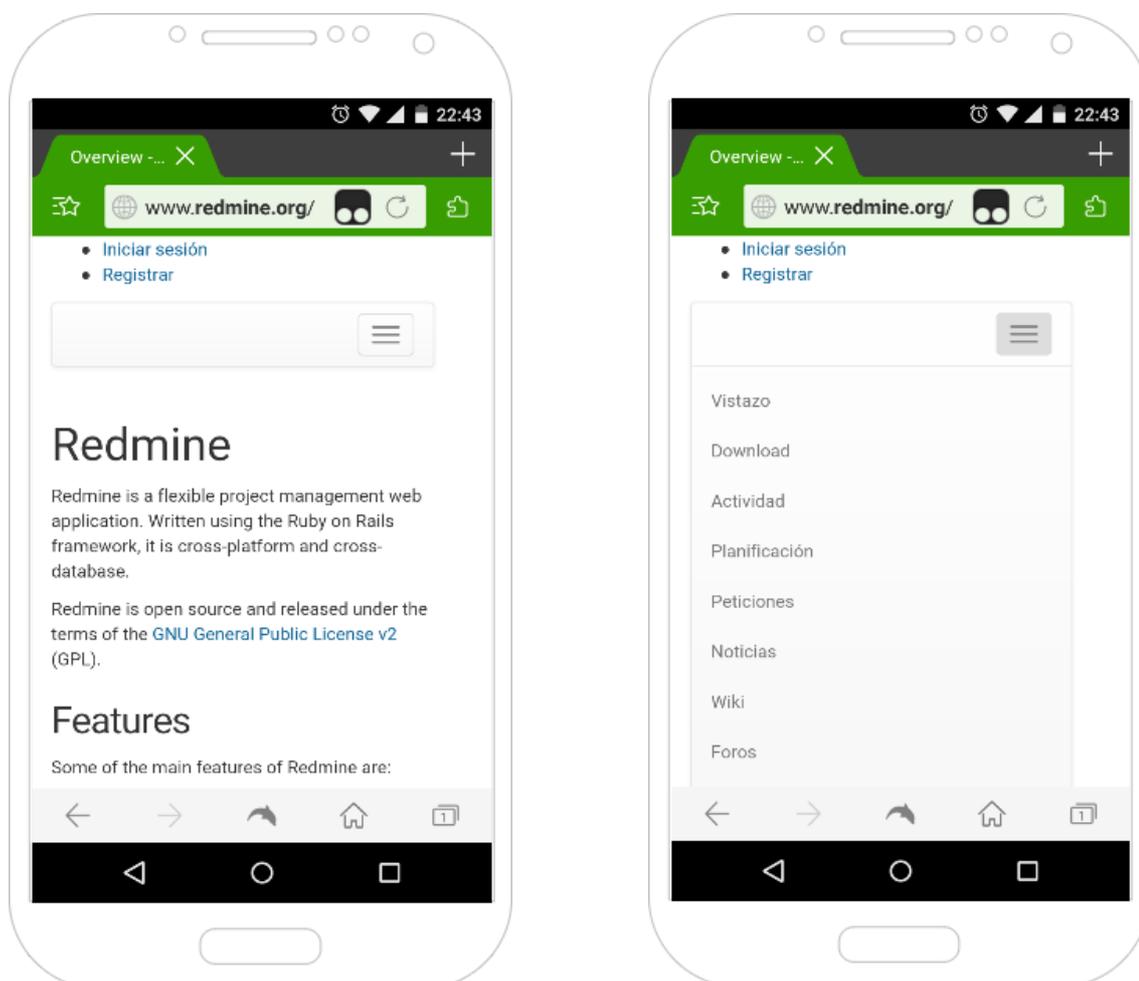


Figura 4.11 - Menú adaptado a un dispositivo móvil.

5. Otro caso de estudio: Pidgin

En esta sección se describe otro ejemplo del proceso de portabilización utilizando la herramienta, las adaptaciones se realizaron sobre el sitio Web de Pidgin²⁰, un cliente de chat de software libre multiplataforma, que da soporte a múltiples protocolos de mensajería. El sitio Web posee características comunes a los sitios Web 1.0 y además no cuenta con una versión para dispositivos móviles. Se realizó una portabilización del sitio Web que incluye la adaptación de 3 páginas: home, register²¹ y about²².

Para realizar las capturas de las diferentes páginas del sitio Web que se presentan a continuación, se utilizó un celular con las siguientes características: pantalla de 4,5 pulgadas, resolución de 1280x720 píxeles y CSS pixel ratio 2.

5.1. Portabilización de la página Home

En la figura 5.1 se muestra la vista que se obtiene al navegar la página home de Pidgin con el dispositivo mencionado. A simple vista, en el caso no portabilizado, puede observarse como el contenido de la página se muestra con un tamaño muy reducido a fin de presentar toda la información en el viewport del celular. El texto resulta ilegible y el acceso al menú principal resulta una tarea difícil, lo que obliga al usuario a realizar gestos de zoom y scroll constantemente para navegar y acceder al contenido de la página Web.

²⁰ <https://www.pidgin.im/> - Visitado el 31/10/2017

²¹ <https://developer.pidgin.im/register> - Visitado el 31/10/2017

²² <https://www.pidgin.im/about/> - Visitado el 31/10/2017

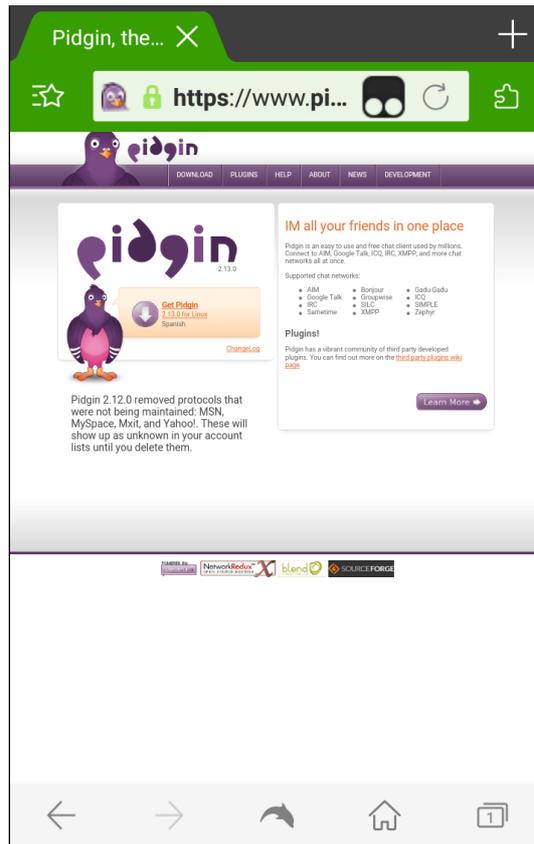


Figura 5.1 - Vista desde el celular de la página Home.

Utilizando la herramienta de portabilización en la página Home del sitio, se configuró el diseño a utilizar, se optó por el template *mobilePhone*, que hace un buen aprovechamiento del espacio en pantallas pequeñas, luego a través de la herramienta de selección se eligieron los elementos HTML para la portabilización y para cada uno de ellos se escogió una adaptación a realizar entre las opciones que provee la herramienta. La figura 5.2 muestra la previsualización que genera la herramienta con las adaptaciones configuradas.

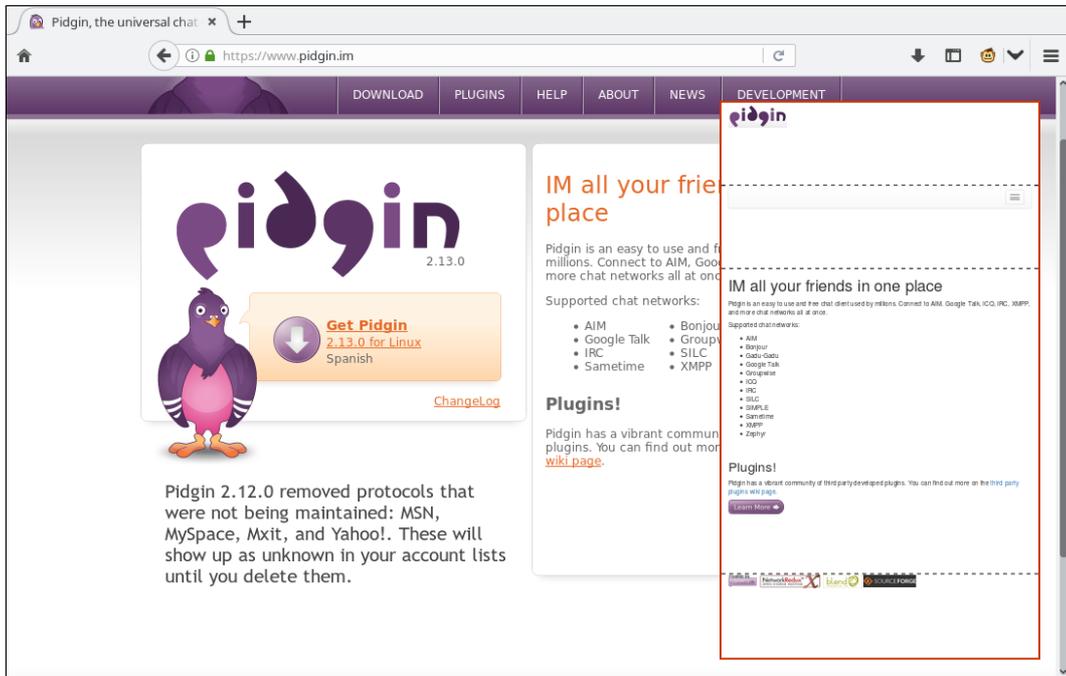


Figura 5.2 - Vista previa de la tarea de portabilización de la página Home.

La figura 5.3 presenta parte del documento en formato JSON con la información de la portabilización que se obtiene con la opción de exportación de la herramienta. En el capítulo 6 (Arquitectura y detalles técnicos de la herramienta) se describe con mayor grado de detalle los atributos del documento JSON.

```
[
  ...
  {
    "url": "https://www.pidgin.im/",
    "url-compare-type": "equal",
    "template": "mobilePhone",
    "page-adaptation": {
      "header-0": {
        "xpath": ["/html/body/DIV[1]/a[1]"],
        "pattern": "copy"
      },
      "navigation-0": {
        "xpath": ["/html/body/DIV[1]/ul[1]"],
        "pattern": "menu"
      },
      "main-0": {
        "xpath": ["/html/body/DIV[2]/DIV[2]/div[1]"],
        "pattern": "copy"
      },
      "footer-0": {
        "xpath": ["/html/body/div[3]"],
        "pattern": "copy"
      }
    }
  }
]
```


vista que se obtiene al navegar la página con el dispositivo de pruebas puede verse en la figura 5.5.

Además de los problemas que se expusieron en la página home, relacionados al tamaño del contenido, esta página presenta una nueva dificultad, el ingreso de datos en el formulario de registro. Debido a la presentación que se obtiene al navegar la página, el usuario se ve obligado a realizar múltiples eventos zoom y scroll para poder efectuar el ingreso de valores en los campos del formulario. Todo esto termina perjudicando la usabilidad de la página, al generar incomodidades y esfuerzo extra por parte del usuario.

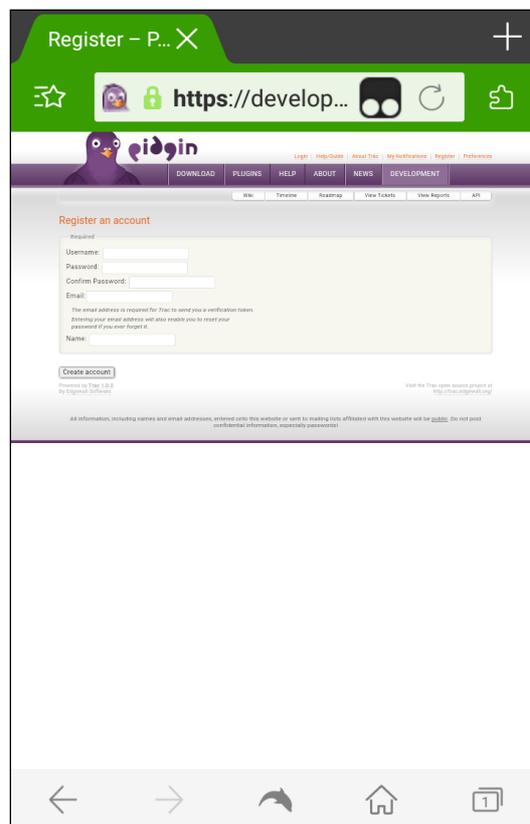


Figura 5.5 - Vista desde el celular de la página Register.

La figura 5.6 muestra la previsualización del resultado de trabajar con la herramienta de portabilización en la página Register, donde lo más destacado fue aplicar una adaptación al formulario, que produce un mejor aprovechamiento del espacio disponible al desplazar las etiquetas sobre los campos de ingreso y ampliar el área de ingreso de los datos.

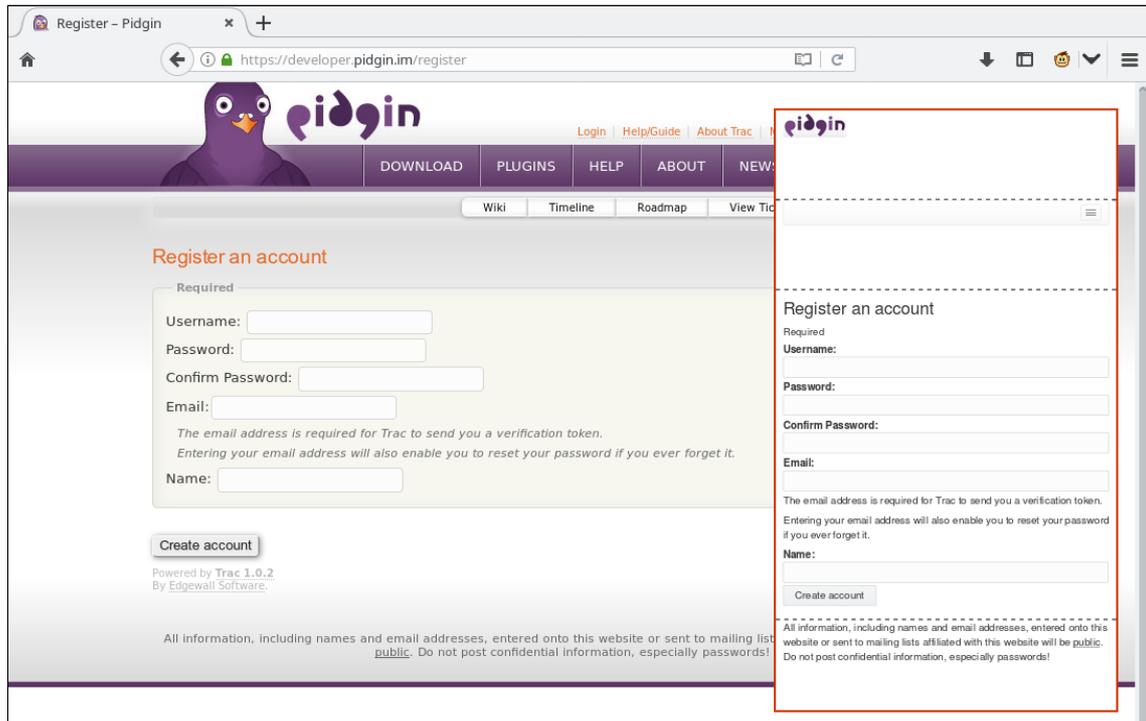


Figura 5.6 - Vista previa de la portabilización de la página Register.

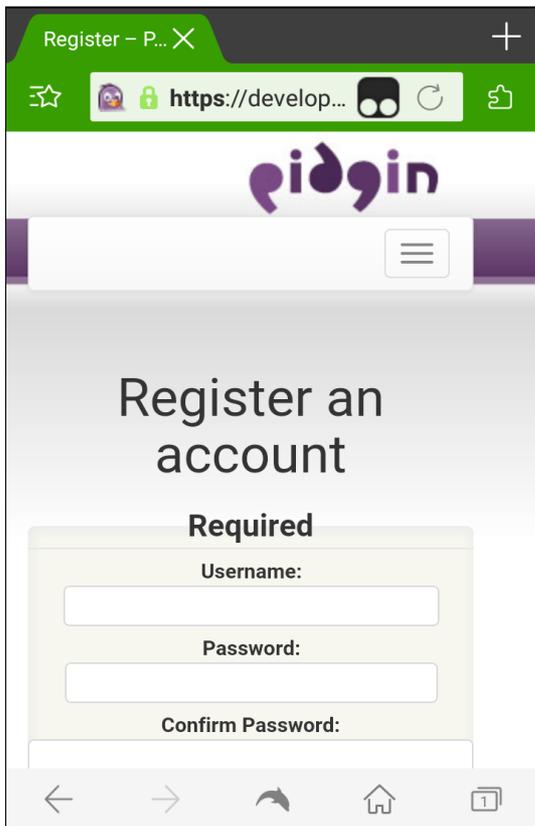
Un fragmento del documento JSON con las configuraciones para esta página se muestra en la figura 5.7.

```
[
  ...
  {
    "url": "https://developer.pidgin.im/register",
    "urlCompareType": "equal",
    "template": "mobilePhone",
    "pageAdaptation": {
      "header-0": {
        "xpath": ["/html/body/DIV[1]/DIV[1]/DIV[1]/a[1]"],
        "pattern": "copy"
      },
      "navigation-0": {
        "xpath": ["/html/body/DIV[1]/DIV[1]/DIV[1]/ul[1]"],
        "pattern": "menu"
      },
      "main-0": {
        "xpath": ["/html/body/DIV[1]/DIV[1]/DIV[4]/div[2]"],
        "pattern": "form"
      },
      "footer-0": {
        "xpath": ["/html/body/DIV[1]/DIV[1]/div[6]"],
        "pattern": "copy"
      }
    }
  }
]
```

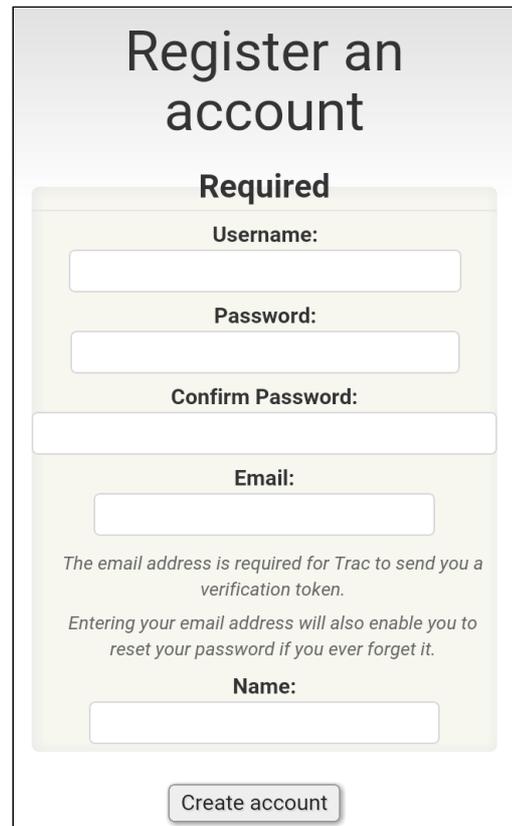


Figura 5.7 - Fragmento del documento JSON de portabilización de la página Register.

La figura 5.8 presenta el resultado final de la portabilización de la página Register visualizada desde el dispositivo móvil de pruebas. En la figura 5.8 (A) vemos la imagen de la página Register apenas ingresamos, luego al hacer scroll y ocultarse automáticamente la barra de direcciones del explorador, se ve el formulario completo, este comportamiento se puede observar en la figura 5.8 (B).



(A)



(B)

Figura 5.8 - Vista desde un celular de la página Register portabilizada.

Los beneficios obtenidos en la portabilización de la página son: 1) mejoras en la legibilidad, 2) mayor espacio para el ingreso de datos en los campos del formulario, 3) no es necesario realizar el gesto de zoom sobre el formulario para visualizar y seleccionar correctamente sus componentes.

5.3. Portabilización de la página About

El proceso de portabilización también se aplicó a la página About. La navegación hacia la página se realiza a través del menú principal de la página Web Home. La figura 5.9 muestra la vista que se obtiene al navegar la página Web desde el dispositivo móvil.

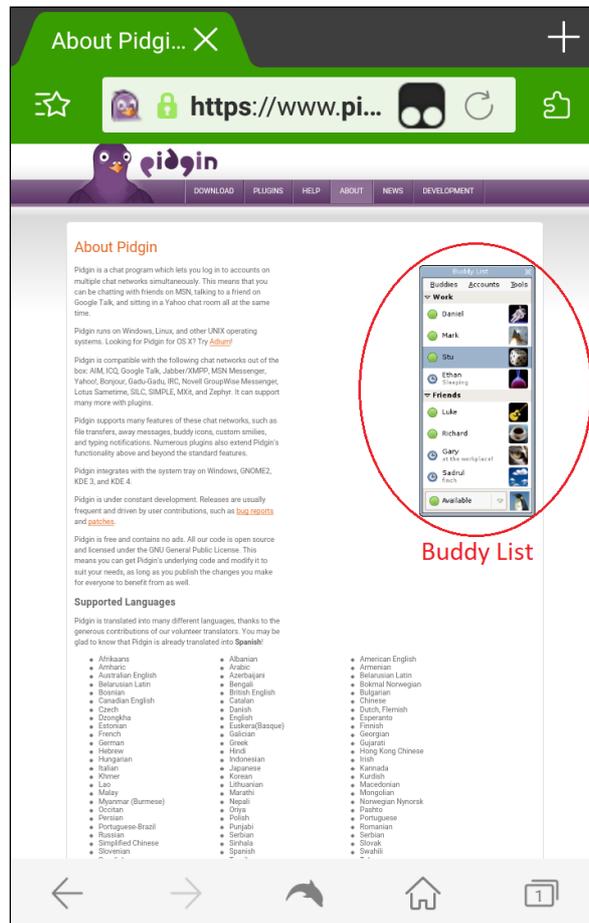


Figura 5.9 - Vista desde el celular de la página About.

Luego, a través de la herramienta se definieron las adaptaciones a realizar, y mediante el componente de previsualización pueden observarse los cambios aplicados a la página Web como muestra la figura 5.10. Estos cambios obedecen a evitar el exceso de información en la página, por lo que se opta por quitar la imagen del “Buddy List” (ver figura 5.12 A y 5.12 B) que no ofrece datos relevantes, pero sobrecarga la UI al usar un celular.

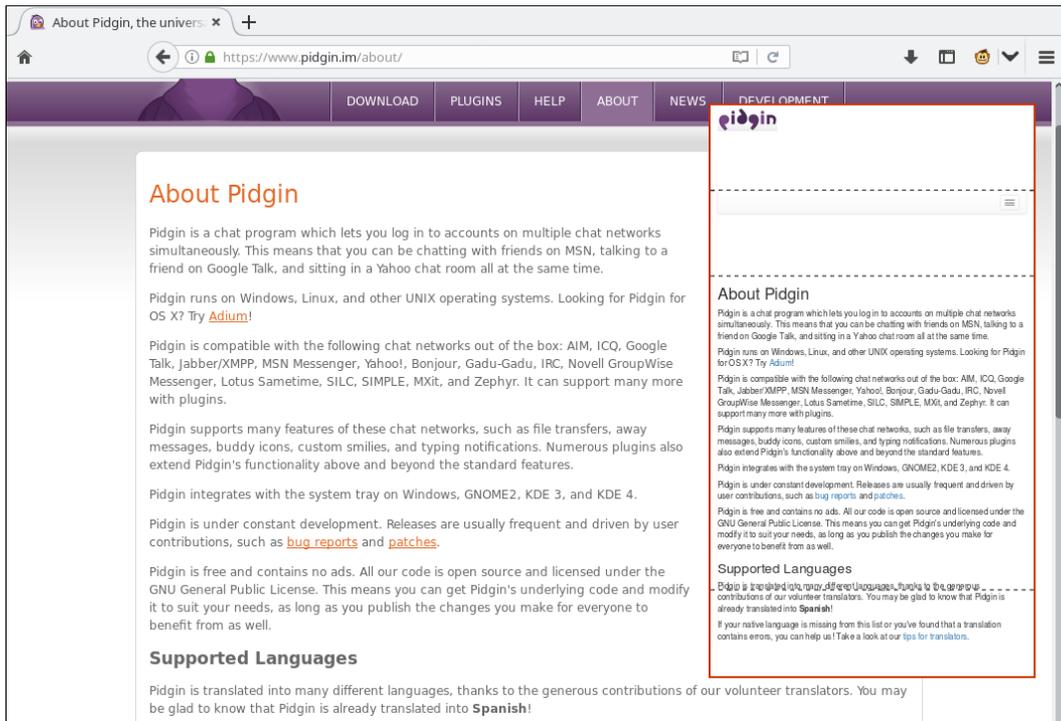


Figura 5.10 - Vista previa de la portabilización de la página About.

Como paso siguiente la configuración es exportada un documento en formato JSON (ver figura 5.11).

```
[
  ...
  {
    "url": "https://www.pidgin.im/about/",
    "urlCompareType": "equal",
    "template": "mobilePhone",
    "pageAdaptation": {
      "header-0": {
        "xpath": ["/html/body/DIV[1]/a[1]"],
        "pattern": "copy"
      },
      "navigation-0": {
        "xpath": ["/html/body/DIV[1]"],
        "pattern": "menu"
      },
      "main-0": {
        "xpath": ["/html/body/DIV[2]/DIV[1]/div[1]"],
        "pattern": "copy"
      },
      "footer-0": {
        "xpath": ["/html/body/div[3]"],
        "pattern": "copy"
      }
    }
  }
]
```

```
    "pattern": "copy"
  }
}
...
]
```

Figura 5.11 - Fragmento del documento JSON de portabilización de la página About.

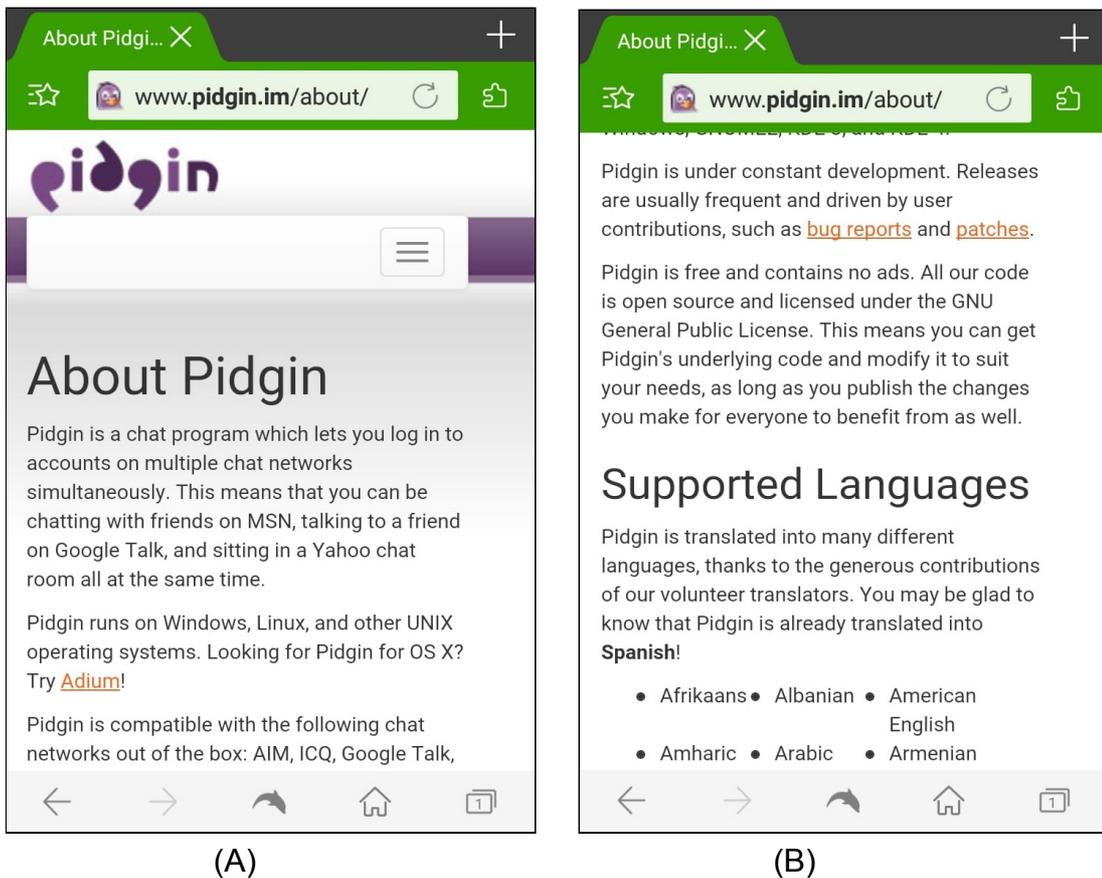


Figura 5.12 - Vista desde el celular de la página About portabilizada.

Lo más relevante obtenido con la portabilización fueron: 1) mejoras en la legibilidad, 2) la selección del contenido, se pudo elegir el contenido importante a destacar y eliminar la información no relevante, en este caso la imagen con el “Buddy List”.

6. Arquitectura y detalles técnicos de la herramienta

La implementación de la herramienta de portabilización posee tecnologías ampliamente adoptadas por la comunidad de desarrollo Web tal como Javascript, JQuery, HTML5, CSS y Twitter Bootstrap (Figura 6.1). Para su instalación en un dispositivo móvil es necesario el uso de un administrador de scripts como Greasemonkey o Tampermonkey. La herramienta está formada por dos componentes principales, el Diseñador de Adaptaciones y el Motor de Adaptaciones, que en conjunto implementan la portabilización.



Figura 6.1 - Tecnologías utilizadas en la implementación de la herramienta.

6.1. Diseñador de Adaptaciones

El Diseñador de Adaptaciones permite definir los ajustes para obtener una versión móvil de un sitio Web. El usuario responsable de realizar la portabilización trabaja con la herramienta de selección y navegación de elementos del DOM (por ej., navegación a través del elemento padre y hermanos), para integrarlos en un nuevo template adaptativo. La herramienta provee también el soporte de validación en vivo, mediante el previsualizador, y permite la exportación de la configuración para su publicación. A continuación se describen las funcionalidades principales.

6.1.1. Selección de template adaptativo

La herramienta permite optar entre varias opciones de templates adaptativos, dependiendo del tamaño del viewport del dispositivo y las necesidades de diseño del sitio Web resultará más conveniente un template u otro.

Templates:

- A. **Template Celular:** posee una única columna y 4 filas, por lo que resulta útil para dispositivos con un viewport de tamaño pequeño. Ver figura 6.2.



Figura 6.2 - Template Celular.

- B. **Template Tablet:** tiene dos columnas y 4 filas. Se ajusta mejor a dispositivos móviles con un viewport de tamaño medio. Ver figura 6.3.

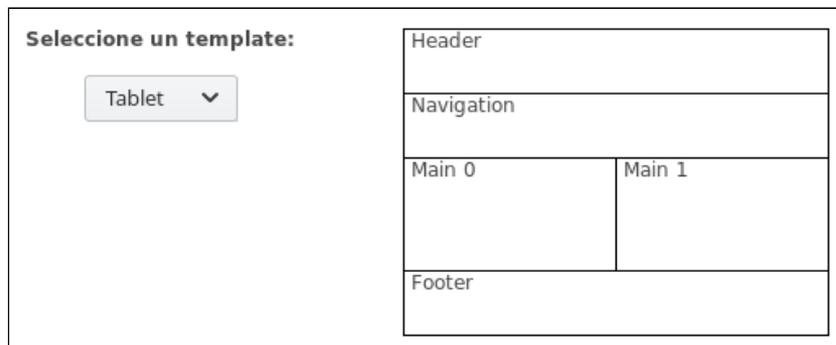


Figura 6.3 - Template Tablet.

- C. **Template Genérico:** cuenta con 3 columnas y 4 filas, permitiendo una mayor flexibilidad en el diseño del sitio que se está portabilizando. Ver figura 6.4.

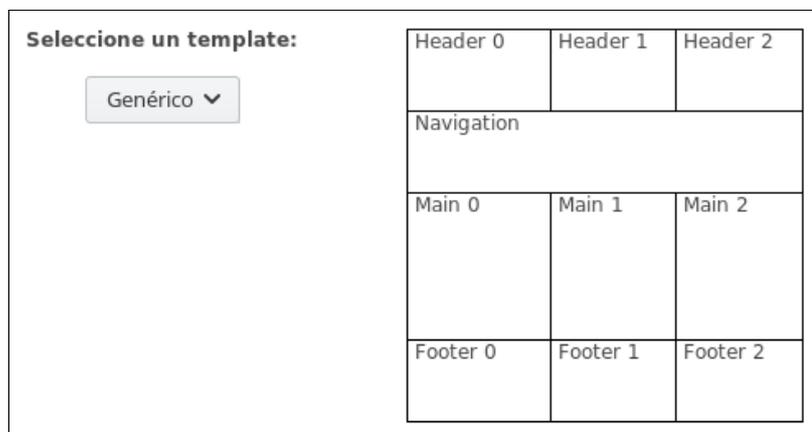


Figura 6.4 - Template Genérico.

Una vez seleccionado un template se actualiza la propiedad "template" en el objeto JSON con la portabilizacion del sitio Web. Ver figura 6.5.

```
{
  ...
  "template": "mobilePhone",
  ...
}
```

Figura 6.5 - Fragmento del documento JSON. Configuración de template.

6.1.2. Configuración de URL

La portabilización de un sitio Web implica portabilizar sus páginas Web con la herramienta, por lo que es necesario identificar las adaptaciones de cada página con una URL en particular. Por defecto la herramienta completa el valor para la URL con la ruta completa de la página Web actual, pero este valor puede modificarse de acuerdo a las necesidades del diseñador. Luego hay que seleccionar que tipo de comparación se va a realizar entre las URLs, los valores posibles son por "igualdad" o "contiene". Generalmente tendremos una relación uno a uno, entre la portabilización de una página Web y una URL, en este caso se utiliza la comparación "igual", pero existen situaciones donde resulta útil reutilizar la misma portabilización en páginas Web con similar diseño y URLs dentro del mismo sitio, en este caso se utiliza la opción de comparación de URL "contiene".

Por ej. si tenemos una URL **http://www.redmine.org/issues/[A]**, con una parte dinámica, donde **[A]** puede tomar valores como 25239 o 25238, los valores a utilizar son (ver figura 6.6):

- URL: **http://www.redmine.org/issues/**
- tipo de comparación: **comparar por contiene**



URL:

comparar por: igualdad contiene

Figura 6.6 - Configuración de URL.

Los valores seleccionados se almacenan en las propiedades "url" y "urlCompareType" del objeto JSON, con la portabilizacion del sitio Web (figura 6.7).

```
{
  ...
  "url": "http://www.redmine.org/issues",
  "urlCompareType": "contain",
  ...
}
```

```
}
}
```

Figura 6.7 - Fragmento del documento JSON. Configuración de url y urlCompareType.

6.1.3. Selección de elementos, patrón adaptativo y ubicación

Se implementaron funciones para la selección de elementos HTML de la página Web, a través del puntero del mouse (figura 6.8). Los elementos seleccionados son remarcados en un recuadro con fondo de color verde.

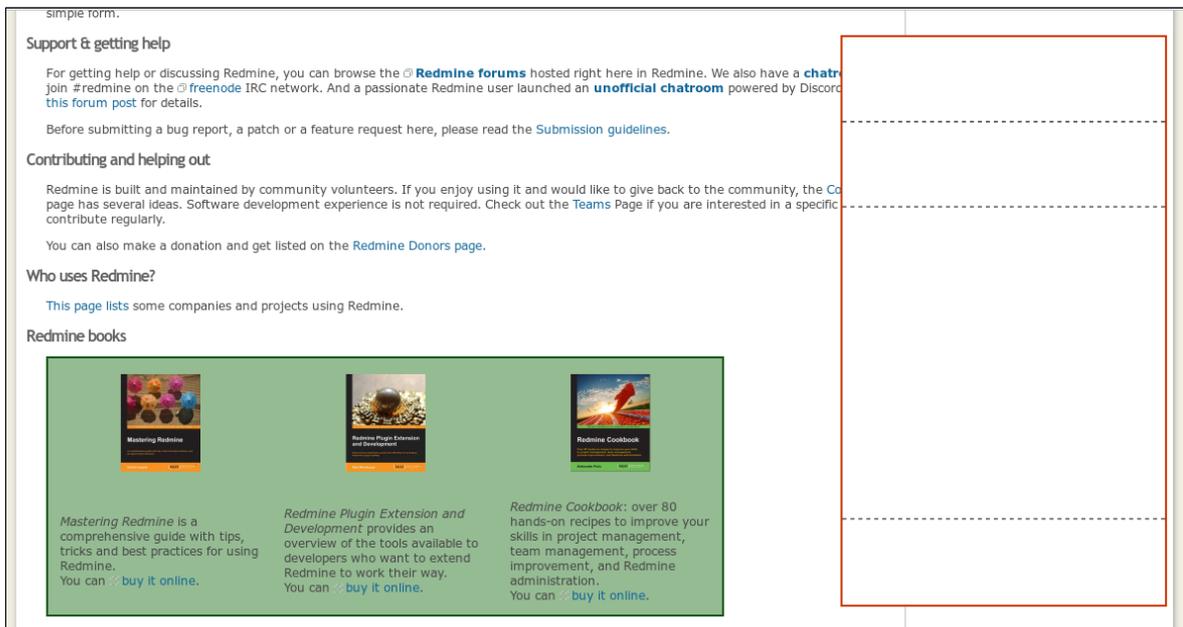


Figura 6.8 - Selección de elementos.

Una vez seleccionado un elemento, la herramienta permite la navegación entre los elementos del DOM relacionados, hermanos y padre (figura 6.9).



Mastering Redmine is a comprehensive guide with tips, tricks and best practices for using Redmine. You can [buy it online](#).

Redmine Plugin Extension and Development provides an overview of the tools available to developers who want to extend Redmine to work their way. You can [buy it online](#).

Redmine Cookbook: over 80 hands-on recipes to improve your skills in project management, team management, process improvement, and Redmine administration. You can [buy it online](#).

Seleccionar elemento:

Padre TBODY - 0

Operación:

Ubicar elemento

Figura 6.9 - Opciones al seleccionar un elemento.

Luego, se debe optar por un patrón de adaptación y una ubicación en el template adaptativo (figura 6.10). La herramienta obtiene la referencia a el elemento del DOM con una expresión XPath y almacena la información en el objeto JSON con la portabilización.

Seleccione un patrón de adaptación:

Menú ▾

Seleccione el lugar:

Header
Navigation
Main
Footer

Figura 6.10 - Selección de patrón y ubicación en el template adaptativo.

Finalmente, se actualizan las propiedades “xpath” y “pattern” correspondientes a la ubicación elegida, como destino en el template adaptativo dentro del objeto JSON con la portabilización del sitio Web (figura 6.11).

```
{
  ...
  "navigation-0":{
    "xpath":"/html/body/DIV[1]/DIV[1]/DIV[1]/DIV[2]/div[2]",
    "pattern":"menu"
  }
  ...
}
```

Figura 6.11 - Fragmento del documento JSON. Configuración de xpath y pattern.

6.1.4. Previsualización

Durante la etapa de diseño de la versión para dispositivos móviles, el Motor de Adaptaciones ejecuta las transformaciones sobre el documento HTML del previsualizador (figura 6.10), a medida que se definen los patrones de adaptación a ejecutar. Luego el previsualizador muestra el resultado en una ventana que simula la pantalla de un dispositivo móvil, permitiendo al diseñador ver una aproximación del resultado rápidamente y realizar modificaciones si lo considera necesario.

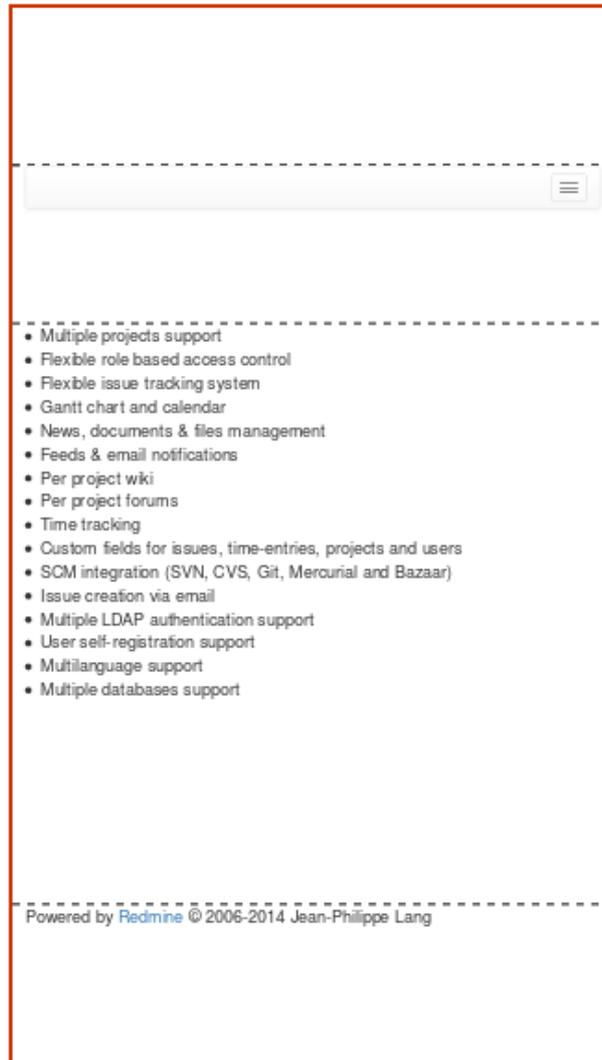


Figura 6.10 - Pantalla de previsualización.

6.1.5. Exportación e importación de la configuración

La herramienta cuenta con opciones para importar (figura 6.11) y exportar (figura 6.12), utilizando un documento JSON, las adaptaciones realizadas a un sitio Web. Esto permite al diseñador transportar su trabajo en diferentes ambientes donde cuenta con la herramienta y continuar con la portabilización. Luego, una vez que tiene lista la versión final de la portabilización, la misma puede ser provista a los usuarios finales, los cuales usaran la función de importación para ver portabilizado el sitio Web.

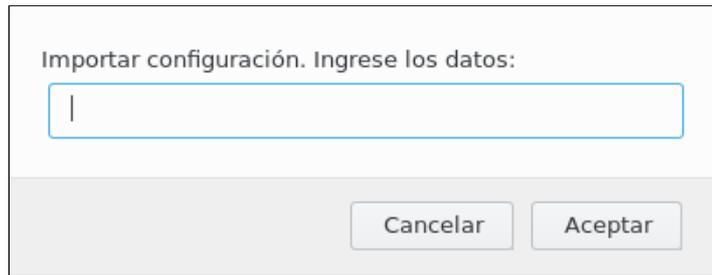


Figura 6.11 - Importación de la configuración de un sitio Web.



Figura 6.12 - Exportación de la configuración de un sitio Web.

6.1.6. Persistencia de datos

La información recolectada durante el proceso de portabilización es almacenada en variables en memoria, utilizando objetos JSON. Luego, éstos son serializados y persistidos en el *Local Storage* del navegador Web utilizando comandos Javascript estándar (figura 6.13). Este tipo de almacenamiento sólo admite datos de tipo String, por lo que la herramienta realiza la conversión de la información desde objetos JSON a Strings para la persistencia, y la conversión de String a objetos JSON cuando se recuperan los datos para ser manipulados.

```
...
localStorage.setItem("portabilizacion",
JSON.stringify(portabilizacion));
...
JSON.parse(localStorage.getItem("portabilizacion"));
...
```

Figura 6.13 - Almacenamiento y recuperación de la información de la portabilización.

6.2. El Motor de Adaptaciones

El Motor de Adaptaciones es el encargado de realizar transformaciones en el documento HTML de la página Web, con el fin de obtener una versión mejorada para dispositivos móviles.

Las tareas que realiza son:

1. Recibe la información en formato JSON generada en el Diseñador de Adaptaciones, a través de la operación de importación.
2. Procesa los datos, obtiene las adaptaciones a realizar, modela la información en memoria y persiste los datos en el navegador Web haciendo uso del Local Storage.
3. Ejecuta las transformaciones y actualiza el DOM de la página Web.

Para implementar las adaptaciones se incorporan en la página Web las dependencias de las librerías jQuery y Bootstrap, que proveen de funcionalidad útil para realizar las adaptaciones, como el manejo del DOM, estilos CSS adaptativos y media queries. La ejecución de una adaptación implica realizar acciones como incorporar código a nivel de hojas de estilos (CSS), modificar elementos del DOM HTML y agregar funciones JavaScript.

6.2.1. Patrones de adaptación

La herramienta implementa un conjunto de patrones de adaptación con el fin de mejorar la usabilidad de la aplicación Web en dispositivos móviles. Estos están orientados a resolver casos concretos que se presentan al navegar un sitio Web 1.0 en un dispositivo móvil. Para realizar esta adaptación opcionalmente se transforma el DOM obtenido de la página original.

6.2.1.1. Menú

Adapta un menú de opciones que posee un conjunto de links HTML en un nuevo menú desplegable verticalmente, conocido también como menú hamburguesa, que optimiza el espacio en pantalla y mejora la usabilidad. Los conceptos de esta implementación están relacionados con el patrón Toggle Menu mencionado en la subsección 3.2.5.2.

En la figura 6.14 puede observarse el menú antes de realizar la adaptación. En la figura 6.15 (A) vemos el menú adaptado en forma de menú hamburguesa sin desplegar y en la figura 6.15 (B) el menú hamburguesa desplegado.

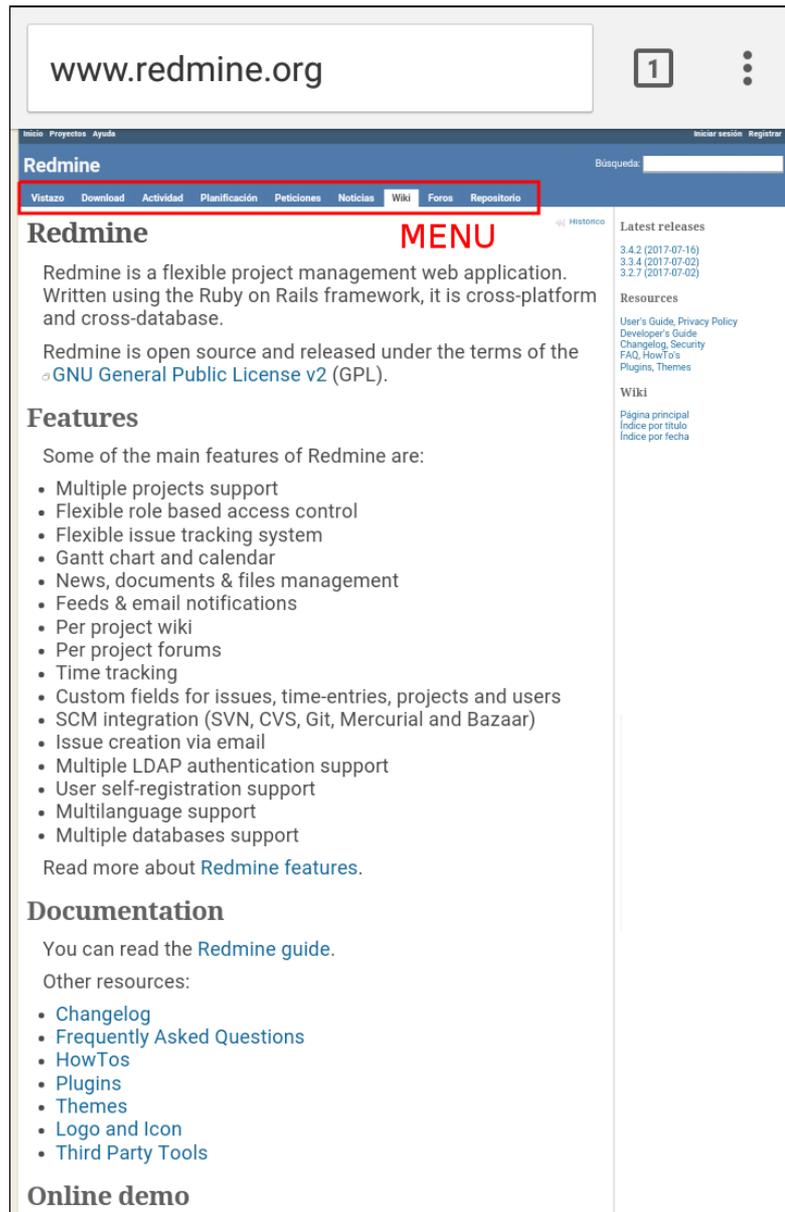
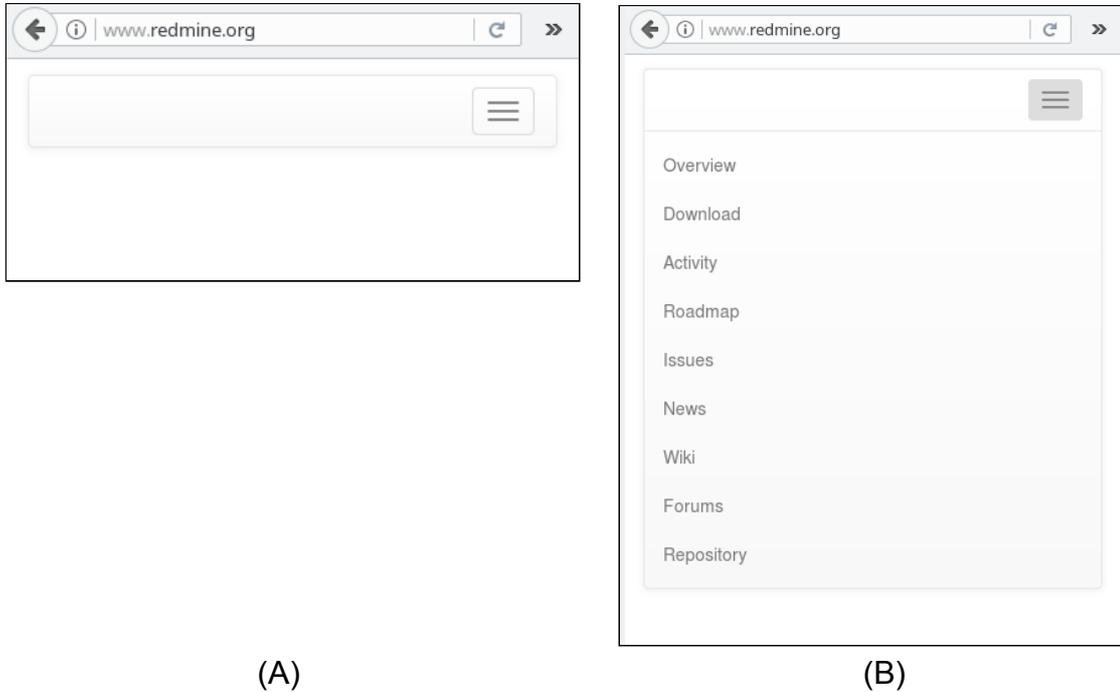


Figura 6.14 - Menú antes de la adaptación visto en un celular.

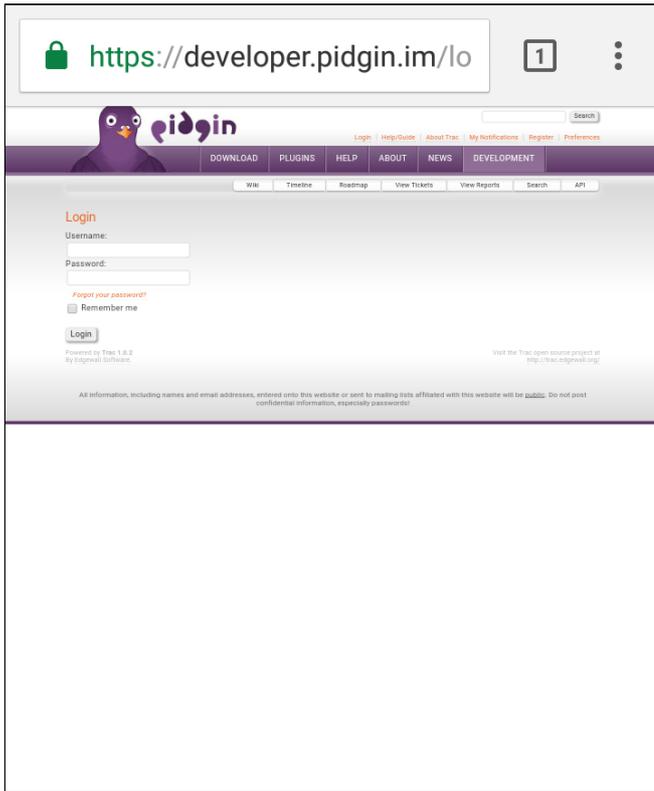


(A) (B)
Figura 6.15 - Menú adaptado a un dispositivo móvil.

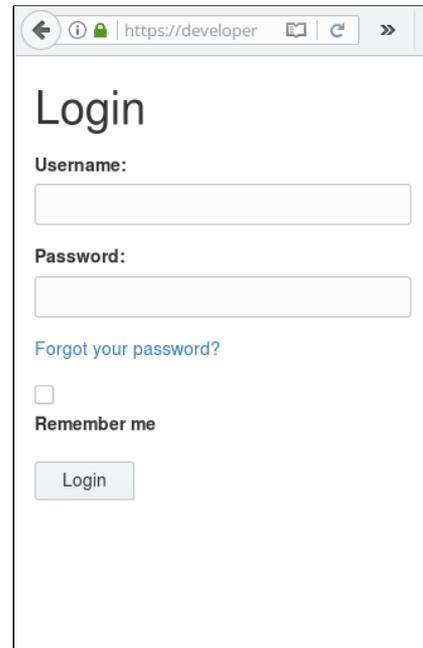
6.2.1.2. Formulario

Este patrón de adaptación está basado en el patrón Form (subsección 3.2.5.5). La adaptación para formularios aplica estilos CSS para mejorar la visualización de los campos. Para ello, se definen e incorporan nuevos estilos a la página Web, que aprovechan el ancho de la pantalla para mostrar la información.

La figura 6.16 (A) muestra el antes y la figura 6.16 (B) muestra el después de la adaptación vista desde el celular.



(A)



(B)

Figura 6.16 - Formulario visto en el celular.

6.2.1.3. Títulos

Utilizando la base del patrón Text (subsección 3.2.5.6), con el fin de aumentar la legibilidad y principalmente destacar el texto seleccionado, se implementó el patrón Títulos. En la versión móvil de un sitio puede ser necesario destacar algún texto en particular, para ello implementamos este patrón que aumenta el tamaño de la fuente y el grosor del trazo (figura 6.17). Para su implementación se definen estilos CSS que se incorporan a la versión móvil del sitio.

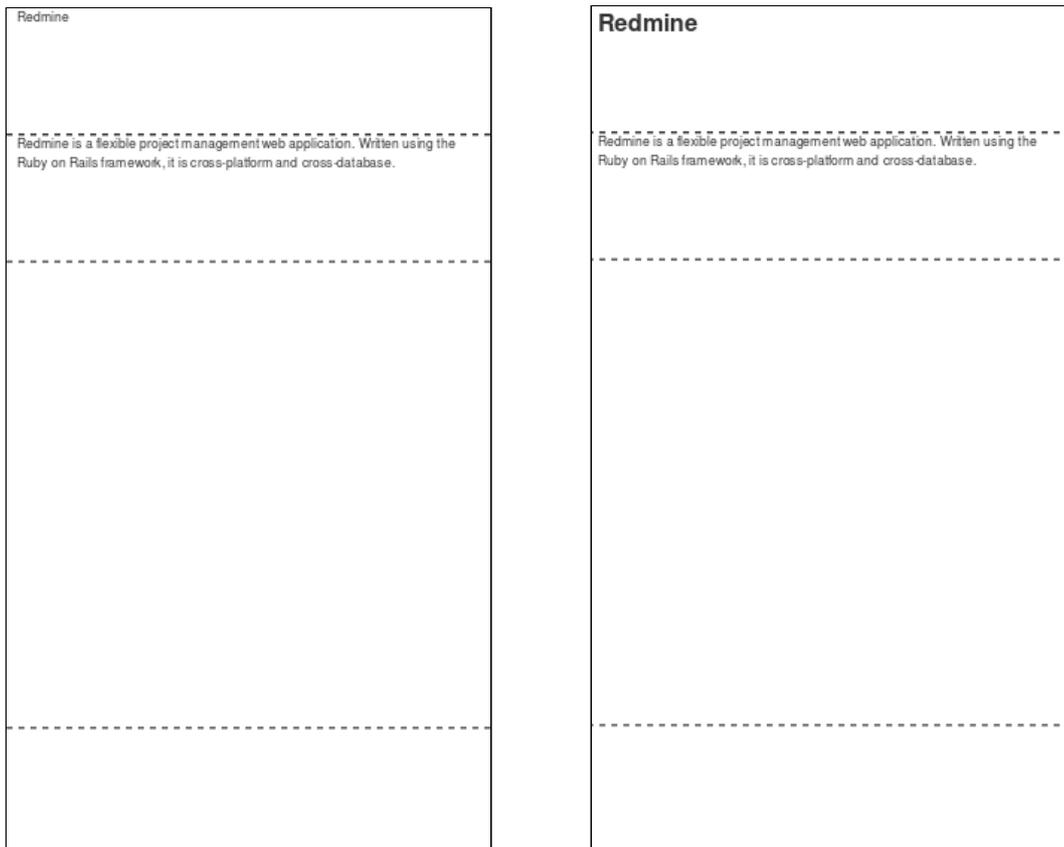


Figura 6.17 - Efecto de aplicar el patrón Títulos.

6.2.1.4. Copiar

La funcionalidad “copiar” que provee la herramienta no es un patrón de adaptación, pero se agregó a la herramienta por su utilidad. Esta operación no realiza ninguna modificación del elemento seleccionado, sólo define que el elemento se debe copiar en la nueva UI, manteniendo las propiedades HTML y CSS originales. En la figura 6.18 podemos observar en el previsualizador el resultado de “copiar” una sección de la página Home de Redmine.

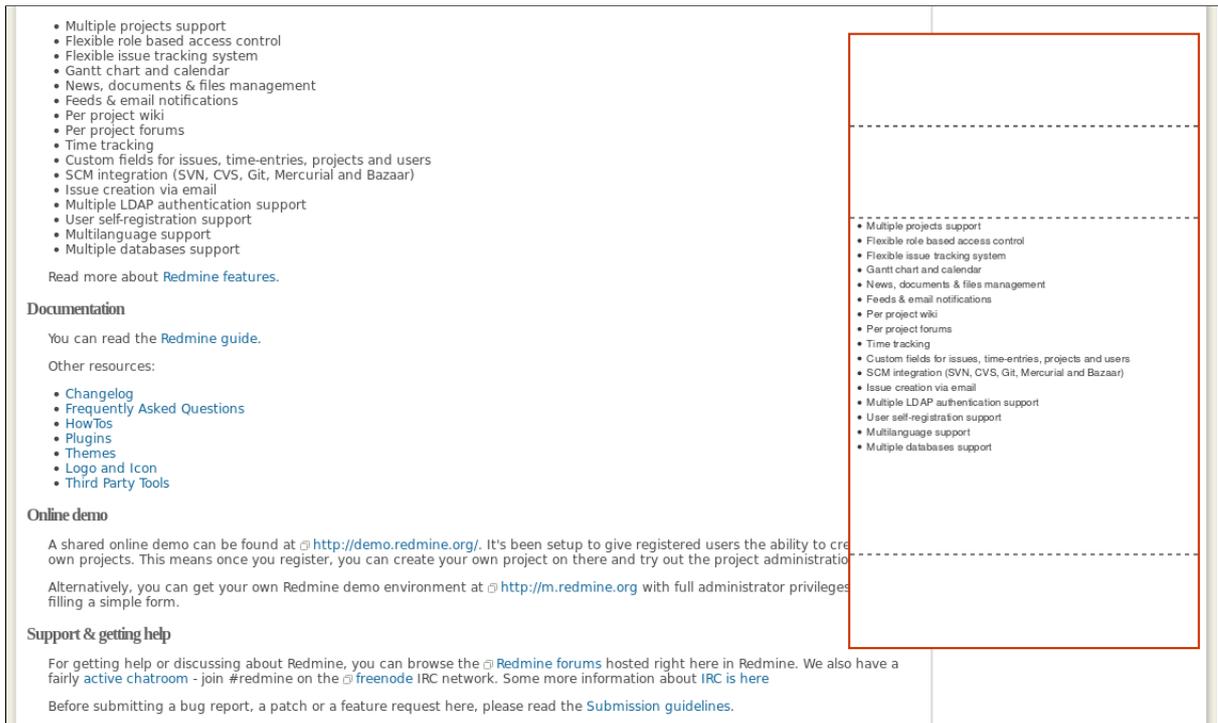


Figura 6.18 - Utilización de la opción copiar en el template adaptativo.

6.2.2. Importación de la portabilización

Una vez que tenemos las adaptaciones sobre los sitios Web, la herramienta permite importar la portabilización a través de un documento JSON. Este documento es obtenido por medio de la función de exportación del Diseñador (ver subsección 6.1.5). En el Anexo A.2 se explica en detalle este proceso de importación.

7. Encuestas y Evaluación

En este capítulo se describe la metodología utilizada para realizar las encuestas, la planificación del experimento, la obtención y el análisis de los datos. Todo esto es realizado a fin de evaluar la aplicabilidad y efectividad de la propuesta planteada en este trabajo. Los participantes de las encuestas se tomaron de manera aleatoria con el objetivo de tener un panorama similar a lo que sucede, en general, con cualquier página Web. Cada encuesta constaba de dos tareas específicas sobre una página Web, de las distintas interacciones realizadas para resolver las mismas se obtuvieron los datos. Estos datos fueron digitalizados y analizados como se detalla a continuación.

Cabe mencionar que los primeros esbozos de encuestas se realizaron en la Facultad de Informática de La Plata (UNLP), en el marco de la “Expo Ciencia y Tecnología 2016”, como parte del enfoque “Métodos y herramientas para portabilizar aplicaciones Web Convencionales”. Esto nos sirvió de sondeo general para exponer la herramienta de portabilización, realizar una breve entrevista a los interesados y retroalimentar el desarrollo de la herramienta. De esta experiencia se obtuvieron buenas ideas y mejoras para la misma. Adicionalmente se logró refinar las encuestas y el modo de llevarlas a cabo, lo que fue sumamente enriquecedor para la realización de las encuestas definitivas que se presentan a continuación.

7.1. GQM

Con el fin de evaluar la utilidad de la herramienta de adaptación aplicada a un sitio Web, se optó por el método de GQM [6, 7] ya que se consideró el más adecuado por estar orientado a objetivos. Un breve resumen de la metodología GQM se encuentra en el Anexo 2 para más detalles. Siguiendo los lineamientos de GQM definimos el objetivo del experimento de la siguiente manera:

Medir la mejora de la herramienta con el propósito de capturar oportunidades desde la perspectiva de un usuario final.

Luego, definimos las diferentes preguntas que permitieron realizar el análisis, éstas se encuentran detalladas en el “Modelo de encuesta” subsección 7.2.3. Las mismas hacen hincapié en la usabilidad, en la experiencia de usuario, la mejora tanto en los tiempos de respuesta, así como en la reducción de interacciones necesarias para realizar una tarea.

Las métricas que se van a usar son:

1. El tiempo empleado para realizar las tareas.
2. La satisfacción por parte del encuestado, medida del 1 al 5.
3. Las interacciones del usuario, medidas mediante clicks, scrolls y zooms.
4. Cantidad de opiniones negativas sobre el tamaño de fuente o imagen, exceso de contenido y manejo del menú.

7.2. Planificación del Experimento

Esta sección describe los elementos involucrados en la planificación y realización del experimento, el procedimiento utilizado y analiza los resultados obtenidos [16].

7.2.1. Metas

La meta principal es determinar si existen beneficios al utilizar la herramienta de augmentation en aplicaciones Web desde dispositivos móviles. Se realizó un experimento controlado que incluyó 4 actividades a realizar, con el objetivo de determinar la eficiencia, productividad y satisfacción desde el punto de vista de un usuario final. Para esto se tuvieron en cuenta el tiempo, las interacciones realizadas por los encuestados en la realización de las tareas propuestas y la satisfacción obtenida al finalizar el experimento.

7.2.2. Participantes

Para evaluar el impacto que tiene en el usuario final el uso de una herramienta de portabilización se realizó una serie de encuestas que incluyeron a personas de distintas edades y niveles de educación, con y sin conocimientos informáticos. Con esta motivación esperamos investigar sobre la experiencia real del usuario, posea estudios en sistemas o no. Esto es, su impresión final tras haber estado trabajando con sitios Web no adaptativos y si el aporte de la herramienta de portabilización plantea una mejora real en su experiencia de navegación.

Se encuestaron en total a 34 personas, cada una de las cuales realizó 2 encuestas. Las encuestas constaban de 2 tareas que debían realizarse utilizando la aplicación, por lo cual cada encuestado realizó 4 tareas en total. Con el propósito de no introducir un sesgo en el experimento, las actividades y las aplicaciones fueron intercambiadas, esto se detalla más adelante.

Datos Demográficos

En el siguiente gráfico (figura 7.1) podemos apreciar los datos demográficos de los encuestados:

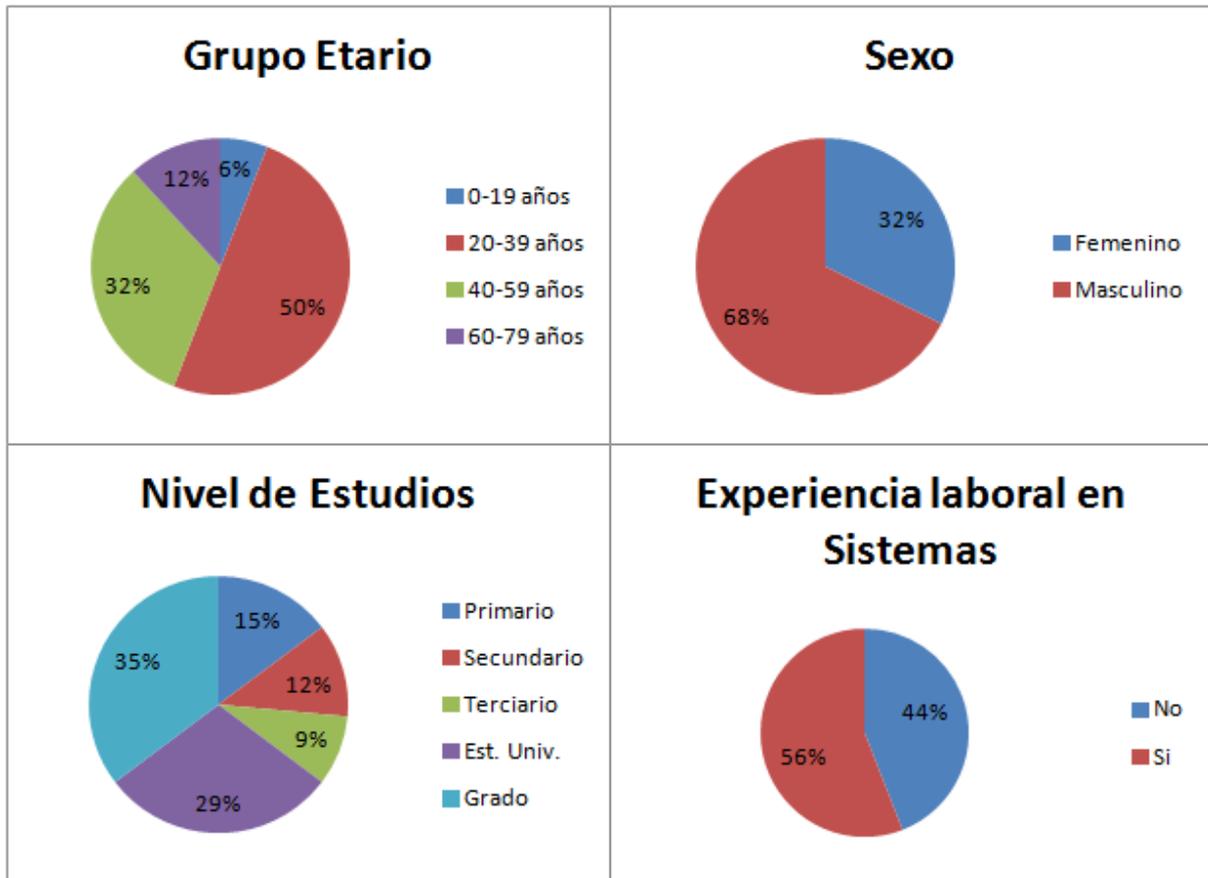


Figura 7.1 - Características de los participantes del experimento.

Podemos destacar que la mayor cantidad de participantes reportaron edades de entre 20 y 60 años que representó un 82% de los encuestados y sólo una minoría pertenecía al resto de las edades. Dos tercios de la población fueron hombres. Es importante destacar que la mayoría de las personas, un 85%, tenían al menos educación secundaria y un tercio eran personas calificadas con nivel terciario, estudiantes universitarios o graduados. Cabe mencionar que solo la mitad de ellos estaban relacionados con el área de sistemas de información o un trabajo a fin a la tecnología de la información.

7.2.3. Materiales del Experimento

En el experimento se utilizaron enunciados, cuestionarios, dispositivo móvil con sistema operativo Android, el administrador de scripts Tampermonkey, el navegador Web Dolphin y la herramienta de portabilización.

Encuesta:

El experimento consistió en diferentes actividades sobre las aplicaciones, tales como búsquedas, selecciones de menú, lecturas, ingresos de textos y visualización de imágenes simulando una interacción habitual de un navegante Web. Con esto se busca analizar la experiencia del usuario con el fin de evaluar la utilidad de una herramienta de adaptación, sobre todo en lo concerniente a la experiencia de usuario.

La encuesta se compone de dos partes:

- I. **Enunciado de las tareas a realizar sobre la aplicación:** la lista de actividades que el usuario debía realizar.
- II. **Cuestionario:** en la primer parte se obtiene la información demográfica y luego se captura la percepción del usuario con respecto a la usabilidad de la página Web portabilizada con la herramienta. Vemos una captura del cuestionario en la figura 7.2.

Hardware: Dispositivo móvil con S.O. Android 6.0, Moto G (3ª generación) con una resolución de 1280x720 píxeles y CSS pixel ratio 2.

Software:

- Tampermonkey Android v4.2.5291,
- Dolphin Browser Android v11.5.11,
- script de portabilización, versión modificada especialmente para realizar las encuestas que incorpora las portabilizaciones de Redmine y Pidgin.

Descargable en:

<https://github.com/serranorm/UNLP/blob/master/Encuestas/ResponsivePageEncuesta.user.js>

Encuesta

- Edad:
- Sexo: M F
- Estudios: primario secundario terciario estudiante universitario grado
- Experiencia laboral o estudios en el área de sistemas: Sí No
- Sitio Web:

<p>Tarea 1 -</p> <ul style="list-style-type: none"> • Realizada con herramienta <input type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p> <p>1-Qué problema encontró?</p> <p>a-Tamaño de fuente o img inadecuado <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>b-Menú poco accesible <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>c-Exceso de contenido en la página <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>Otro _____</p> <p>2-Satisfacción del 1 al 5: ____</p> <p>A ser completado por el Encuestador:</p> <p>3-Pudo finalizarla? <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>4-Cuánto tiempo le llevó? ____ seg.</p> <p>5-Cuántas interacciones realizó?</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 25%;">Zoom</th> <th style="width: 25%;">scroll</th> <th style="width: 25%;">clicks</th> <th style="width: 25%;">Rotaciones</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p>6-Errores registrados con el uso de Herramienta (QA):</p> <p>Tuvo que recargar la página? <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>No tomo el script <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>Otro _____</p>	Zoom	scroll	clicks	Rotaciones					<p>Tarea 2 -</p> <ul style="list-style-type: none"> • Realizada con herramienta <input type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p> <p>1-Qué problema encontró?</p> <p>a-Tamaño de fuente o img inadecuado <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>b-Menú poco accesible <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>c-Exceso de contenido en la página <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>Otro _____</p> <p>2-Satisfacción del 1 al 5: ____</p> <p>A ser completado por el Encuestador:</p> <p>3-Pudo finalizarla? <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>4-Cuánto tiempo le llevó? ____ seg.</p> <p>5-Cuántas interacciones realizó?</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 25%;">Zoom</th> <th style="width: 25%;">scroll</th> <th style="width: 25%;">clicks</th> <th style="width: 25%;">Rotaciones</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p>6-Errores registrados con el uso de Herramienta (QA):</p> <p>Tuvo que recargar la página? <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>No tomo el script <input type="checkbox"/> Sí <input type="checkbox"/> No</p> <p>Otro _____</p>	Zoom	scroll	clicks	Rotaciones				
Zoom	scroll	clicks	Rotaciones														
Zoom	scroll	clicks	Rotaciones														

Figura 7.2 - Cuestionario del experimento.

7.2.4. Tareas

A fin de realizar las tareas, el encuestado recibe una breve introducción del experimento. Luego, ya con el dispositivo listo para trabajar, se leen paso a paso las tareas a realizar. Al finalizar las tareas se lo entrevista, volcando la información al cuestionario de la figura 7.2. Parte de los datos no se consultan al encuestado, como por ej. cantidad de clicks, scrolls, tiempo total transcurrido, etc, para ello se utiliza un programa para grabar las acciones del entrevistado, más la observación y toma de notas por parte del encuestador.

Las actividades que podía realizar cada persona son las siguientes:

En Redmine

1. Tarea 1 (RT1)
 - A. Desde la home page localizar la imagen del libro “redmine cookbook”.
 - B. Informar al encuestador una vez encontrado.

2. Tarea 2 (RT2)
 - A. Desde la home page, seleccionar la opción peticiones, ubicado en el menú principal.
 - B. Ubicar la quinta petición y hacer click en esta.
 - C. En el detalle de la petición ver qué prioridad tiene e informarlo al encuestador.

En Pidgin

1. Tarea 1 (PT1)
 - A. Desde la home page, seleccionar la opción “About”, ubicada en el menú principal.
 - B. Bajo el título “Supported Languages” se encuentran los lenguajes soportados por la aplicación Pidgin. Informar al encuestador si se menciona el idioma “Italian”.

2. Tarea 2 (PT2)
 - A. Desde la home page, seleccionar la opción “Development”, ubicado en el menú principal.
 - B. Aparecerá un nuevo menú, que entre sus opciones tendrá la opción “register”. Seleccionar esta opción “register”.
 - C. Ya en la página "Register an account" completar el dato “username” para finalizar la tarea.

7.2.5. Hipótesis y Métricas

A partir del objetivo definido en la subsección 7.1.1 se trabajó en diferentes preguntas que permitieron conducir la investigación. Para cada Pregunta de investigación (PI) se definió una Hipótesis nula (H0), una Hipótesis Alternativa (HA) y una métrica para evaluar los resultados.

*PI₁: ¿Permite la herramienta tener una mejor **productividad** frente a no utilizarla?*

H0: No hay diferencia entre la productividad de ambas técnicas.

HA: La transformación mejora la productividad del usuario, frente a su no uso.

Métrica: tiempo en realizar las tareas. Donde a menor tiempo mayor productividad. La métrica se obtiene a partir de controlar el tiempo que demoró el entrevistado en realizar la tarea.

*PI₂: ¿La **eficiencia** se ve incrementada con la portabilización?*

H0: no hay diferencia en la cantidad de interacciones realizadas para concluir las tareas utilizando ambas técnicas.

HA: La transformación reduce la cantidad de interacciones realizadas.

Métrica: Cantidad de interacciones. Donde a menor cantidad de interacciones, mayor eficiencia. La métrica se obtiene a partir de observar las interacciones realizadas por el encuestado.

*PI₃: ¿Mejora la **satisfacción** del usuario al utilizar la herramienta?*

H0: no hay diferencia en cuanto a la satisfacción del usuario al realizar las tareas utilizando ambas técnicas.

HA: La transformación eleva el nivel de satisfacción del usuario.

Métrica: Valoración del usuario tras realizar las tareas. Donde mayor sea el número de la valoración (1 al 5), mayor es la satisfacción. La métrica se obtiene a partir de la respuesta 2 del cuestionario.

*PI₄: ¿Cuál técnica reporta más **problemas de usabilidad**?*

H0: Ambas técnicas presentan los mismos problemas

HA: La técnica de portabilización reporta menos incidentes de usabilidad que la aplicación estándar.

Métrica: Cantidad de quejas sobre el tamaño de fuente o imagen, exceso de contenido y manejo del menú. Donde mayor sea el número de problemas reportados, mayor son los problemas de usabilidad. La métrica en este caso se obtiene a partir de lo recabado en el cuestionario sobre tres preguntas simples por sí o por no.

7.2.6. Diseño del Experimento

Para lograr un mejor muestreo de la experiencia del usuario, fueron seleccionados dos sitios Web con diseño no adaptativo y se plantearon actividades a realizar sobre cada sitio para evaluar cómo se desenvuelve la persona al usar la herramienta y al no usarla. A fin de evitar el sesgo fruto de la experiencia adquirida en el sitio, a cada usuario se le presentaron las actividades

de un sitio haciendo uso de la herramienta de portabilización, mientras que se le pedía hacer las actividades sobre el otro sitio sin el uso de la herramienta. Los participantes se eligieron de forma aleatoria.

Las actividades a realizar fueron tareas puntuales sobre las opciones propias de cada sitio, a fin de tener un resultado homogéneo en la muestra. Estas tareas incluyeron la búsqueda de enlaces, el acceso a opciones del menú, lectura de textos e imágenes y la introducción de texto mediante la carga de formularios. Los sitios Web elegidos para realizar las tareas de las encuestas fueron Redmine y Pidgin en sus versiones originales y portabilizadas mediante el uso de la herramienta.

Cada persona tomaba dos configuraciones cruzadas (ver tabla 7.3):

- Persona 1: Configuraciones 3,2
- Persona 2: Configuraciones 4,1

las dos personas siguientes del experimento tendrán una distribución igual de configuraciones y así sucesivamente.

	Con Herramienta	Sin Herramienta
Pidgin	1	2
Redmine	3	4

Tabla 7.3 - Configuraciones de encuestas.

A modo ilustrativo, al primer encuestado se le pidió realizar las siguientes cuatro tareas:

Configuración	Experimento
3	Redmine con uso de herramienta. Tareas 1 y 2. Ver <i>Figura 7.5</i>
2	Pidgin sin uso de herramienta. Tareas 1 y 2. Ver <i>Figura 7.6</i>

Tabla 7.4 - Configuraciones para el primer encuestado.

• Sitio Web: **Redmine** (<http://www.redmine.org>)

<p>Tarea 1 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p>	<p>Tarea 2 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p>
---	---

Figura 7.5 - Vista de encuesta con combinatoria 1 (configuración 3).

• Sitio Web: **Pidgin** (<https://pidgin.im>)

<p>Tarea 1 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input type="checkbox"/> Sí <input checked="" type="checkbox"/> No <p>Preguntar al Participante:</p>	<p>Tarea 2 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input type="checkbox"/> Sí <input checked="" type="checkbox"/> No <p>Preguntar al Participante:</p>
---	---

Figura 7.6 - Vista de encuesta con combinatoria 2 (configuración 2).

Al segundo encuestado se le pidió realizar las siguientes cuatro tareas:

Configuración	Experimento
4	Redmine sin uso de herramienta. Tareas 1 y 2. Ver <i>Figura 7.8</i>
1	Pidgin con uso de herramienta. Tareas 1 y 2. Ver <i>Figura 7.9</i>

Tabla 7.7 - Configuraciones para el segundo encuestado.

• Sitio Web: **Redmine** (<http://www.redmine.org>)

<p>Tarea 1 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input type="checkbox"/> Sí <input checked="" type="checkbox"/> No <p>Preguntar al Participante:</p>	<p>Tarea 2 -</p> <ul style="list-style-type: none"> Realizada con herramienta <input type="checkbox"/> Sí <input checked="" type="checkbox"/> No <p>Preguntar al Participante:</p>
---	---

Figura 7.8 - Vista de encuesta con combinatoria 3 (configuración 4).

<ul style="list-style-type: none"> • Sitio Web: Pidgin (https://pidgin.im) 	
<p>Tarea 1 -</p> <ul style="list-style-type: none"> • Realizada con herramienta <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p>	<p>Tarea 2 -</p> <ul style="list-style-type: none"> • Realizada con herramienta <input checked="" type="checkbox"/> Sí <input type="checkbox"/> No <p>Preguntar al Participante:</p>

Figura 7.9 - Vista de encuesta con combinatoria 4 (configuración 1).

Y así sucesivamente con el resto de los encuestados.

Como podemos ver la encuesta es siempre la misma, sólo que se parametriza de 4 formas distintas.

7.2.7. Procedimiento

A las personas encuestadas no se les dio ningún tipo de entrenamiento, el escenario era casual y se les requería prestar atención, evitando el uso televisores o cualquier otro dispositivo que pueda distraer la atención. Luego, se les explicaba brevemente en qué consistía el experimento, se les informaba las tareas a realizar y se les entregaba el dispositivo móvil a utilizar. Previo a la entrega del mismo se procedía a verificar la conexión WIFI, se iniciaba desde cero el navegador borrando la caché y se activaba la aplicación de grabación. La persona realizaba el experimento y luego se completaba el cuestionario definido en la subsección 7.2.3. A través del cuestionario, se recolectó información para evaluar la herramienta, obtener datos demográficos y conocer el nivel de satisfacción del usuario.

7.2.8. Procedimiento de Análisis

Las muestras se procesaron utilizando la prueba U de Mann-Whitney [15], que es una técnica de prueba de hipótesis estadística no paramétrica, considerando un nivel de confianza estándar (α) de 0,05. Esta técnica se puede usar con un tamaño de muestra desigual, que es el caso de este experimento. Además, calculamos el tamaño del efecto usando la técnica Cliff's Delta. Para lograr la tarea de procesar los resultados recopilados, primero procesamos y digitalizamos las respuestas. Luego, utilizamos diferentes scripts basados en el lenguaje Python (versión 3.6.1) y la *biblioteca SciPy* (versión 0.19.0) para calcular pruebas de hipótesis no paramétricas.

7.3. Resultados

En esta sección presentaremos los resultados obtenidos durante el experimento. Estos datos y su análisis son motivados por las preguntas de investigación definidas en la subsección 7.2.5. Las muestras analizadas

permitieron medir la eficiencia, la productividad, la satisfacción del usuario y los problemas de usabilidad.

Eficiencia

A partir de los resultados que muestra la figura 7.10, se puede concluir que el zoom es la interacción que presenta una mayor diferencia en favor del uso de la herramienta, ya que prácticamente no es necesario una vez hecha la portabilización.

La cantidad de scrolls también se redujo con el uso de la herramienta, principalmente porque una vez que se hace zoom la persona debe hacer una mayor cantidad de desplazamientos para encontrar los elementos en la página, sobre todo hay muchos scrolls laterales (de izquierda a derecha o de derecha a izquierda) que no se dan con el uso de la herramienta.

En cuanto a los clicks se ve un pequeño aumento en el caso del uso de la herramienta ya que para hacer uso del menú adaptativo, se necesita hacer click en él para abrirlo.

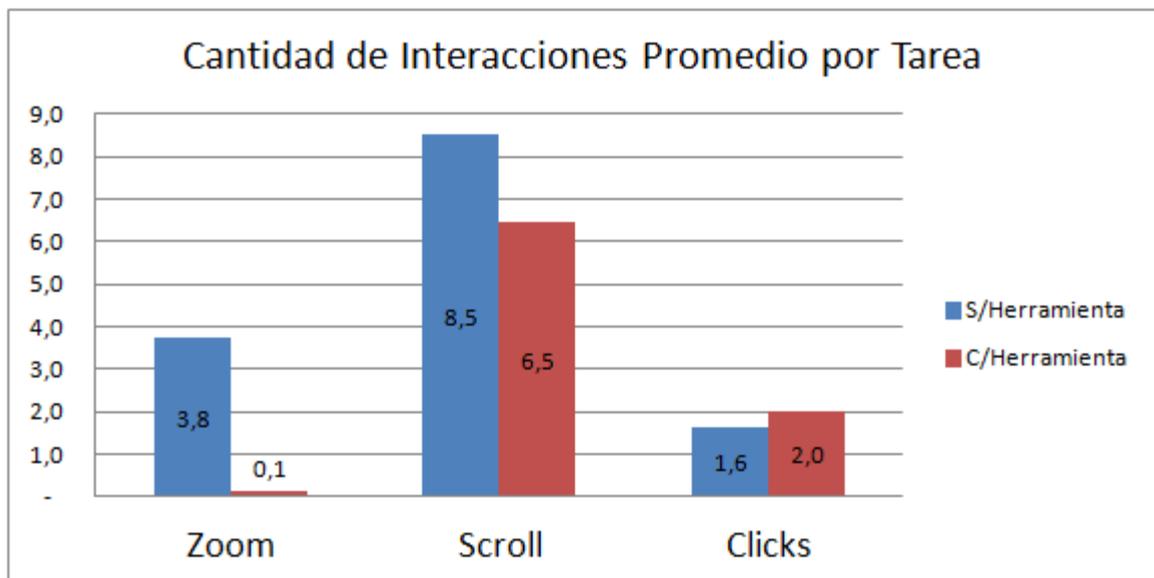


Figura 7.10 - Cantidad de Interacciones Promedio por Tarea.

Tiempos

Los tiempos promedios para la realización de las tareas que muestran la figura 7.11 han sido bastante parejos, obteniendo una mejora en 6,5 segundos promedio a favor del uso de la herramienta de portabilización.

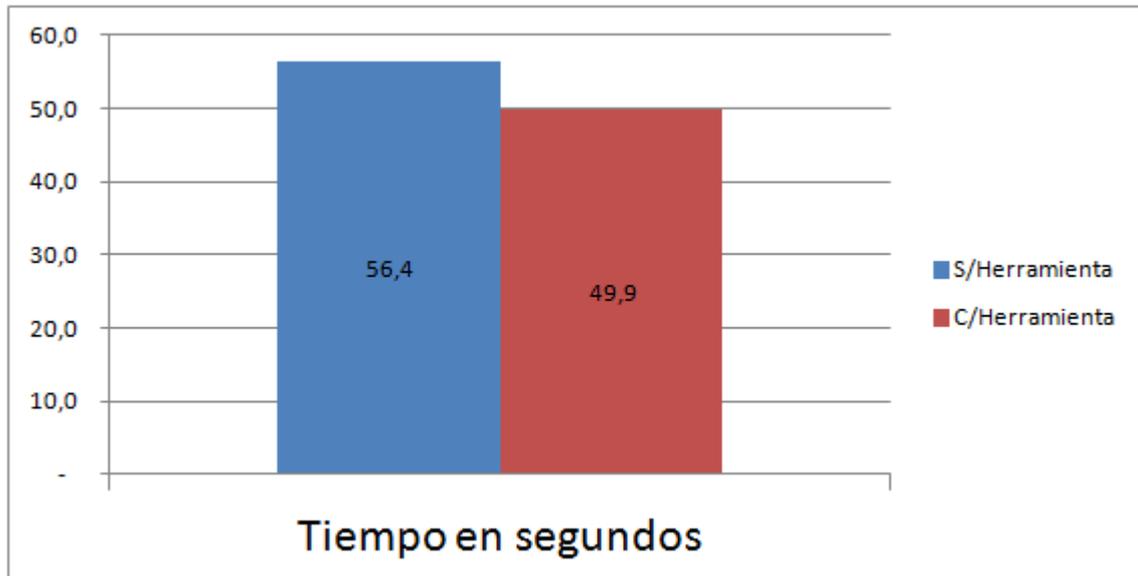


Figura 7.11 - Tiempos promedios con y sin uso de la herramienta.

Satisfacción del usuario

Se consultó a los encuestados a fin de obtener la satisfacción del usuario con respecto al uso del sitio Web. Como ayuda a los encuestados se les brindó la siguiente guía:

- 1 - No me ha gustado nada
- 2 - No me ha gustado
- 3 - Está bien
- 4 - Me ha gustado
- 5 - Me ha encantado

La navegación con la herramienta de portabilización obtuvo una satisfacción de 4,2 en promedio mientras que sin el uso de la herramienta dieron un resultado de 3,1. La figura 7.12 refleja la satisfacción promedio de ambos sitios.

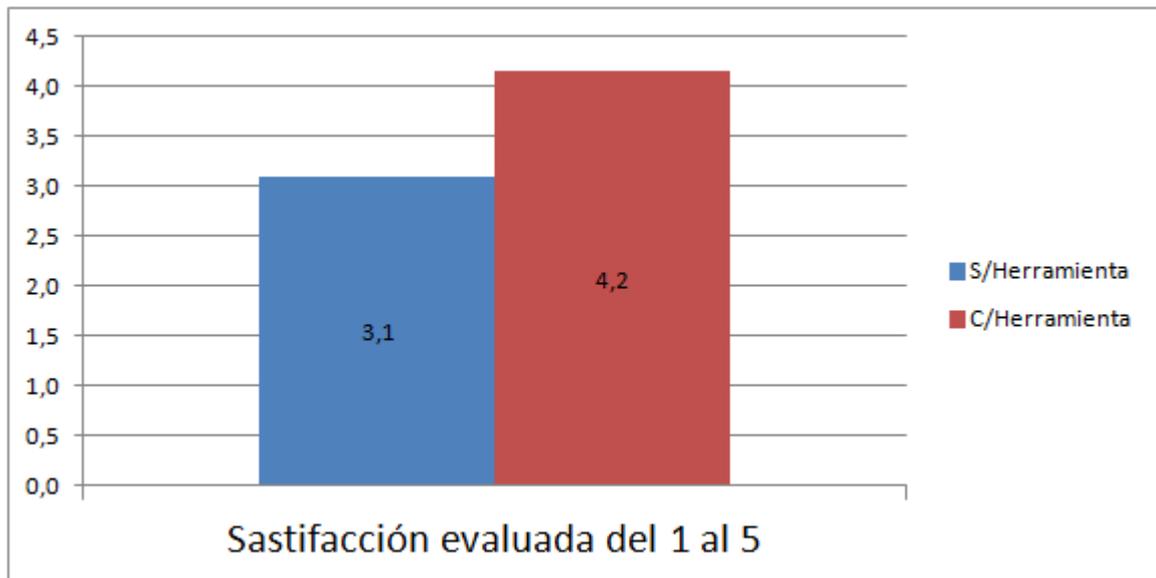


Figura 7.12 - Satisfacción del usuario evaluada del 1 al 5.

Problemas de usabilidad

Los problemas evaluados fueron de navegación y legibilidad. Los mayores inconvenientes estuvieron dados por el tamaño de las fuentes o imágenes, el uso de los menús y el exceso de información presentada en la pantalla, evidenciándose la necesidad del uso de interacciones extras como zooms, scrolls, y clicks para poder llevar a cabo las tareas, en las pruebas que no hicieron uso de la herramienta de portabilización.

Estos datos se obtienen de las preguntas 1a, 1b y 1c del cuestionario del experimento, realizadas a los encuestados.

a) Problemas con el Tamaño de fuente o imagen inadecuado

La figura 7.13 indica que los participantes encontraron problemas con el tamaño de fuente inadecuado o imágenes demasiado pequeñas en un 41% sin el uso de la herramienta y el mismo problema se redujo al 12% con el uso de la herramienta de portabilización.

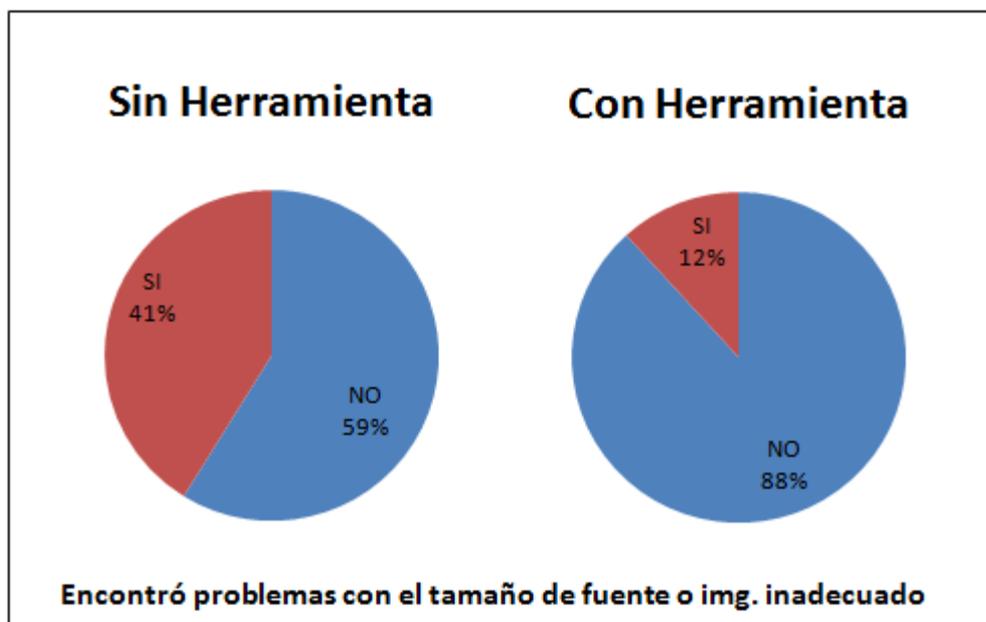


Figura 7.13 - Problemas de tamaño de fuentes o imágenes.

b) Problemas con el Uso del Menú

En el caso del uso del menú, la figura 7.14 muestra que un 32% de los participantes encontraron problemas para localizarlo y/o utilizarlo sin el uso de la herramienta y el problema se presentó en un 7% con el uso de la herramienta.

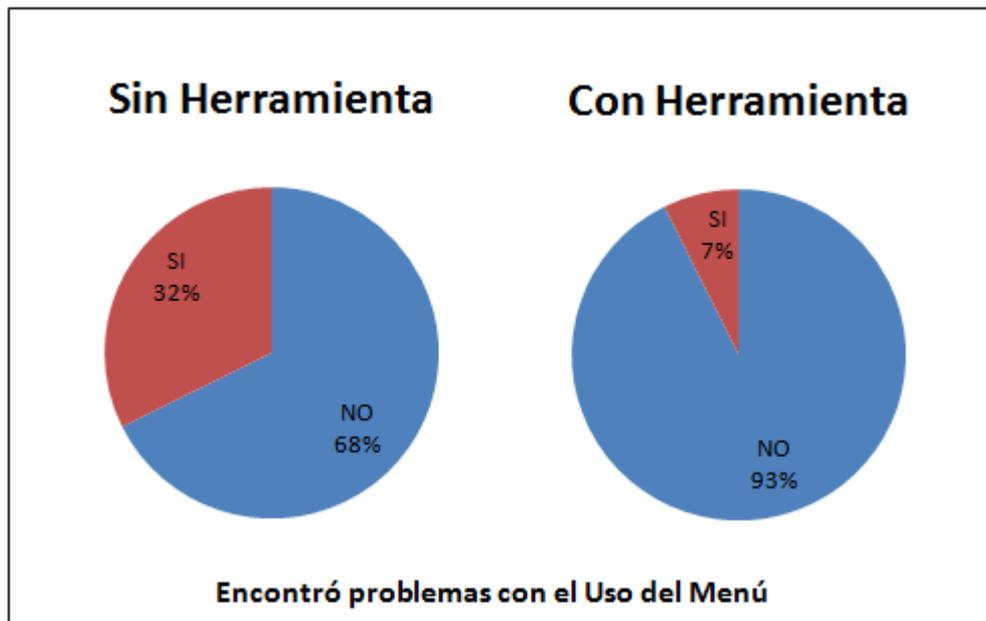


Figura 7.14 - Problemas con el uso del menú.

c) Exceso de Contenido en la Página

Los problemas de exceso de contenido en los dispositivos de tamaño reducido pueden ser mitigados mediante la selección de las partes que se quieren mostrar, esta tarea es sencilla para el diseñador, que con el Diseñador de Adaptaciones puede seleccionar solo las partes que considera relevantes y sencillamente al no seleccionar las secciones que no se quieren mostrar, serán omitidas automáticamente por la herramienta.

La figura 7.15 muestra los porcentajes totales de respuestas positivas y negativas a las preguntas relacionadas a los problemas por el exceso de contenido en la página. Sobre el total de las encuestas, un 34% de los encuestados expresó problemas sin el uso de la herramienta y se redujo al 19% con el uso de la herramienta.

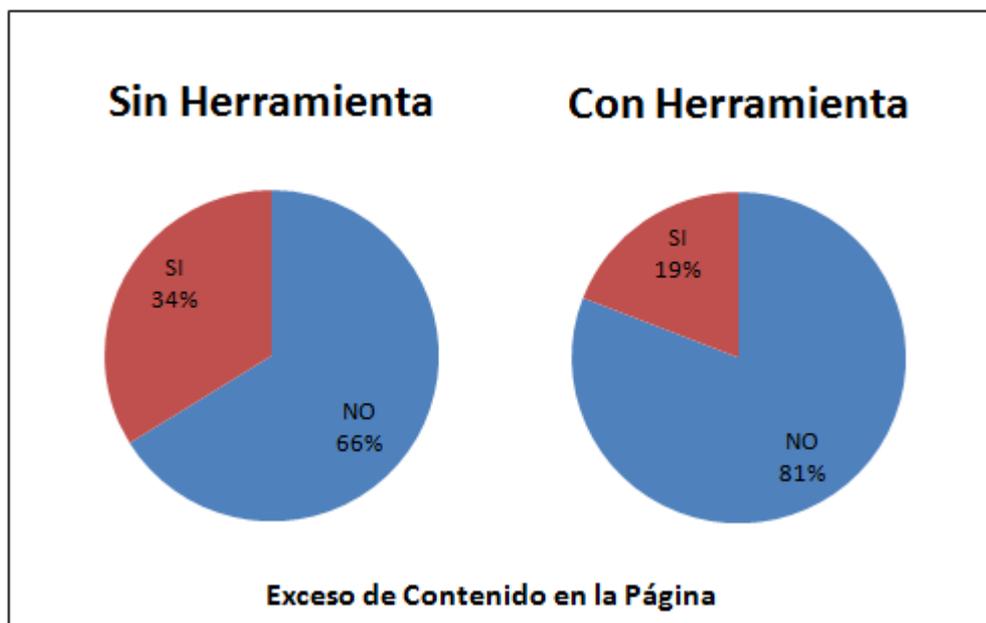


Figura 7.15 - Problemas de exceso de contenido en la página.

7.4. Análisis e Implicancias

Analizaremos el resultado de las métricas introducidas en la subsección 7.2.5 que se muestran en la Tablas 7.16, 7.17, 7.18 y 7.19. Para cada una de estas métricas se informa el valor para el tamaño del efecto (effect size) y el *p*-valor (p-value) resultante de la prueba de hipótesis que compara los resultados de ambos enfoques. Para cada una de las Preguntas de Investigación se expondrá su tabla correspondiente, haciendo una interpretación de los resultados. En los casos que el *p*-valor es menor que el nivel alfa 0.05 nos permite rechazar la hipótesis nula y apoyar la hipótesis alternativa.

PI₁: ¿Permite la herramienta tener una mejor productividad frente a no utilizarla?

Los tiempos promedio para realizar una tarea pasaron de 56,4 segundos a 49,9 segundos con el uso de la herramienta, reduciendo un 12% el tiempo mediante su uso.

Tarea	Tiempo	
	Effect size	p-value
RT1	-0.778	0.024
RT2	0.151	0.640
PT1	-0.195	0.172
PT2	-1.075	0.003

Tabla 7.16

En la métrica de finalización del tiempo, la mitad de las tareas muestran una mejora en el tiempo requerido para realizar la tarea. La tarea PT1 muestra alguna mejoría pero no una mejora estadísticamente significativa ($p = 0.172$) y RT2 no muestra preferencia ($p = 0.640$). Un estudio a posteriori señala que la introducción de un menú "hamburguesa" requería un evento de click adicional para abrirlo y consumía tiempo para animar la expansión del menú.

PI₂: ¿La eficiencia se ve incrementada con la portabilización?

La tasa de eventos se dividió en tres submétricas: Zoom, Desplazamiento (scroll) y Clicks. En las interacciones, el zoom presenta la mayor mejora con un 97% con el uso de la herramienta, los scrolls se redujeron en un 23% con su uso y los clicks vemos que, al contrario, aumentaron en un 25% mediante su uso.

Tarea	Zoom		Scroll		Clicks	
	Effect size	p-value	Effect size	p-value	Effect size	p-value
RT1	-2.805	0.000	-1.264	0.001	-0.343	0.173
RT2	-2.358	0.000	-0.928	0.002	1.086	1.000
PT1	-3.684	0.000	0.919	0.986	-	1.000
PT2	-1.879	0.000	-2.450	0.000	-0.343	0.173

Tabla 7.17

Los eventos de UI de zoom presentan una mejora significativa para todas las tareas con un *p-valor* inferior a 0.05 en todos los casos, debido básicamente a que el usuario ya no necesita hacer zoom una vez adaptada la interfaz.

Los eventos de desplazamiento informaron un mejor rendimiento al usar UIs aumentadas, pero la tarea PT1 no muestra ninguna diferencia ($p = 0.986$). Un estudio post hoc mostró que los usuarios finales comienzan a usar eventos de desplazamiento en lugar del acercamiento/alejamiento porque el contenido se adaptó al ancho de la ventana gráfica en una página más larga, la cual debían desplazar para verificar si el idioma italiano era compatible o no.

Los eventos click muestran algunas mejoras, pero no estadísticamente significativas, en RT1 ($p = 1.73$) y PT2 ($p = 1.73$). Ninguna diferencia en RT2 ($p = 1$) ni en PT1 ($p = 1$). Después de estudiar las muestras, dedujimos que las transformaciones de UI no introdujeron una refactorización que reduzca la cantidad de clicks requeridos para llevar a cabo la tarea.

PI₃: ¿Mejora la satisfacción del usuario al utilizar la herramienta?

La satisfacción de los usuarios pasa de 3,1 a 4,2 puntos en una escala de 1 al 5. Lo que representa una mejora del 35% con el uso de la herramienta.

Tarea	Satisfacción	
	Effect size	p-value
RT1	1.520	0.000
RT2	1.688	0.000
PT1	1.641	0.000
PT2	1.735	0.000

Tabla 7.18

La métrica de satisfacción sobresale en el rendimiento en todas las tareas. En todos los casos, el *p-valor* es menor que el nivel alfa 0.05, lo que significa un mejor rendimiento de la métrica.

PI₄: ¿Cuál técnica reporta más problemas de usabilidad?

De los resultados obtenidos, se concluye la mejora sustancial que aporta la herramienta de adaptación, ya que los tres problemas evaluados tuvieron una sensible tasa de disminución. En el caso de Pidgin sin herramienta en total se encontraron 38 problemas y con herramienta 12, lo que significa una reducción del 68%. Para el caso de Redmine se encontraron 35 problemas en total sin el uso de la herramienta y 14 mediante el uso de la herramienta, en este caso los problemas se mitigaron en un 60%.

Tarea	Problemas por tamaño de fuente inadecuado		Problemas con el menú	
	Effect size	p-value	Effect size	p-value
RT1	-0.686	0.030	-	99
RT2	-0.519	0.072	-1.098	0.003
PT1	-0.951	0.006	-0.142	0.347
PT2	-0.630	0.040	-1.294	0.001

Tabla 7.19

Los problemas de usabilidad tienen una reducción importante mediante el uso de la UI aumentada. Para el caso de “Problemas por

tamaño de fuente inadecuado” hubo una gran mejora, teniendo 3 de las 4 tareas el *p-valor* menor al nivel alfa 0.05. Salvo en el caso de “RT2 - Problemas por tamaño de fuente inadecuado” en donde existe mejora pero no alcanza estadísticamente para ser concluyente ($p = 0.072$).

En “Problemas con el menú” también se observan mejoras, cabe aclarar que 1 de las 4 tareas no era medible, esta es la tarea RT1, la cual no incluía el uso del menú ($p = .99$). De las 3 tareas que se pudieron medir, dos dieron excelentes resultados con *p-valor* mucho menor al nivel alfa 0.05. Ya en la tarea “PT1 - Problemas con el menú” los resultados fueron parejos, no arrojando mejora aparente con la herramienta, ocurre lo contrario con “PT2 - Problemas con el menú”, se obtuvieron resultados muy alentadores ya que al ser reestructurado un submenú que era complejo de entender, haciéndolo más legible desde un dispositivo móvil, dió un *p-valor* de 0.001 permitiendo rechazar la hipótesis nula y apoyar la hipótesis alternativa.

Resumiendo, los participantes al utilizar las UIs aumentadas fueron más eficientes en el tiempo y más precisos en la ejecución de tareas que aquellos que trabajaron con el original. Afirmamos que estos resultados son evidencia preliminar de los beneficios del enfoque de Augmentation.

8. Conclusiones y Trabajo Futuro

8.1. Conclusiones

Las aplicaciones Web 1.0 fueron diseñadas para ser accedidas por computadoras de escritorio, pero si la navegación se realiza con dispositivos móviles podemos encontrar problemas que perjudican la experiencia del usuario en el sitio. En este trabajo se presentó una propuesta que portabiliza este tipo de aplicaciones, transformándolas en aplicaciones Web adaptativas que puedan ser correctamente visualizadas en dispositivos móviles, con un costo prácticamente nulo.

Con el objetivo de implementar una herramienta que dirija el proceso de portabilización y permita especificar las adaptaciones, se investigó sobre tecnologías Web actuales y patrones de diseño adaptativo. El enfoque estuvo centrado en los siguientes puntos:

1. Local Storage para el almacenamiento y manejo de las adaptaciones.
2. jQuery, Bootstrap, JavaScript y CCS para convertir las viejas interfaces en adaptativas. Entre estas tecnologías cabe destacar el uso del framework Bootstrap, que aportó dinamismo a la herramienta y facilidad de mantenimiento.
3. TamperMonkey/Greasemonkey se empleó para diseñar y utilizar la herramienta.
4. Patrones de diseño Web adaptativo, su utilización fue un recurso enriquecedor para la herramienta por la potencia y facilidad de uso que brinda. Los patrones permiten que quien está haciendo el diseño entienda rápidamente qué comportamiento tendrá.

La herramienta implementada en este trabajo presenta las siguientes ventajas:

1. Podemos lograr que aplicaciones Web muy complejas de acceder mediante dispositivos móviles pasen a ser adaptativas de manera relativamente sencilla.
2. El costo de portabilizar es muy bajo en comparación al costo de rehacer un sitio Web completo.
3. Permite reorganizar el contenido a voluntad del diseñador, dando la posibilidad de quitar información que el *analista UX* o el diseñador considere no relevante y “resaltar lo más importante”, algo que no se puede lograr con herramientas automáticas.
4. Las adaptaciones se realizan en el cliente (client-side) y sin necesidad de realizar modificaciones de contenido o diseño en el servidor de aplicaciones.
5. Las tecnologías usadas son todas de licencia de software libre, por lo que no existen costos asociados a licencias.

6. Podrían crearse repositorios con scripts de portabilización para ser reutilizados, modificados y compartidos por distintos usuarios adaptándolos a usos particulares o simplemente a los gustos del usuario.
7. Las mejoras logradas en el Diseñador de Adaptaciones agilizan la tarea del diseñador. Tanto las opciones de configuración como el agregado del previsualizador dan una gran facilidad de uso a la herramienta, con lo que se acortan los tiempos empleados por el diseñador.

Como casos de uso con la herramienta se trabajó en dos sitios Web, Redmine y Pidgin. Sobre estos sitios se realizaron experimentos controlados y encuestas a fin de evaluar la respuesta de los usuarios a las interfaces portabilizadas y sin portabilizar, obteniendo como resultado evidencia, que luego de su análisis, nos permiten concluir que el uso de la de la herramienta mejora la experiencia del usuario, lo cual vimos reflejado tanto en el análisis de las encuestas como en los comentarios de los encuestados.

8.2. Trabajo Futuro

Algunas mejoras y ampliaciones a los trabajos realizados podrían ser las siguientes:

1. Incorporar nuevas opciones de templates adaptativos. Pueden ser para dispositivos en específico, nuevas tendencias de diseño, o de manera genérica pensando en pantallas muy pequeñas o muy grandes. También pueden incluirse diseños específicos pensados en brindar accesibilidad a personas con capacidades diferentes.
2. Agregar más patrones de diseño adaptativo a la herramienta para dar mayores opciones al diseñador, separándolos por categorías a fin de facilitar su selección.
3. Personalización de la herramienta a gusto del diseñador, por ej. agregar colores para la selección de elementos, permitir cambiar la ubicación de las ventanas y las fuentes usadas.
4. Permitir que el previsualizador cambie de dimensiones de acuerdo al dispositivo móvil seleccionado, su resolución y pixel ratio. Esta configuración de dispositivo simulado podría venir precargada y permitir agregar otras nuevas. Otras opciones interesantes serían mover el previsualizador, poder girarlo y ocultarlo.
5. Presentar un teclado virtual para tener una mejor referencia del espacio disponible que tendrá el usuario al cargar los datos en un formulario.
6. Investigar cómo implementar la conservación de los estilos CSS embebidos y externos a nivel CSS al portabilizar un sitio Web, mantener los colores de fondo, estilos de fuentes y estructura del sitio. Una estrategia para obtener el color, podría ser tomar píxeles del fondo de pantalla y promediarlos.

7. Subir las portabilizaciones de sitios Web generadas a un servicio de almacenamiento general, luego servicios tipo REST podrían bajar las portabilizaciones, recuperándolas por ej. por id.
8. Generar nuevos casos de estudio para obtener métricas más generales, portabilizando más sitios y evaluándolos con las mismas métricas, obteniendo estadísticas más concluyentes, a partir de una mayor cantidad de información.
9. Implementar estrategias de mejoras pasivas ante problemas de conectividad en entornos móviles. Por ejemplo, el uso de una caché client-side que permita al conectarse almacenar algunas páginas de manera offline, a fin de poder seguir navegando una vez que se perdió la conexión de datos.
10. Configurar un proxy Web antes del servidor de la aplicación Web a transformar, instalando en este proxy el script de portabilización para devolver la aplicación Web ya transformada en adaptativa directamente al usuario del dispositivo móvil, evitando que el mismo deba bajar e instalar el script por medio de un administrador de scripts como Tampermonkey.
11. El Diseñador de Adaptaciones podría estar disponible para ser descargado por usuarios finales, y no solo para los diseñadores, en una plataforma pública y amigable como UserStyles²³. Los mismos podrían usarlo para generar sus propios scripts, personalizando las aplicaciones Web a su gusto o necesidad. Para esto la herramienta debería tener un mayor grado de madurez y facilidad de uso.

²³ <https://userstyles.org/> - Visitado el 31/10/2017

Glosario

Analista UX: o Analista de Usabilidad, es responsable de realizar el análisis de usabilidad enfocado en usuarios finales.

Biblioteca SciPy: es una colección de herramientas numéricas para Python que se distribuye libremente. Brinda funciones de cálculo numérico de gran capacidad, optimización de funciones, integración, funciones especiales, ecuaciones diferenciales, entre otras.

Diseño Web Adaptativo (Responsive Web Design, RWD): Permite que una página Web pueda verse bien, y sea fácil de usar, independientemente de si el dispositivo es una computadora de escritorio, tableta o teléfono. Utiliza CSS y HTML para cambiar su contenido, modificando el tamaño, ocultando, reduciendo, ampliando o moviendo el mismo de modo que la página se vea bien en cualquier pantalla.

GQM: del inglés Goal Question Metric, define mediciones del proceso y de los resultados de un proyecto. Se basa en la idea de que un programa de medición puede ser más satisfactorio si es diseñado teniendo en cuenta las metas (Goal). Las preguntas deberán ser medibles para poder evaluar si se está alcanzando la meta. Define un objetivo, clarifica el objetivo con preguntas y define métricas que darán información para responder a estas preguntas. Se puede aplicar a todo el ciclo de vida del producto, procesos y recursos. Además se puede alinear fácilmente con el ambiente organizacional.

El método fue definido originalmente por Basili y Weiss (1984) y extendido posteriormente por Rombach (1990) como resultado de muchos años de experiencia práctica e investigación académica. Tiene como principio básico que la medición debe ser realizada, siempre, orientada a un objetivo.

Local Storage: es una característica de HTML5 que permite almacenar datos hasta que sean eliminados explícitamente por la aplicación o el usuario. El almacenamiento local permite almacenar los datos por dominio y persistir más allá de la sesión actual, aunque el navegador se cierre o el dispositivo se reinicie.

Media queries: Es una técnica CSS introducida en CSS3. Utiliza la regla @media para incluir un bloque de propiedades CSS sólo si una determinada condición es verdadera.

Viewport: es el área visible de una página Web para el usuario. El viewport varía con el dispositivo y será menor en un teléfono móvil que en una pantalla de computadora.

HTML5 introdujo un método para permitir que los diseñadores de páginas Web tomen el control sobre el viewport, a través de la etiqueta <meta>.

Web Augmentation: es una técnica para la adaptación de aplicaciones de terceros, se basa en la alteración de su interfaz de usuario original, generalmente mediante el uso de scripts que se ejecutan en el lado del cliente (por ejemplo, el navegador). Su propósito no es crear una nueva aplicación, sino desarrollar un nuevo marco de experiencia web sobre un sitio existente.

Referencias bibliográficas

[1] Fielding, J. (2014). Beginning Responsive Web Design with HTML5 and CSS3. Apress. ISBN: 978-1430266945

[2] Archuby, F., Pierini, S., Takahashi, D., Firmenich, S., Rivero, J. M. & Urbietta, M. (2015). Métodos y herramientas para portabilizar aplicaciones Web Convencionales, Programa de Desarrollo de Aplicaciones e Innovación, Fac. Informática, UNLP.

[3] Cochran, D. (2012). Twitter Bootstrap Web Development How-To. David Cochran. Packt Publishing. ISBN: 978-1849518826

[4] Spurlock, J. (2013). Bootstrap: Responsive Web Development. O'Reilly Media. ISBN: 978-1449343910

[5] Shenoy, A. & Sossou, U. (2014). Learning Bootstrap. Packt Publishing. ISBN: 978-1782161844

[6] Van Solingen, R. & Berghout, E. (1999). The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. McGraw-Hill. ISBN: 978-0077095536

[7] Southehal, P. H. (2014). Implementing the Stakeholder based Goal-Question-Metric (GQM) Measurement Model for Software Projects. Trafford Publishing. ISBN: 978-1490740096

[8] Díaz, O. (2012). Understanding Web Augmentation. In Current Trends in Web Engineering (pp. 79-80). Springer Berlin Heidelberg.

[9] Firmenich, D., Firmenich, S., Rivero, J. M. & Antonelli, L. (2014). A Platform for Web Augmentation Requirements Specification. In Web Engineering (pp. 1-20). Springer International Publishing.

[10] Pierini, S., Takahashi, D., Fleitas, M., Serrano, R., Witkin, M., Sottile, C., Firmenich, S., Torres, D., Rivero, J. M. & Urbietta, M. (2016). Técnicas avanzadas de aumentación Web: portabilización y Web semántica, Programa de Desarrollo de Aplicaciones e Innovación, Fac. Informática, UNLP.

[11] Díaz, O. & Arellano, C. (2012). Sticklet: An end-user client-side augmentation-based mashup tool. In ICWE, volume 7387 of Lecture Notes in Computer Science, pages 465-468. Springer International Publishing.

[12] Sarkis, M., Concolato, C. & Dufourd, J.C. (2017). A multi-screen refactoring system for video-centric Web applications. New York: Springer Science+Business Media.

[13] Bibeault, B., Katz, Y. & De Rosa, A. (2015). JQuery in Action, Third Edition. Manning. ISBN: 978-1617292071

[14] Pilgrim, M. (2005). Greasemonkey Hacks: Tips & Tools for Remixing the Web with Firefox. O'Reilly Media. ISBN: 978-0596553579

[15] Ross, S.M. (2004) Introduction to probability and statistics for engineers and scientists. Academic Press, Cambridge, MA, USA.

[16] Shull, F., Singer, J. & Sjøberg, D.I.K. (2008). Guide to Advanced Empirical Software Engineering. Chapter 8, 3.7. Experiment Planning (pp 219-225). Springer London. ISBN: 978-1848000438

Anexo A: Guía de instalación de la herramienta de portabilización

La herramienta de portabilización está formada por dos componentes principales, el Diseñador (script WebAdaptation) y el Motor de Adaptaciones (script ResponsivePage). Cada componente tiene su propio conjunto de requerimientos y método de instalación, que se describen a continuación.

A.1. Instalación del Diseñador de Adaptaciones

El diseñador fue implementado para ser ejecutado en una computadora de escritorio o notebook. El tamaño de las pantallas de estos dispositivos permiten trabajar cómodamente sobre las páginas Web dado que facilitan la visualización y la selección de los elementos HTML.

Los requerimientos de software necesarios para ejecutar el diseñador son:

- A. Mozilla Firefox²⁴ versión 56.0.1,
- B. Greasemonkey²⁵ versión 3.17, complemento para Firefox,
- C. El componente *diseñador* de la herramienta disponible en:
<https://github.com/serranorm/UNLP/blob/master/scripts/WebAdaptation.user.js>

Los pasos a realizar para la instalación son los siguientes:

1. Descargar e instalar Mozilla Firefox (A) en la computadora.
2. Abrir el navegador Web Firefox, ingresar a la página del complemento Greasemonkey (B), seleccionar la versión 3.17, y presionar el botón "Agregar a Firefox (ver Figura A.1).

²⁴ <https://ftp.mozilla.org/pub/firefox/releases/56.0.1/> - Visitado el 15/07/2018

²⁵ <https://addons.mozilla.org/es/firefox/addon/greasemonkey/versions/> - Visitado el 15/07/2018

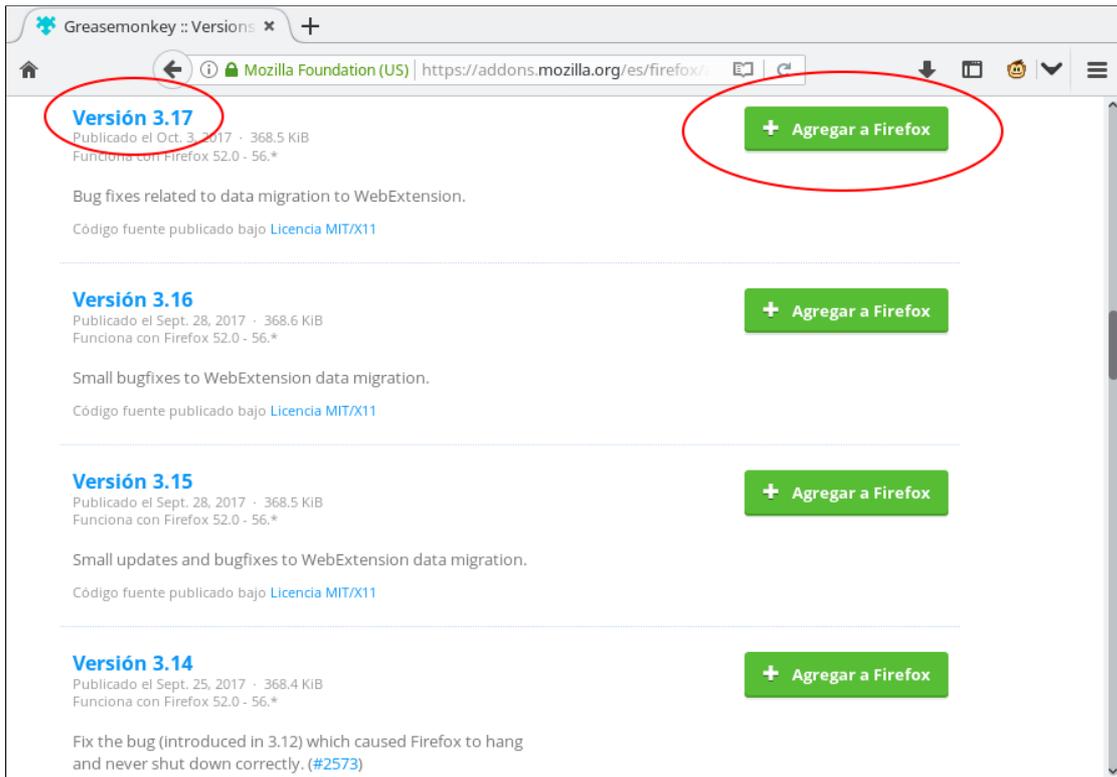


Figura A.1 - Instalación de Greasemonkey.

3. Ingresar a Greasemonkey, seleccionar la opción “Nuevo script” (ver figura A.2), completar los datos de la definición del script Web Adaptation (ver figura A.3) y presionar el botón “Aceptar”.

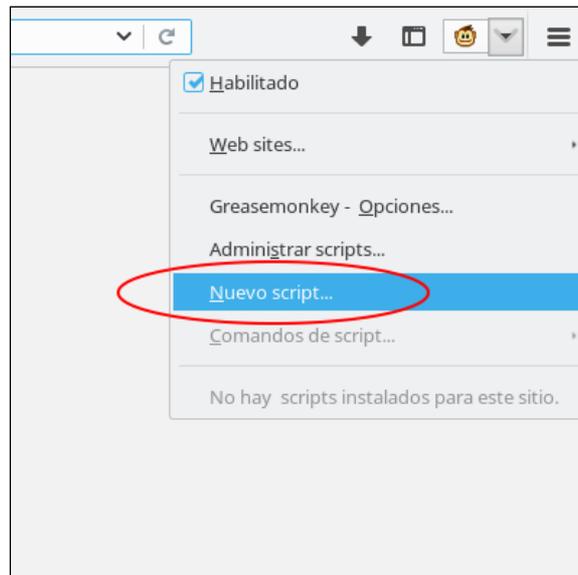


Figura A.2 - Greasemonkey: opción “Nuevo script”.

Figura A.3 - Greasemonkey: definición del script WebAdaptation.

4. Copiar el código fuente del componente diseñador (C) en el editor de código de Greasemonkey (ver figura A.4) y guardar los cambios.

```

Archivo  Editar  Ver  Ayuda
Abrir archivo...  Guardar  Guardar como...  Impresión prolija
1  // ==UserScript==
2  // @name      WebAdaptation
3  // @namespace  http://lifa.unlp.edu.ar
4  // @author    Mauricio Witkin, Ramon Serrano
5  // @description  A design tool for adapting a web application to mobile.
6  // @version    0.9
7  // @match     https://*/
8  // @match     http://*/
9  // @require   https://code.jquery.com/jquery-2.1.4.min.js
10 // @grant     GM_registerMenuCommand
11 // @noframes
12 // @run-at   document-end
13 // ==/UserScript==
14
15 (function() {
16   'use strict';
17
18   // Comandos del GreaseMonkey
19   GM_registerMenuCommand('Seleccionar template adaptativo', selectTemplate, "S");
20   GM_registerMenuCommand('Seleccionar elementos de la página', startEdit, "E");
21   GM_registerMenuCommand('Ver elementos seleccionados', viewSelected, "V");
22   GM_registerMenuCommand('Detener selección de elementos', stopEdit, "D");
23   GM_registerMenuCommand('Adaptar página', previewPage, "A");
24   GM_registerMenuCommand('Eliminar datos almacenados', delLocalSite, "L");
25   GM_registerMenuCommand('Importar configuración', importJson, "I");
26   GM_registerMenuCommand('Exportar configuración', exportJson, "X");
27

```

Línea 8, Col 27

Figura A.4 - Greasemonkey: código fuente del script WebAdaptation.

Una vez finalizadas estas tareas el Componente Diseñador está listo para su ejecución sobre algún sitio Web que se navegue con Mozilla Firefox (ver figura A.5).

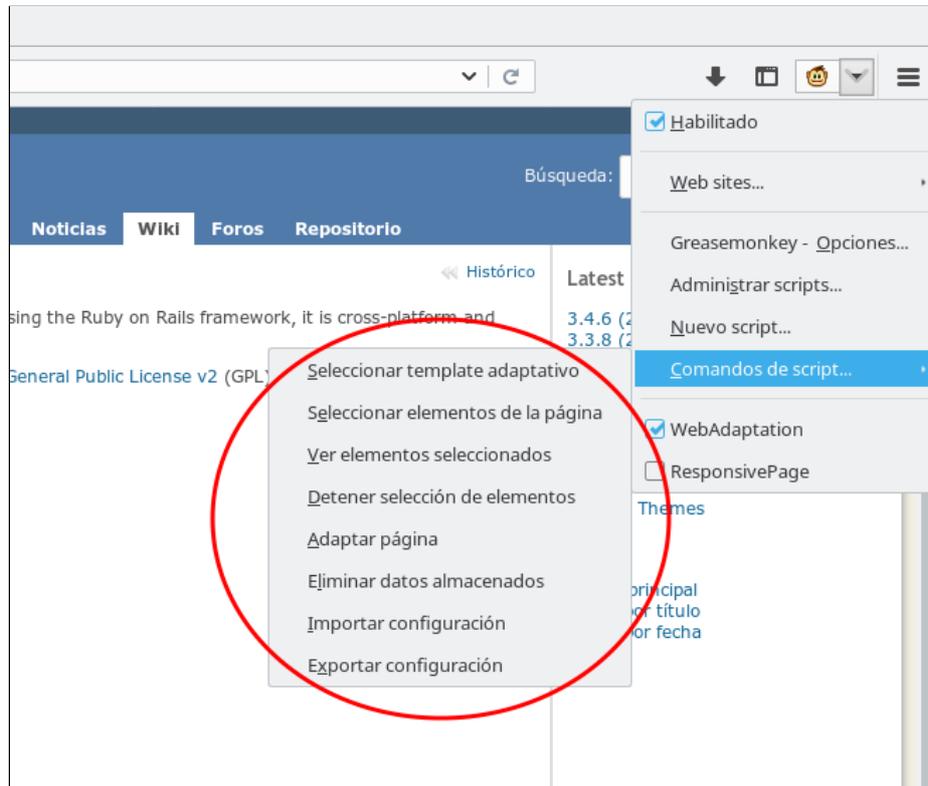


Figura A.5 - Greasemonkey: opciones principales de la herramienta de portabilización.

A.2. Instalación del Motor de Adaptaciones

Una vez finalizadas las tareas con el Componente Diseñador, sobre uno o varios sitios Web, podemos exportar la configuración con las portabilizaciones, para posteriormente agregarlas en el dispositivo móvil a través de la función de importación de la herramienta. Luego, mientras el usuario navega los sitios Web que fueron portabilizados, el Motor de Adaptaciones procesa las configuraciones y ejecuta de forma automática las adaptaciones en tiempo real.

Los requerimientos de software necesarios para ejecutar la portabilización de un sitio Web en un dispositivo móvil, por medio del Motor de Adaptaciones, se enumeran a continuación:

- Dispositivo móvil con sistema operativo Android versión 6.
- Dolphin Browser versión 11.5.11.
- Tampermonkey versión 4.2.5291.
- El componente *Motor de Adaptaciones* de la herramienta de portabilización disponible en:
<https://github.com/serranorm/UNLP/blob/master/scripts/ResponsivePage.user.js>

Los pasos a seguir para realizar la instalación son los siguientes:

1. Instalación de Dolphin Browser y Tampermonkey

Por medio del dispositivo móvil se debe acceder a la tienda de aplicaciones oficial de Android llamada Google Play, primero se debe instalar Dolphin Browser (ver figura A.6) y luego Tampermonkey (ver figura A.7).

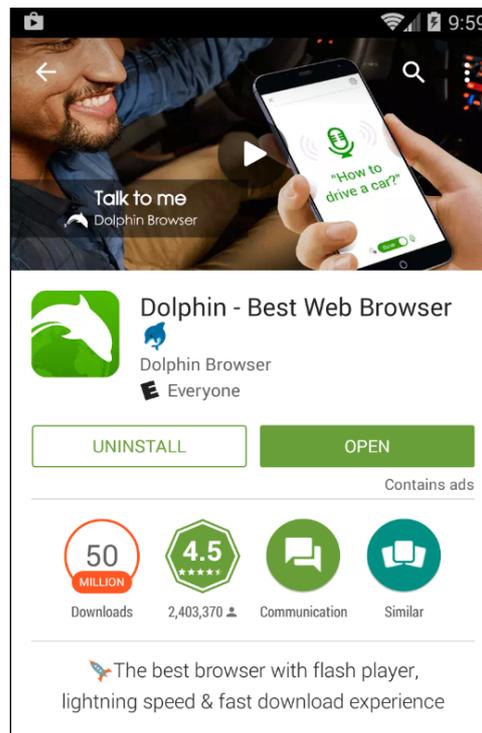


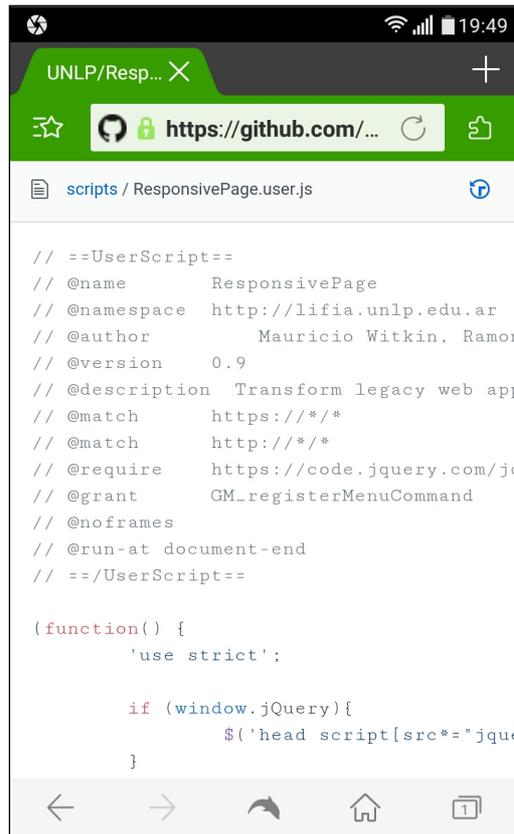
Figura A.6 - Dolphin Browser.



Figura A.7 - Tampermonkey.

2. Descarga del Motor de Adaptaciones

El script con el Motor de Adaptaciones (ver figura A.8) se encuentra disponible en la URL especificada en la lista de requerimientos de software que se mencionó anteriormente.



The screenshot shows a mobile browser interface with a green header. The address bar displays 'https://github.com/...' and the page title is 'scripts / ResponsivePage.user.js'. The main content area shows the following JavaScript code:

```
// ==UserScript==
// @name      ResponsivePage
// @namespace http://lifia.unlp.edu.ar
// @author    Mauricio Witkin, Ramon
// @version   0.9
// @description Transform legacy web app
// @match     https://**/*
// @match     http://**/*
// @require   https://code.jquery.com/jc
// @grant     GM_registerMenuCommand
// @noframes
// @run-at   document-end
// ==/UserScript==

(function() {
    'use strict';

    if (window.jQuery){
        $('head script[src*="jqque
    }
```

Figura A.8 - Script ResponsivoPage alojado en Github.

La descarga en el dispositivo móvil puede realizarse por medio del navegador Web (ver figura A.9).

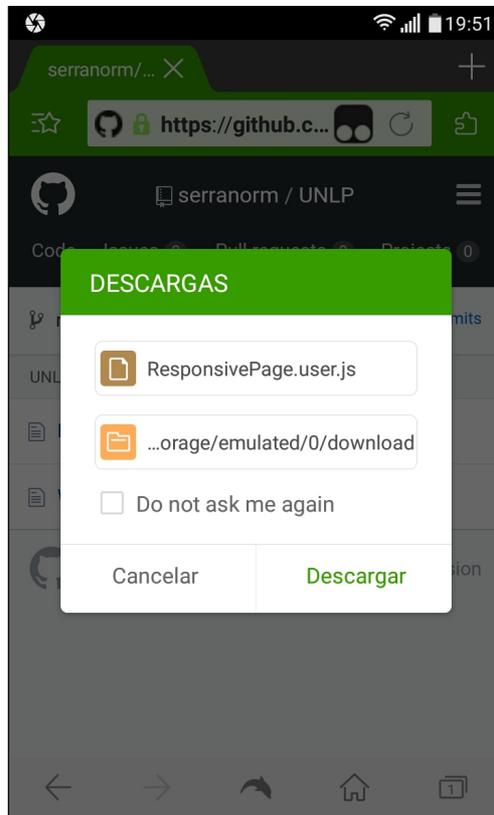


Figura A.9 - Descarga del script ResponsivePage.

3. Configuración del Motor de Adaptaciones en Tampermonkey

Abrir la aplicación Tampermonkey y seleccionar la pestaña **Utilidades**, luego completar en el campo **URL** el path dentro del dispositivo móvil del script “ResponsivePage.user.js” (ver figura A.10), descargado en el paso anterior.

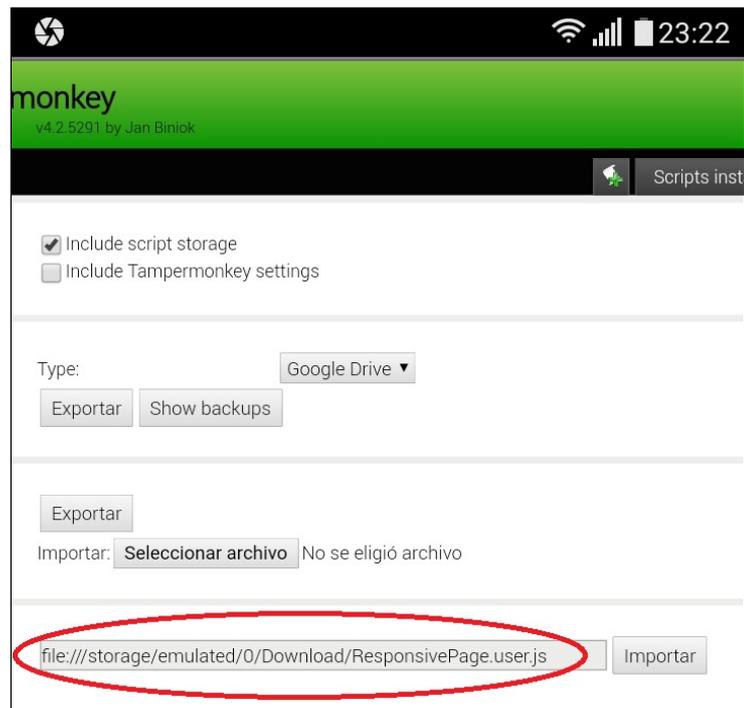


Figura A.10 - Tampermonkey: Datos para la Importación del script.

La siguiente acción es presionar el botón **Importar** y posteriormente el botón **Instalar** (ver figura A.11).

Nota: el path de descarga por defecto del script en un dispositivo Android es: "file:///storage/emulated/0/Download/ResponsivePage.user.js".



Figura A.11 - Tampermonkey: Instalación del script.

Una vez finalizada la instalación del Motor de Adaptaciones, es posible utilizar la opción de **Importar configuración** de la herramienta (ver figura A.12) para incorporar las configuraciones de las portabilizaciones de los sitios Web realizadas (ver figura A.13). Una configuración de demostración, con la portabilización del sitio Web Redmine se encuentra disponible en la URL: “https://github.com/serranorm/UNLP/blob/master/demo/redmine.json”.

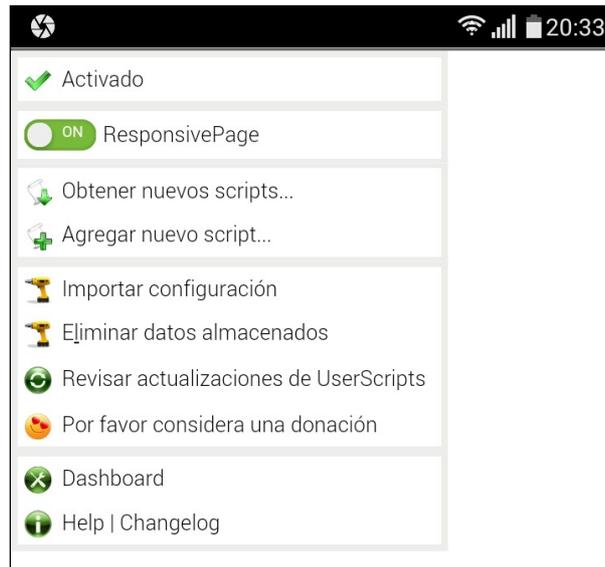


Figura A.12 - Tampermonkey: Opciones del Motor de Adaptaciones.



Figura A.13 - Tampermonkey: Importación de la configuración de los sitios portabilizados.

Una vez finalizados los pasos anteriores, el usuario puede navegar los sitios Web portabilizados (ver figura A.14).

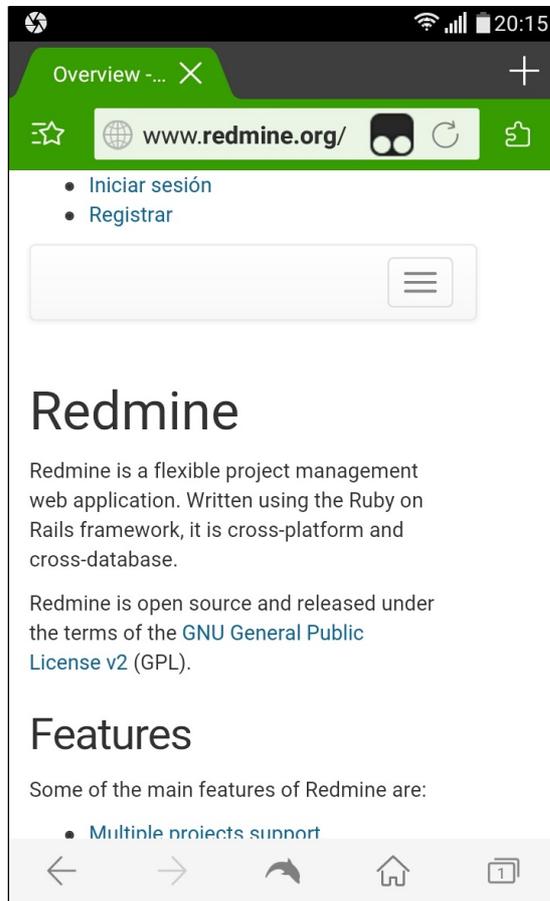


Figura A.14 - Navegación del sitio Redmine portabilizado.

Anexo B: Resumen de la metodología GQM

Es un enfoque que indica que para medir lo que está pasando y generar una base de conocimiento hacia el futuro se deben identificar las metas que se desean, identificar objetivos a medir de manera cuantificable y establecer un marco que permita interpretar la información respecto a los objetivos. El proceso GQM consta de 6 pasos.

Paso 1, Establecer las Metas: Establece el propósito que se persigue con el análisis.

Paso 2, Generación de Preguntas: Generar las preguntas (basadas en modelos) que definen objetivos de la manera más completa y cuantificable posible.

Paso 3, Especificación de Medidas: Especificar las medidas necesarias a ser recolectadas para contestar las preguntas y seguir la evolución del proceso y del producto con respecto a las metas.

Paso 4, Preparar Recolección de datos: Desarrollar mecanismos para la recolección de datos.

Paso 5, Recolectar, Validar y Analizar los datos para la toma de decisiones: Recoger, validar y analizar los datos en tiempo real, para proporcionar la retroalimentación de proyectos en una acción correctiva.

Paso 6, Analizar los datos para el logro de los objetivos y el aprendizaje: Analizar los datos una vez alcanzada una meta para determinar el grado de conformidad y hacer las recomendaciones para mejoras futuras.

Los primeros tres pasos son de definición, provee la estructura de proceso para pasar al concepto de métricas que proveen datos significativos para la toma de decisión. Los pasos del 4 al 6 son para recolectar y usar los resultados para la toma de decisiones.

Implementación

La implementación de GQM se hace a través de fases que están integradas con la planificación y gestión del proyecto. Las fases son cuatro.

Fase 1, Planificación: Se selecciona, define, caracteriza y planifica un proyecto para la aplicación de la métrica, obteniéndose como resultado un plan de proyecto. Comprende los pasos 1 al 5 del proceso.

Fase 2, Definición: Se definen medidas, preguntas e hipótesis, revisiones, chequeos y documenta el programa de medición. Esta comprende los tres primeros pasos.

Fase 3, Recopilación de Datos: Se recogen los datos reales de la medición. Esta fase corresponde a los pasos 4 y 5 del proceso.

Fase 4, Interpretación: Se procesan los datos recopilados para obtener respuestas a las preguntas definidas, a partir de las cuales se puede evaluar el logro de los objetivos planteados. Esta última fase aplica a los pasos 5 y 6 del proceso.

Anexo C: Resultados de las encuestas

Aquí se muestra unas capturas de las encuestas. La planilla de cálculo completa se puede descargar en el siguiente link:

“<https://github.com/serranorm/UNLP/tree/master/Encuestas/Analisis.xlsx>”.

En la figura C.1 podemos ver los datos digitalizados de las encuestas realizadas en Redmine con el uso de la herramienta de portabilización. Los mismos los visualizamos en el tab “E1-Redmine CH” de la planilla de cálculo indicada en el link de descarga.

#Persona	Encuesta	Tarea	problemas			Satisfacción	Finaliza	Tiempo	interacciones		
			Tamaño	Menú	Exceso				Zoom	Scroll	Clicks
1	1	1	0	0	1	4	1	67	0	13	0
		2	0	0	0	5	1	95	2	3	3
3	5	1	0	0	0	4	1	57	1	13	0
		2	0	0	0	4	1	91	0	10	3
5	9	1	0	0	0	3	1	52	0	15	0
		2	0	0	0	4	1	85	0	11	3
7	13	1	0	0	0	5	1	46	0	8	0
		2	0	0	0	5	1	88	0	10	3
9	17	1	0	0	0	4	1	38	0	6	0
		2	0	0	1	5	1	81	0	4	3
11	21	1	0	0	0	5	1	49	2	9	0
		2	0	0	0	4	1	93	0	3	3
13	25	1	0	0	0	5	1	51	0	8	0
		2	1	0	1	4	1	74	0	5	3
15	29	1	0	0	0	4	1	46	0	9	0
		2	0	1	1	4	1	69	0	6	3
17	33	1	0	0	0	4	1	44	1	8	0
		2	0	0	0	4	1	73	0	5	3
19	37	1	0	0	0	5	1	29	0	6	0
		2	0	0	0	5	1	65	0	4	3
21	41	1	0	0	0	4	1	50	0	7	0
		2	1	0	1	3	0		0	11	4
23	45	1	0	0	0	5	1	25	0	4	0
		2	0	0	0	5	1	58	0	2	3
25	49	1	0	0	0	4	1	45	0	12	0
		2	0	0	0	4	1	68	1	5	3
27	53	1	0	0	1	3	1	32	0	9	0
		2	0	0	0	4	1	72	0	5	3
29	57	1	1	0	0	4	1	28	0	5	0
		2	0	1	1	4	1	61	0	3	2
31	61	1	1	0	0	4	1	34	1	14	0
		2	1	0	0	4	1	67	0	5	3
33	65	1	0	0	0	5	1	33	0	7	0
		2	0	0	0	5	1	94	0	5	3

Figura C.1 - Tab: E1-Redmine CH

En la Figura C.2 se muestran los datos obtenidos en Pidgin sin uso de herramienta. Estos corresponden al tab "E2-Pidgin SH".

#Persona	Encuesta	Tarea	problemas			Satisfacción	Finaliza	Tiempo	interacciones		
			Tamaño	Menú	Exceso				Zoom	Scroll	Clicks
1	2	1	1	0	0	3	1	61	2	4	1
		2	1	0	1	2	1	78	11	6	3
3	6	1	0	1	1	4	1	56	6	8	1
		2	0	1	0	3	1	97	5	4	3
5	10	1	0	0	0	3	1	51	4	9	1
		2	0	0	0	3	1	72	2	3	3
7	14	1	1	0	0	3	1	48	5	7	1
		2	0	1	0	2	1	67	2	3	3
9	18	1	1	0	1	3	1	38	6	6	1
		2	0	1	0	3	1	64	3	1	3
11	22	1	0	0	0	2	1	44	3	5	1
		2	0	0	0	3	1	38	1	2	3
13	26	1	1	1	0	3	1	44	2	5	1
		2	0	0	0	3	1	35	2	3	3
15	30	1	1	0	0	2	1	43	3	7	1
		2	1	1	0	2	1	50	2	4	3
17	34	1	0	0	0	4	1	22	2	3	1
		2	0	0	0	4	1	35	2	3	3
19	38	1	0	0	0	3	1	47	4	9	1
		2	0	1	1	3	1	85	5	4	5
21	42	1	1	1	1	3	1	42	5	11	1
		2	1	1	0	3	1	51	7	8	3
23	46	1	0	0	0	3	1	35	2	7	1
		2	0	0	1	2	1	47	1	2	3
25	50	1	1	0	0	3	1	28	5	11	1
		2	1	1	1	3	1	44	4	5	3
27	54	1	0	0	0	2	1	39	4	9	1
		2	0	0	0	2	1	55	3	6	3
29	58	1	0	1	1	3	1	27	2	3	1
		2	0	1	1	4	1	33	2	3	3
31	62	1	1	0	1	4	1	33	5	12	1
		2	1	0	0	3	1	58	3	7	3
33	66	1	1	0	1	4	1	57	5	10	1
		2	0	0	1	4	1	64	2	4	3

Figura: C.2 - Tab: E2-Pidgin SH

En la Figura C.3 observamos los datos obtenidos en Redmine sin uso de herramienta. Corresponden al tab "E3-Redmine SH".

#Persona	Encuesta	Tarea	problemas			Satisfacción	Finaliza	Tiempo	interacciones		
			Tamaño	Menú	Exceso				Zoom	Scroll	Clicks
2	3	1	1	0	0	3	1	62	5	16	0
		2	1	0	1	3	1	93	9	9	2
4	7	1	1	0	0	4	1	43	9	13	0
		2	1	1	0	3	1	71	5	10	2
6	11	1	0	0	0	2	1	51	4	15	0
		2	0	1	0	3	1	76	3	11	2
8	15	1	1	0	0	4	1	48	5	15	0
		2	1	1	0	3	1	81	3	11	2
10	19	1	0	0	0	4	1	40	3	9	0
		2	0	0	0	4	1	67	2	7	2
12	23	1	1	0	1	3	1	56	4	15	0
		2	1	0	1	2	0		6	21	5
14	27	1	0	0	0	3	1	48	4	8	0
		2	0	1	0	3	1	76	3	11	2
16	31	1	0	0	0	4	1	52	1	20	0
		2	1	0	1	3	1	83	3	4	2
18	35	1	1	0	1	2	1	69	7	16	0
		2	1	1	1	2	1	91	5	12	2
20	39	1	0	0	0	3	1	36	2	11	1
		2	0	0	0	4	1	58	2	5	2
22	43	1	0	0	1	2	1	55	5	12	0
		2	0	1	0	3	1	65	3	7	2
24	47	1	0	0	1	3	1	47	5	18	0
		2	0	0	0	4	1	71	2	6	2
26	51	1	0	0	1	3	1	84	6	22	0
		2	0	1	1	3	1	92	5	11	4
28	55	1	0	0	0	4	1	48	4	9	0
		2	0	1	0	4	1	77	3	5	2
30	59	1	1	0	1	4	1	46	3	11	0
		2	0	1	0	4	1	68	2	6	2
32	63	1	0	0	0	3	1	51	4	15	0
		2	0	1	0	4	1	76	3	11	2
34	67	1	1	0	0	3	1	42	2	8	0
		2	1	0	0	3	1	61	1	6	2

Figura: C.3 - Tab: E3-Redmine SH

Por último, en la Figura C.4 se presentan los datos de Pidgin con uso de herramienta. Corresponden al tab "E4-Pidgin CH".

#Persona	Encuesta	Tarea	problemas			Satisfacción	Finaliza	Tiempo	interacciones		
			Tamaño	Menú	Exceso				Zoom	Scroll	Clicks
2	4	1	0	0	1	4	1	72	0	20	2
		2	0	0	0	3	1	47	0	1	3
4	8	1	0	0	0	4	1	45	0	12	2
		2	0	0	0	4	1	34	0	1	3
6	12	1	0	0	0	4	1	48	0	10	2
		2	0	0	0	3	1	54	0	0	3
8	16	1	1	0	0	4	1	28	0	8	2
		2	0	0	0	3	1	45	0	0	3
10	20	1	0	0	1	4	1	35	0	14	2
		2	0	0	0	4	1	42	0	1	3
12	24	1	0	0	1	5	1	32	0	6	2
		2	0	0	0	5	1	38	0	1	3
14	28	1	0	0	0	4	1	24	0	11	2
		2	0	0	0	4	1	42	0	1	3
16	32	1	0	1	1	5	1	32	0	7	2
		2	0	0	0	4	1	34	0	0	3
18	35	1	0	0	0	4	1	41	0	15	2
		2	0	0	0	4	1	36	0	0	3
20	40	1	0	0	0	5	1	37	0	8	2
		2	0	0	0	5	1	44	0	0	3
22	44	1	0	0	1	4	1	33	0	9	2
		2	0	0	0	4	1	47	0	2	3
24	48	1	0	0	0	4	1	24	0	8	2
		2	0	0	0	5	1	35	0	0	3
26	52	1	0	1	0	3	1	64	0	14	2
		2	0	0	0	4	1	71	0	2	3
28	56	1	1	0	0	3	1	31	0	6	2
		2	1	0	0	4	1	36	0	0	3
30	60	1	0	1	0	4	1	41	0	9	2
		2	0	0	0	4	1	29	0	0	3
32	64	1	0	0	1	4	1	50	0	16	2
		2	0	0	0	4	1	34	0	0	3
34	68	1	0	0	0	4	1	38	0	7	2
		2	0	0	0	5	1	32	0	0	3

Figura: C.4 - Tab: E4-Pidgin CH