

UNIVERSIDAD DE OVIEDO



Centro Internacional De Postgrado

TRABAJO FIN DE MÁSTER

“Herramienta web para la medición de accesibilidad y usabilidad de una aplicación Android”

DIRECTOR: Begoña Cristina Pelayo García Bustelo

CODIRECTOR: Jordán Pascual Espada

AUTOR: Ismael Ramos Silván

Vº Bº del Director del
Proyecto

Resumen

Este trabajo fin de máster se basa en una herramienta web para analizar interfaces Android. El principal objetivo es encontrar posibles fallos en el diseño de estas interfaces que puedan provocar graves problemas, tanto de usabilidad, accesibilidad y rendimiento.

La idea es que el usuario suba un fichero XML con la interfaz Android, se analice, y posteriormente, se genere un informe para que pueda ver aquellos aspectos en los que falla. Si lo desea también es posible ver lo que está correcto.

La gran mayoría de los posibles problemas que se analizan aparecen en la guía de desarrolladores de Google, en la que se recopilan todas las buenas prácticas sobre Android.

Esta herramienta estará disponible en Internet para cualquier tipo de usuario bajo un dominio específico, y por lo tanto cualquier desarrollador de Android podrá utilizarla, por supuesto de forma gratuita.

Las tecnologías con las que se han usado en este proyecto son PHP y MySQL para la programación en el servidor, y un framework de CSS llamado SemanticUI y jQuery para la programación en el cliente. También se han utilizado, las herramientas de desarrolladores de Google, así como Google Analytics.

Palabras Clave

Herramienta web, analizador, interfaces Android, PHP, Productividad, generador de informes.

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO	8
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	8
CAPÍTULO 2. INTRODUCCIÓN	9
2.1 JUSTIFICACIÓN DEL PROYECTO	9
2.2 OBJETIVOS DEL PROYECTO	10
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL	11
2.3.1 <i>Evaluación de Alternativas</i>	12
2.3.2 <i>Selección de la solución</i>	14
CAPÍTULO 3. ASPECTOS TEÓRICOS	15
3.1 FICHEROS XML	15
3.2 INTERFACES ANDROID	16
3.3 PHP	16
3.4 SIMPLEXML	17
3.5 PHPMAILER	18
3.6 TCPDF	18
3.7 USABILIDAD	18
3.8 ACCESIBILIDAD	19
3.9 RESPONSIVE DESIGN	19
3.10 INTERNACIONALIZACIÓN	20
3.11 MATERIAL DESIGN	21
CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DEL PRESUPUESTO	22
4.1 PLANIFICACIÓN Y LISTA DE TAREAS	22
4.2 RESUMEN DEL PRESUPUESTO	26
CAPÍTULO 5. ANÁLISIS	27
5.1 ALCANCE DEL SISTEMA	27
5.1.1 <i>Descripción de los actores</i>	27
5.2 IDENTIFICACIÓN DE LOS SUBSISTEMAS	28
5.3 DIAGRAMAS DE CASOS DE USO	29
5.3.1 <i>Subsistema del informe</i>	29
5.4 DIAGRAMA DE CLASES	32
5.4.1 <i>Modelo de clases del sistema</i>	32
5.5 ANÁLISIS DE INTERFACES DE USUARIO	33
5.5.1 <i>Mapa de navegación</i>	33
5.5.2 <i>Descripción de las Interfaces</i>	33
5.6 ANÁLISIS DEL PLAN DE PRUEBAS	38
5.6.1 <i>Pruebas funcionales</i>	38
5.6.2 <i>Pruebas de rendimiento</i>	39
5.6.3 <i>Pruebas de Usabilidad</i>	39
5.6.4 <i>Pruebas de accesibilidad</i>	40
CAPÍTULO 6. DISEÑO DEL SISTEMA	41
6.1 ARQUITECTURA DEL SISTEMA	41

6.1.1	Entorno tecnológico de implantación	41
6.1.2	Dominio	41
6.1.3	Diagrama de clases.....	42
6.1.4	Patrones de diseño implementados.....	42
6.1.5	Diagrama de Despliegue.....	43
6.1.6	Guías de diseño comunes	44
6.1.7	Reescritura de URLs	45
6.2	FALLOS ANALIZADOS	46
6.2.1	Formato incorrecto	46
6.2.2	Uso de "wrap_content" y "match_parent".....	46
6.2.3	Usa RelativeLayout	47
6.2.4	Utiliza píxeles independientes de la densidad	47
6.2.5	Añade descripciones de contenido	48
6.2.6	Diseño de navegación direccional.....	48
6.2.7	Separar el diseño del contenido.....	48
6.2.8	Usa compound drawables	49
6.2.9	Optimizar mezclando.....	49
6.2.10	Crear listas y tarjetas.....	50
6.2.11	Crear fichas con estilo de Material Design	50
6.2.12	Tamaño de letra	51
6.2.13	Cadenas literales	51
6.2.14	El texto de los botones deberían ser verbos	52
6.2.15	El texto dinámico probablemente debe ser seleccionable.....	52
6.2.16	Menús de Android	53
6.2.17	Atributo inputType desaparecido	53
6.2.18	Sobrescribiendo parámetros del layout.....	53
6.3	DISEÑO DE LA BASE DE DATOS.....	55
6.3.1	Descripción del SGBD Usado.....	55
6.3.2	Integración del SGBD en Nuestro Sistema	55
6.3.3	Diagrama E-R.....	55
6.4	DISEÑO DETALLADO DEL SISTEMA	57
6.4.1	Subsistema del informe	57
6.4.2	Otros archivos y funciones.....	64
6.5	DIAGRAMAS DE INTERACCIÓN DE LOS ESCENARIOS DEL SUBSISTEMA	66
6.5.1	Subsistema del informe	66
6.6	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	69
6.6.1	PHPStorm de JetBrains (IntelliJ)	69
6.6.2	MySQL Workbench de Oracle	70
6.6.3	Filezilla	71
6.6.4	Herramientas para Webmasters de Google	72
6.6.5	Google Analytics	73
CAPÍTULO 7. DESARROLLO DE LAS PRUEBAS		74
7.1	PREPARACIÓN DE LAS PRUEBAS	74
7.1.1	Pruebas funcionales.....	74
7.1.2	Pruebas de Usabilidad	75
7.1.3	Pruebas de Rendimiento.....	76
7.2	IMPLEMENTACIÓN DE LAS PRUEBAS	77
7.2.1	Pruebas funcionales.....	77
7.2.2	Pruebas de Usabilidad	82

7.2.3	Pruebas de Accesibilidad	86
7.2.4	Pruebas de Rendimiento.....	93
CAPÍTULO 8.	MANUALES DEL SISTEMA.....	97
8.1	MANUAL DE INSTALACIÓN	97
8.1.1	Instalación en un servicio de Hosting.....	97
8.1.2	Instalación en localhost	99
8.2	MANUAL DE USUARIO	101
8.2.1	Página de inicio.....	101
8.2.2	Página del informe.....	102
8.2.3	Página de contacto.....	103
8.3	MANUAL DEL PROGRAMADOR.....	104
8.3.1	Requisitos hardware	104
8.3.2	Recursos humanos.....	104
8.3.3	Desarrollo futuro	104
CAPÍTULO 9.	CONCLUSIONES Y AMPLIACIONES.....	106
9.1	CONCLUSIONES.....	106
9.2	AMPLIACIONES	107
CAPÍTULO 10.	REFERENCIAS BIBLIOGRÁFICAS	108
10.1	LIBROS Y ARTÍCULOS.....	108
10.2	REFERENCIAS EN INTERNET	109
CAPÍTULO 11.	ANEXOS	110

Índice de Figuras

Ilustración 1: Interfaces de aplicaciones Android	9
Ilustración 2: Logo de Android	11
Ilustración 3: PHP y MySQL	14
Ilustración 4: Diseño adaptado a diferentes dispositivos	19
Ilustración 5: Ficha de ejemplo	20
Ilustración 6: Aplicación con estilo Material Design	21
Ilustración 7: Diagrama de general de subsistemas.....	28
Ilustración 8: Casos de uso del subsistema Informe	29
Ilustración 9: Diagrama de clases del análisis.....	32
Ilustración 10: Mapa de navegación	33
Ilustración 11: Prototipo de la página de inicio	34
Ilustración 12: Prototipo de seleccionar archivo	34
Ilustración 13: Prototipo del informe	35
Ilustración 14: Prototipo del PDF	36
Ilustración 15: Prototipo de la página de contacto.....	37
Ilustración 16: Prototipo de los avisos legales.....	38
Ilustración 17: Distintos tipos de pruebas	39
Ilustración 18: Diagrama de clases del diseño	42
Ilustración 19: Diagrama de despliegue	43
Ilustración 20: Diagrama E-R	55
Ilustración 21: Contenido de la tabla issues	56
Ilustración 22: Clase Report	57
Ilustración 23: Clase ReportsController	59
Ilustración 24: Clase ReportView	63
Ilustración 25: Diagrama de interacción del caso de uso seleccionar XML.....	66
Ilustración 26: Diagrama de interacción del caso de uso Analiza XML.....	67
Ilustración 27: Diagrama de interacción del caso de uso Descargar ejemplos	67
Ilustración 28: Diagrama de interacción del caso de uso Crear PDF.....	68
Ilustración 29: Entorno PHPStorm.....	69
Ilustración 30: Entorno MySQL Workbench	70
Ilustración 31: Entorno Filezilla	71
Ilustración 32: Ejemplo de Google Webmasters Tools.....	72
Ilustración 33: Ejemplo de Google Analytics	73
Ilustración 34: Resultados nivel A de la página de inicio	87
Ilustración 35: Resultados nivel A de la página de contacto	87
Ilustración 36: Resultados nivel AA	88
Ilustración 37: Resultados de rendimiento de Google Page Speed	94
Ilustración 38: Resultados de rendimiento de YSlow	95
Ilustración 39: Timeline de carga de la página.....	96
Ilustración 40: Página de inicio	101
Ilustración 41: Página del informe	102
Ilustración 42: Página de contacto	103

Capítulo 1. Memoria del Proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

Actualmente Android dispone de muchas interfaces para una gama muy amplia de dispositivos, y es necesario que estas interfaces sean accesibles y usables para los usuarios.

En este proyecto se ha construido una herramienta web, para que los desarrolladores de aplicaciones Android, puedan obtener un informe de qué aspectos se pueden mejorar respecto a dichas interfaces.

Google ha ofrecido una serie de buenas prácticas que las aplicaciones deben cumplir, y estas serán las que se traten de comprobar de cara a generar un informe final con los fallos a corregir.

Cualquier desarrollador de Android que quiera construir una aplicación accesible y usable, podrá entrar en la web, subir los ficheros XML de las interfaces que está construyendo y ver qué es lo que tiene que cambiar y mejorar para cumplir con las buenas prácticas.

Android es un mundo en constante movimiento, en el que las prácticas y recomendaciones cambian muy rápido. Esto puede ser un problema para los desarrolladores que a menudo tienen malas costumbres y, por ese motivo, este proyecto puede serles de gran ayuda y utilidad.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

Cada día que pasa, los dispositivos móviles y más concretamente Android toman gran importancia en nuestras vidas. Los usamos a diario y por eso necesitamos que las aplicaciones sean eficientes y presenten unas interfaces pensadas para los seres humanos y que cumplan una serie de criterios acordados por la comunidad de desarrolladores. Estos criterios buscan el beneficio del usuario final, y Google a través de su guía de desarrolladores¹, proporciona una amplia lista de todos ellos. Aparte de Google, la comunidad de Internet también ofrece aspectos relevantes a tener en cuenta para esas interfaces.

Dado que todo esto, se puede analizar y procesar, sería muy beneficioso para un desarrollador de aplicaciones Android, tener una herramienta, que esté online y disponible en Internet, mediante la que poder analizar las interfaces, y ese es el motivo principal del proyecto. Además, el usuario puede almacenar el informe generado y estudiarlo para mejorar sus conocimientos de Android y no volver a cometer los mismos errores.



Ilustración 1: Interfaces de aplicaciones Android

¹ <http://developer.android.com/index.html>

2.2 Objetivos del Proyecto

Los objetivos de este proyecto quedan resumidos en la siguiente tabla:

Objetivo	Descripción
Disponer de una herramienta online	Con el motivo de que cualquier usuario pueda acceder a ella, es preciso que sea desplegada en un servidor bajo un dominio específico.
Generar un informe de errores	El usuario debe poder subir el fichero XML de la interfaz a la web, y ver un informe con los fallos y aciertos que tiene dicha interfaz.
Categorías	El informe debe estar organizado en categorías y debe aportar información suficiente para que el usuario comprenda el problema a tratar de forma sencilla.
Información adicional	Para cada problema que se analice, se deberá ofrecer un enlace con más información ampliada por si acaso el usuario la necesita.
Posibilidad de filtros	El informe debe tener la posibilidad de filtrar el tipo de mensaje ya sea un fallo o un acierto.
Página de contacto	Deberá existir una página de contacto para que el usuario pueda consultar cualquier cosa relacionada con la herramienta.
Herramientas de análisis	Se deberá hacer uso de herramientas de análisis estadístico para poder mejorar la web con el tiempo.
Avisos legales	La web debe disponer de aviso legal y política de privacidad y si fuese necesario política de cookies.
HTML válido	Todo el código de la web, debe validar para HTML5, sin advertencias.
Responsive Design	Para una mayor usabilidad y accesibilidad, la web debe adaptarse a cualquier dispositivo de forma automática.
Internacionalización	La web deberá estar disponible en dos idiomas como mínimo, español e inglés. Además deberá poderse añadir más, de forma sencilla.
Seguridad	Para que cualquier usuario pueda hacer uso de una forma segura, la web deberá contar con validaciones en los formularios tanto en el cliente como en el servidor.
Guardar informe en PDF	De cara a mejorar la usabilidad de la herramienta, será necesario que el usuario pueda guardar el informe completo en PDF.
SEO	Será necesario que la página se encuentre entre las 5 primeras páginas para los términos relacionados (Android, analyzer, interfaces)

2.3 Estudio de la Situación Actual

En la actualidad, no existe ninguna herramienta online que se encargue de analizar elementos concretos respecto a la accesibilidad y usabilidad en la codificación de las interfaces Android. Es por tanto, una situación privilegiada para desarrollar un proyecto como este, y poder implementar esta herramienta en este momento en concreto.

Una ventaja importante del proyecto es que cualquier usuario sin instalar ningún tipo de aplicación, tendrá disponible una herramienta online y gratuita para analizar las interfaces con las que esté trabajando, y de esa forma obtener una visión general de cómo son sus interfaces y de qué fallos está cometiendo a la hora de implementarlas. Esto le permitirá aprender rápidamente, cuáles son los principales consejos y buenas prácticas recomendadas por Google y por la comunidad de desarrolladores.

Por otra parte existe alguna herramienta con una funcionalidad parecida, como puede ser Lint. Según la Wikipedia “originalmente lint era el nombre de una herramienta de programación utilizada para detectar código sospechoso, confuso o incompatible entre distintas arquitecturas en programas escritos en C; es decir, errores de programación que escapan al habitual análisis sintáctico que hace el compilador. En la actualidad, se utiliza este término para designar a herramientas que realizan estas tareas de comprobación en cualquier lenguaje de programación.”

Existe una herramienta para Android de este tipo, pero la principal desventaja es que se centra en su totalidad, en analizar el código para detectar referencias no válidas, código muerto o comprobación de tipos. Aparte de eso, la herramienta debe ser instalada como plugin para el entorno de desarrollo en el que se esté implementando la aplicación Android, y el usuario no tiene ninguna perspectiva visual de cómo están implementadas las interfaces.

Sin embargo este proyecto se ha centrado en aspectos concretos de usabilidad y accesibilidad de las interfaces Android, el usuario puede ver un informe de manera gráfica e imprimirlo o guardarlo en PDF.



Ilustración 2: Logo de Android

2.3.1 Evaluación de Alternativas

Al comienzo del proyecto, se han estudiado diferentes alternativas que se explican a continuación.

2.3.1.1 *Aplicación Java*

2.3.1.1.1 Descripción

Esta posibilidad, se trata de una aplicación JAVA que reciba un fichero XML con la interfaz y devuelva una lista con los posibles errores. Esta solución tiene mucho en común con la asignatura de Arquitecturas Web y Procesamiento de Lenguajes.

2.3.1.1.2 Ventajas

- Sencillez a la hora de desarrollar la aplicación.
- Mucha documentación para librerías de procesamiento de ficheros XML
- Buen conocimiento del lenguaje de programación JAVA

2.3.1.1.3 Desventajas

- El usuario deberá tener instalado JAVA en su equipo
- Muy pocas posibilidades de llegar a usuarios potenciales (desarrolladores de Android)
- Gran dificultad a la hora de generar la lista de errores
- Informe muy poco útil para el usuario

2.3.1.2 *Aplicación Web basada en jQuery*

2.3.1.2.1 Descripción

Esta solución se basa en la implementación de una web dónde toda la implementación para analizar el fichero XML de la interfaz se realiza en la parte del cliente usando jQuery.

2.3.1.2.2 Ventajas

- El usuario podrá ver el informe a través de Internet accediendo a la web
- Informe visual con posibilidad de generar un PDF
- No será necesaria ninguna instalación de ningún programa o lenguaje
- Si la página está bien posicionada, hay grandes probabilidades de llegar a desarrolladores Android
- Gran cantidad de documentación

2.3.1.2.3 Desventajas

- Se necesita un alto conocimiento de jQuery para desarrollar todo el proyecto en el cliente. Desconocimiento de funciones de jQuery para parsear ficheros XML.

- La funcionalidad de la web se puede ver restringida a las que ofrece jQuery
- Problemas de seguridad en el envío de formularios
- Necesidad de Javascript activado para que funcione
- Múltiples versiones de jQuery y problemas de compatibilidad

2.3.1.3 *Aplicación Web basada en JAVA*

2.3.1.3.1 **Descripción**

Esta solución se basa en la implementación de una web basada en JAVA. En la parte del cliente, se pueden utilizar tecnologías como jQuery, pero todo el procesamiento del fichero se hace en la parte del servidor.

2.3.1.3.2 **Ventajas**

- El usuario podrá ver el informe a través de Internet accediendo a la web
- Informe visual con posibilidad de generar un PDF
- No será necesaria ninguna instalación de ningún programa o lenguaje
- Si la página está bien posicionada, hay grandes probabilidades de llegar a desarrolladores Android
- Buen conocimiento del lenguaje de programación JAVA
- Gran cantidad de documentación
- Seguridad en la parte del servidor
- No es necesario tener Javascript activado

2.3.1.3.3 **Desventajas**

- Problemas de eficiencia en la parte del servidor (Compilador de JAVA)
- Principales desventajas del propio lenguaje, como tipado fuerte, sintaxis o procesado en tiempo de compilación

2.3.1.4 *Aplicación Web basada en PHP*

2.3.1.4.1 **Descripción**

Esta solución se basa en la implementación de una web basada en PHP. En la parte del cliente, se pueden utilizar tecnologías como jQuery, pero todo el procesamiento del fichero se hace en la parte del servidor.

2.3.1.4.2 **Ventajas**

- El usuario podrá ver el informe a través de Internet accediendo a la web
- Informe visual con posibilidad de generar un PDF
- No será necesaria ninguna instalación de ningún programa o lenguaje
- Si la página está bien posicionada, hay grandes probabilidades de llegar a desarrolladores Android

- Buen conocimiento del lenguaje de programación PHP
- Gran cantidad de documentación
- Seguridad en la parte del servidor
- No es necesario tener Javascript activado
- Desarrollo muy rápido y ágil

2.3.1.4.3 Desventajas

- Problemas de seguridad en el lenguaje
- Escalabilidad reducida
- Sintaxis del lenguaje poco clara

2.3.2 Selección de la solución

Se ha optado por la solución de una aplicación web basada en PHP.

2.3.2.1 Explicación de la alternativa elegida

Básicamente existen dos soluciones válidas, que permiten cumplir con todos los objetivos del proyecto. Estas son, una aplicación web basada en JAVA o la otra basada en PHP.

La elección por la alternativa basada en PHP, radica principalmente en que el lenguaje permite desarrollar todo el proyecto de forma más ágil y rápida. Además de tener un gran conocimiento de librerías para el procesado de ficheros XML en PHP.

Estos aspectos, añadidos a que la curva de aprendizaje de PHP es muy baja dan como resultado que sea la mejor alternativa posible en este caso.



Ilustración 3: PHP y MySQL

Capítulo 3. Aspectos Teóricos

3.1 Ficheros XML

Según la Wikipedia “XML es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de trozos de información. Estas partes se llaman elementos, y se las señala mediante etiquetas.”

Un ejemplo de fichero XML podría ser:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">
<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail> Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento
      </Asunto>
      <Parrafo>
        Este es mi documento
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

3.2 Interfaces Android

Las interfaces Android conforman todo aquello que el usuario puede ver en la aplicación. Android provee de gran cantidad de componentes que se pueden incluir, como botones, imágenes, listas, texto y otros módulos como diálogos, notificaciones o menús.

Por tanto, como Google dispone de una documentación² muy completa en la que especifica qué características tiene que cumplir esas interfaces, este proyecto se encarga de analizar las interfaces, en función de dichos criterios.

Estas interfaces se definen en ficheros XML, como se muestra a continuación:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I am a Button" />
</LinearLayout>
```

Además es posible que haya elementos dentro de elementos, es decir de forma recursiva. Todos los elementos, usan un espacio de nombres, y tienen múltiples atributos, como "id", "layout_width" o "layout_height".

3.3 PHP

PHP Documentación, "PHP es un lenguaje de código abierto especialmente orientado para el desarrollo web y que puede ser incrustado en HTML. El código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP."

Se trata por tanto, de un lenguaje con una curva de aprendizaje muy sencilla y que ha permitido desarrollar el proyecto de una forma rápida y ágil.

Ejemplo de código PHP:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <?php
      echo ";Hola, soy un script de PHP!";
    ?>
```

² <https://developer.android.com/training/best-ui.html>


```
</body>  
</html>
```

La principal herramienta que se ha utilizado para analizar los ficheros XML, ha sido una extensión de PHP que se llama SimpleXML, que según PHP “proporciona un conjunto de herramientas muy simple y fácil de usar para convertir XML a un objeto que pueda ser procesado con selectores de propiedades normales e iteradores de arrays”.

Una vez que se ha convertido el XML en un objeto manejable, es muy sencillo acceder a los atributos y nodos, como se muestra a continuación:

```
<?php  
$string = <<<XML  
<?xml version='1.0'?>  
<document>  
  <cmd>login</cmd>  
  <login>Richard</login>  
</document>  
XML;  
  
$xml = simplexml_load_string($string);  
print_r($xml);  
$login = $xml->login;  
print_r($login);  
$login = (string) $xml->login;  
print_r($login);  
?>
```

3.4 SimpleXML

Esta es la librería principal para analizar el fichero XML. Tal y como refleja la documentación de PHP “la extensión SimpleXML proporciona un conjunto de herramientas muy simple y fácil de usar para convertir XML a un objeto que pueda ser procesado con selectores de propiedades normales e iteradores de arrays.”.

Para ver un ejemplo claro de esta sencillez, se muestra el siguiente fragmento de código:

```
<?php  
$xml = '<?xml version="1.0" encoding="UTF-8" ?>  
<rss>  
  <channel>  
    <item>  
      <title>Ismael</title>  
    </item>  
  </channel>  
</rss>';  
  
$xml = simplexml_load_string($xml);  
  
echo $xml->channel->item->title; // Esto devolvería "Ismael"  
?>
```

3.5 PHPMailer

Otra librería que se ha usado en este proyecto es PHPMailer. Se trata de la librería más usada y reconocida para enviar correos. También es muy sencilla de usar, como se puede ver en este ejemplo:

```
require_once('../class.phpmailer.php');

$mail = new PHPMailer();
$body = file_get_contents('contenido.html');
$mail->SetFrom('email@remitente.com', 'Nombre completo');
$mail->AddReplyTo("email@remitente.com","Nombre Completo");
$address = "email@destinatario.com";
$mail->AddAddress($address, "Nombre completo");
$mail->Subject = "Envío de email con PHPMailer en PHP";
$mail->AltBody = "Cuerpo alternativo del mensaje";
$mail->MsgHTML($body);
$mail->AddAttachment("ruta/archivo_adjunto.gif");

if(!$mail->Send()) {
    echo "Error al enviar el mensaje: " . $mail->ErrorInfo;
} else {
    echo "Mensaje enviado!!";
}
```

3.6 TCPDF

En el informe el usuario tiene la posibilidad de guardarlo en un PDF. Esta librería en PHP permite generar documentos PDF en pocos pasos, y prácticamente sin ninguna configuración. Además se trata de un proyecto de código abierto. A continuación se muestra un ejemplo:

```
$pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true, 'UTF-8', false);
$pdf->AddPage();

$html = "<h1>Contenido</h1>";

$pdf->SetAutoPageBreak(TRUE, 0);
$pdf->setPrintFooter(false);

$pdf->writeHTML($html, true, false, true, false, '');
```

3.7 Usabilidad

Estos conceptos son importantes para el Proyecto, ya que la gran mayoría de los aspectos que se analizan en el informe, tienen que ver con la usabilidad y la accesibilidad.

Según la Wikipedia, el término Usabilidad “se refiere a la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto. En interacción persona-ordenador, la usabilidad se refiere a la claridad y la elegancia con que se diseña la interacción con un programa de ordenador o un sitio web.”

A lo largo de los años, se han acordado unas buenas prácticas y ciertos criterios que deben ocurrir para que la usabilidad sea buena para el usuario. Aunque esto también es aplicable

para la web, si las interfaces de Android no son usables, los usuarios rechazarán la aplicación y escogerán otra de las miles que hay. Es por tanto necesario cuidar estos aspectos a la hora de desarrollar una interfaz, y con este proyecto, los desarrolladores de Android tienen una herramienta que lo hace por ellos.

3.8 Accesibilidad

Otro concepto muy importante es el de la accesibilidad. Cualquier desarrollador desea tener el mayor número posible de usuarios, y para esto es necesario que además de ser usable, las aplicaciones sean accesibles.

Según Wikipedia “la accesibilidad es el grado en el que todas las personas pueden utilizar un objeto, visitar un lugar o acceder a un servicio, independientemente de sus capacidades técnicas, cognitivas o físicas. Es indispensable e imprescindible, ya que se trata de una condición necesaria para la participación de todas las personas independientemente de las posibles limitaciones funcionales que puedan tener”.

Por eso, este proyecto también se ha centrado en todos aquellos factores que intervienen para que una interfaz sea accesible y facilitar el trabajo de los desarrolladores.

3.9 Responsive Design

Este concepto representa una filosofía en cuanto al diseño web. Se trata básicamente, de adaptar el diseño de la web al dispositivo que la está visualizando. Para este proyecto en concreto, la web es totalmente *responsive* y para ello se ha hecho uso de un framework de CSS llamado SemanticUI³.



Ilustración 4: Diseño adaptado a diferentes dispositivos

³ <http://semantic-ui.com/>

Mediante clases de CSS concretas, se logra colocar los distintos elementos de HTML y adaptar el diseño de forma rápida, sencilla y sobre todo se obtiene un código muy legible, pues todas las clases representan una palabra del lenguaje común. A continuación se muestra un ejemplo de una ficha con información:

```
<div class="ui card">
  <div class="image">
    
  </div>
  <div class="content">
    <a class="header">Stevie Feliciano</a>
    <div class="meta">
      <span class="date">Joined in 2014</span>
    </div>
    <div class="description">
      Stevie Feliciano is a library scientist living in New York City.
      She likes to spend her time reading, running, and writing.
    </div>
  </div>
  <div class="extra content">
    <a>
      <i class="user icon"></i>
      22 Friends
    </a>
  </div>
</div>
```

El resultado sería el siguiente:

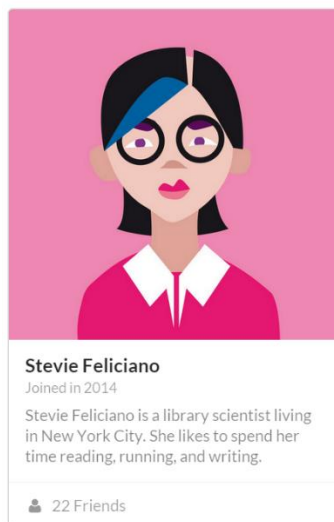


Ilustración 5: Ficha de ejemplo

De esta forma, se pueden crear componentes como botones, listas, tablas, imágenes, párrafos de texto, columnas y todo ello se adapta automáticamente gracias al framework.

3.10 Internacionalización

Tal y como se explicó con anterioridad, la accesibilidad es muy importante tanto para webs, como para aplicaciones móviles, y un factor importante es que usuarios o desarrolladores de otros países puedan acceder al contenido sin ninguna dificultad.

Es por eso que la web se ha traducido al inglés y existe la posibilidad de añadir más idiomas de forma sencilla. Además la base de datos también se ha diseñado pensando en la internacionalización, y todo el contenido está traducido, como veremos en capítulos posteriores.

3.11 Material Design

La nueva versión de Android, trae consigo muchas novedades, entre otras el nuevo diseño que se utiliza para las interfaces que se llama Material Design. Como en la usabilidad y accesibilidad, Google también tiene recomendaciones para esta nueva forma y en este proyecto, también se miden y analizan estas características.

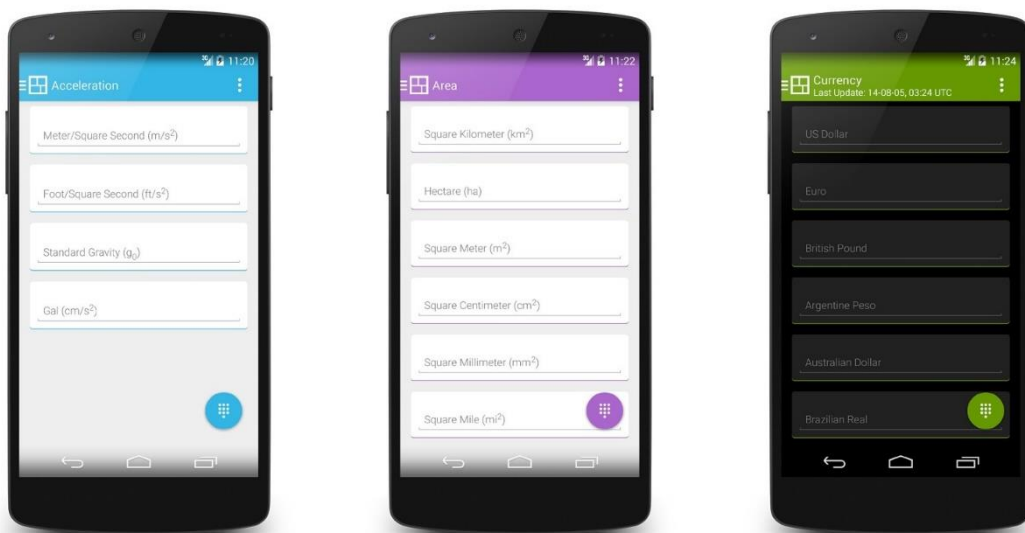


Ilustración 6: Aplicación con estilo Material Design

Capítulo 4. Planificación del Proyecto y resumen del Presupuesto

4.1 Planificación y lista de tareas

Esta es la lista de tareas inicial, por la que se ha seguido la planificación durante el transcurso del proyecto.

Nº	Tarea	Nivel	Descripción
1	Comienzo del proyecto	1	Decidir el proyecto que se quiere y pensarlo detenidamente.
2	Reunión con el director y subdirector, para establecer los principales objetivos	1	
3	Investigación previa	1	Comienza la etapa de investigación previa
4	Búsqueda de nueva tecnología	2	Se deben buscar otras herramientas parecidas o nuevas tecnologías que puedan ser de ayuda.
5	Análisis de la guía para desarrolladores de Google sobre aplicaciones Android	2	Google desarrollo una guía con conceptos que se pueden corregir a la hora de programar. Hay que leerla y explorarla con minuciosidad.
6	Leer la guía para la nueva versión de Android L	2	Android consta de una nueva versión con otra guía aparte que hay que leer y volver a estudiar.
7	Estudio del problema a resolver	2	Observar aquellos aspectos que se pueden mejorar e implementar para resolver los problemas planteados.
8	Elaborar un checklist con los aspectos principales que comprobar en la interfaz Android	2	Usando todo lo visto en la guía, definir un checklist con todos los aspectos que se puedan corregir de una interfaz Android.
9	Establecer prioridades y clasificar dichos aspectos en categorías para el posterior informe	2	
10	Herramientas a usar	1	
11	Valorar y escoger un IDE para desarrollar la herramienta web	2	Debe ser compatible con la versión y el lenguaje de programación que se vaya a escoger.
12	Estudio del IDE para desarrollar aplicaciones Android	2	
13	Determinar el servidor web a utilizar para servir la aplicación	2	Va relacionado también con el tipo de lenguaje y aplicación web que se desarrollará.

14	Estudiar el uso de posibles bases de datos	2	Hay que pensar si será necesaria una base de datos para la aplicación.
15	Determinar navegadores disponibles	2	La aplicación ha de ser compatible y verse igual en todos los navegadores posibles y que más se utilizan.
16	Determinar el lenguaje de programación que se usará	2	Teniendo en cuenta cómo se subirá y se procesará el fichero con la interfaz Android.
17	Estudio de viabilidad	1	Es necesario analizar la solución propuesta y determinar el alcance del proyecto para poder establecer claramente si es viable o no.
18	Analizar el sistema propuesto	2	
19	Estimar el tamaño del sistema	2	
20	Estudio de los posibles beneficios del proyecto y elaborar un informe	2	Hay que determinar qué ventajas aporta el proyecto y elaborar un informe con que beneficios podrían incorporarse y compararlo con herramientas de la misma categoría.
21	Reunión para debatir el estudio de viabilidad	1	
22	Análisis	1	Comienza la etapa de análisis
23	Definir ámbito del sistema: Funciones, Dimensiones, Restricciones, Usuarios	2	
24	Requisitos del Sistema	2	Identificar los requisitos que tendrá la aplicación.
25	Identificar las necesidades de los usuarios	2	Se ha de preguntar y explorar aquellos aspectos que son más requeridos por los usuarios, e identificar qué es lo que realmente quieren.
26	Identificación de los subsistemas	2	
27	Estimación detallada de costes, planificación y recursos	2	
28	Transformar requisitos en especificaciones funcionales	2	Una vez conocidos los requisitos, hay que desarrollar una tabla con las especificaciones funcionales de forma que queden registrados más técnicamente.
29	Construcción de un prototipo muy básico	2	Este prototipo prácticamente no tendrá funcionalidad y un diseño básico.
30	Tomar decisión de continuar con el proyecto o no	1	
31	Reunión para debatir el análisis hecho	1	
32	Diseño	1	Comienza la etapa de diseño

33	Localización de los paquetes software	2	Se deben agrupar distintos casos de uso en diferentes paquetes que engloben aspectos parecidos dentro de la aplicación.
34	Diagramas de clases, iteración, estados y actividades	2	
35	Prototipo más avanzado	2	Este prototipo ya tendrá algo de funcionalidad y diseño que el anterior.
36	Diseño y determinación de la base de datos	2	Se ha de determinar si es necesario una base de datos para la aplicación.
37	Especificación plan de pruebas	2	Elaborar un plan de pruebas adecuado y acorde a la aplicación.
38	Tomar decisión de continuar con el proyecto o no	1	
39	Reunión para debatir el diseño hecho y mostrar el prototipo	1	
40	Implementación	1	Comienza la etapa de implementación
41	Página principal	2	Codificación de la página principal, donde comenzarán los usuarios a usar la herramienta.
42	Menú y plantilla básica	2	Codificación del menú usando una plantilla básica para mostrar los diferentes elementos.
43	Formulario para subir el fichero XML	2	Codificación del formulario para subir el fichero. Contemplar aspectos de todo tipo como la seguridad.
44	Mejorar y completar el diseño y aspecto global de la página	2	Codificación más avanzada y casi definitiva del diseño y aspecto de la página en general. Criterios de usabilidad y accesibilidad serán tenidos en cuenta.
45	Base de datos (si existe)	2	Codificar la base de datos tal y como se diseño
46	Crear la base de datos	3	
47	Crear las tablas necesarias	3	
48	Introducir los datos	3	
49	Evaluador de la interfaz de la aplicación Android	2	Codificar el evaluador que se ejecuta al subir el fichero y que comprueba los diferentes criterios anteriormente establecidos que se estudiaron de la guía de desarrolladores
50	Comprobar aspectos de accesibilidad	2	Codificar criterios relacionados con la accesibilidad
51	Comprobar aspectos de usabilidad	2	Codificar criterios relacionados con la usabilidad
52	Comprobar otros aspectos presentes en el resumen	2	Codificar el resto de criterios u otros nuevos que puedan surgir

53	Elaborar la página de informe	2	Codificar la página que se mostrará al usuario una vez que termine de elaborarse el informe
54	Pruebas	1	Comienza la etapa de planificación de pruebas(calidad)
55	Pruebas de Interfaces y Contenidos	2	Codificar pruebas para determinar si la interfaz cumple con lo requerido
56	Pruebas de Funcionalidades	2	Codificar pruebas para determinar que la aplicación cumple con todas las funcionalidades.
57	Pruebas de Seguridad	2	Otro aspecto importante son las pruebas de seguridad.
58	Pruebas de Carga	2	Codifica pruebas donde se introduzca una interfaz compleja y comprobar los tiempos de ejecución.
59	Instalación	1	Comienza la fase de Instalación y despliegue
60	Instalar el servidor web definitivo donde se desplegará	2	
61	Desplegar la aplicación con la versión definitiva	2	
62	Elaborar documento de usuario	2	Se ha de elaborar un documento para explicar a los posibles usuarios cómo funciona la herramienta y cómo deben usarla
63	Elaborar una guía de mantenimiento	2	Se ha de elaborar un documento para explicar a los posibles desarrolladores futuros cómo está construida la herramienta y cómo
64	Finalizar documentación	1	Terminar de revisar toda la documentación y completar lo que falte
65	Creación de la presentación	1	
66	Detalles finales	1	Retoques finales para que todo esté listo para la presentación del proyecto.
67	Presentación del proyecto	1	
68	Cerrar el proyecto	1	Informar a todos los miembros de su finalización y reflexionar sobre el trabajo realizado así como la aceptación del mismo.

Se puede ver una planificación más detallada, y el seguimiento de la misma, en el anexo 1 de *este documento*.

4.2 Resumen del Presupuesto

Este presupuesto contempla la creación de una herramienta web, que se encargue de analizar interfaces Android, y que ofrezca un informe al usuario tal y como se ha comentado con anterioridad. Este presupuesto incluye, toda la administración del hosting, dominio y base de datos. Todo el análisis y el diseño del mismo así como el desarrollo también están incluidos. Otras características adicionales son; diseño responsive y que se adapte a todos los dispositivos, incluidos dispositivo móviles, posicionamiento en buscadores (para las palabras Android ui analyzer) y herramientas de análisis, en concreto, Google Analytics.

Este presupuesto no contempla en ningún caso el mantenimiento de la herramienta, salvo la administración del hosting y el dominio.

La metodología de trabajo será una entrega final, con revisiones periódicas para comprobar cómo marcha el proyecto.

Recursos	
Nombre	Precio
Análisis y diseño de la herramienta	19225€
Implementación y desarrollo	21000€
Instalación y configuración de la Base de datos	100€
Posicionamiento en buscadores	2000€
Herramientas de análisis (Google Analytics)	1600€
Servicio de Hosting y Dominio	775€
Diseño adaptable a todos los dispositivos	3000€
Otros datos	
Duración	180 días
Coste total	47700€

Se puede ver un presupuesto más detallado y su seguimiento en el Anexo 2 de este documento.

Capítulo 5. Análisis

5.1 Alcance del sistema

Antes de continuar con este capítulo y los siguientes, se recomienda la lectura del apartado “Aspectos teóricos” situado en el capítulo 3.

Los principales objetivos del sistema son:

- Una herramienta disponible a través de una página web que sea *responsive* para poder analizar una interfaz Android subiendo un fichero XML
- Direcciones SEO amigables usando reescritura de URL
- Generar un informe con los distintos fallos y aciertos
- Disponer de funcionalidades adicionales, como son:
 1. Filtros para ver solo los fallos o los aciertos también
 2. Posibilidad de guardar el informe en PDF
 3. Posibilidad de traducir el informe o la web entera
 4. Barra de estado, que indique el nivel de progreso hasta completar y solucionar todos los fallos detectados
 5. Disponer de un enlace para cada fallo o acierto, que enlace con la guía de recomendaciones de Google, o con otra web que lo explique y detalle.
- Implementación de una página de contacto para que los usuarios puedan escribir y resolver cualquier duda
- Todo el proceso de envío tanto de ficheros como de información deberá ser seguro con validaciones tanto en el cliente como el servidor
- Uso de herramientas de análisis como Google Analytics para realizar un seguimiento de los usuarios. Debido a esto, serán necesarios avisos legales y política de privacidad y de cookies.

La aplicación será mayoritariamente utilizada por desarrolladores Android que busquen en Internet una herramienta con estas características, ya que estará posicionada para las palabras clave “Android analyzer interfaces”.

Una vez dentro el usuario sólo tendrá que entrar en la web, y seleccionar el fichero que contiene la interfaz, y pulsar en analizar. No será necesario instalar ningún otro programa, ni importará el sistema operativo que tenga instalado o el navegador que utilice. Se mantiene la compatibilidad con los principales navegadores como son, Chrome, Firefox, Internet Explorer, Opera o Safari.

5.1.1 Descripción de los actores

En esta aplicación no hay diferencia de usuarios. Todos serán del mismo tipo puesto que no se identifican de ninguna manera. Por lo tanto, cualquier usuario de Internet, ya sea desarrollador o no, podrá hacer uso de la aplicación y analizar una interfaz Android.

5.2 Identificación de los Subsistemas

El sistema se compone principalmente de un subsistema de informes. Este subsistema engloba todo aquello que tiene que ver con lo necesario para la generación de los informes. Así pues, subir un fichero XML, analizarlo, o guardar el informe será parte de su contenido. A continuación se muestra un diagrama de paquetes general, dónde se puede ver la estructura de todo el sistema. Como se verá más adelante, se ha usado el patrón MVC:

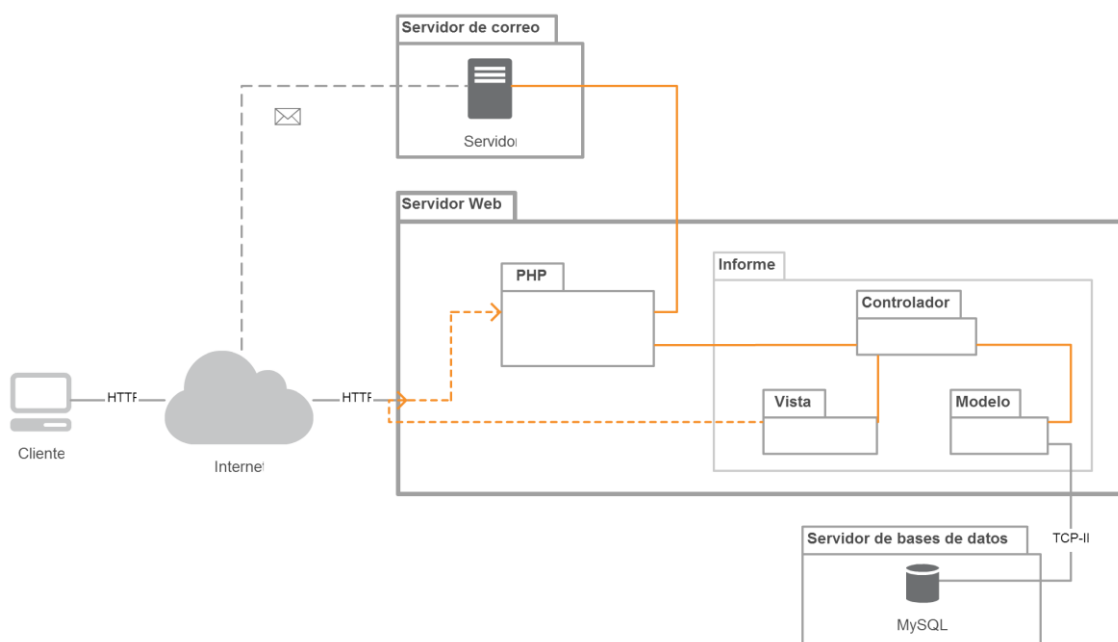


Ilustración 7: Diagrama de general de subsistemas

5.3 Diagramas de casos de uso

5.3.1 Subsistema del informe

5.3.1.1 Diagrama de casos de uso

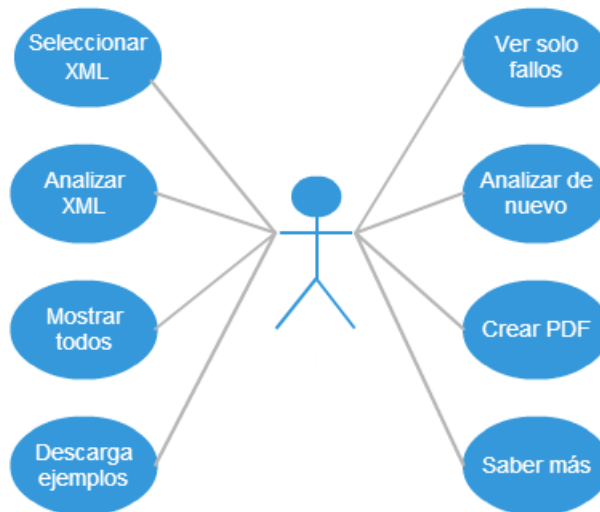


Ilustración 8: Casos de uso del subsistema Informe

5.3.1.2 Detalle de los casos de uso

Seleccionar XML	
Precondiciones	Ninguna
Poscondiciones	El fichero seleccionado pasa a ser el que el usuario haya escogido
Actores	Usuario
Descripción	El usuario pulsará el botón de seleccionar XML, y escogerá un fichero mediante una ventana de selección. El fichero que será subido posteriormente será este mismo.
Excepciones	Ocurrirá un error si el fichero no existe o no se encuentra la ruta hacia él. También si el nombre no es correcto o contiene caracteres no permitidos.

Analizar XML	
Precondiciones	Tener seleccionado un fichero XML
Poscondiciones	Se generará el informe acorde al fichero XML
Actores	Usuario
Descripción	El usuario pulsará el botón de analizar XML y acto seguido aparecerá el informe con los posibles fallos o correcciones y todas las demás opciones disponibles
Excepciones	Ocurrirá un error si el fichero no tiene formato XML o si tiene un tamaño mayor de 5mb.

Mostrar todos	
Precondiciones	Solo se deben estar mostrando los fallos
Poscondiciones	En el informe aparecerán todas las comprobaciones ya sean fallos o aciertos. El botón de Ver solo fallos será ocultado si estaba visible.
Actores	Usuario
Descripción	El usuario pulsará el botón de Mostrar todos situado al comienzo del informe y a continuación podrá consultar los fallos o los aciertos que cumplen la interfaz que ha analizado
Excepciones	Ninguna

Ver solo fallos	
Precondiciones	Se deben estar mostrando tanto aciertos como fallos
Poscondiciones	En el informe aparecerán solo los fallos o advertencias que tenga la interfaz. El botón de Mostrar todos será ocultado si estaba visible.
Actores	Usuario
Descripción	El usuario pulsará el botón de Ver solo fallos situado al comienzo del informe y a continuación podrá consultar sólo los fallos centrándose en aquello que realmente importa y tiene que mejorar
Excepciones	Ninguna

Analizar de nuevo	
Precondiciones	El usuario debe haber analizado con anterioridad un fichero XML y encontrarse en la página de informes
Poscondiciones	Se cargará la página de inicio de nuevo y se deseleccionará el fichero XML.
Actores	Usuario
Descripción	El usuario pulsará el botón de Analizar de nuevo situado al comienzo del informe y se le redirigirá a la página de inicio para que pueda seleccionar de nuevo el fichero XML.
Excepciones	Ninguna

Crear PDF	
Precondiciones	El usuario debe haber analizado con anterioridad un fichero XML y encontrarse en la página de informes
Poscondiciones	Se abrirá una nueva página con el informe en PDF
Actores	Usuario
Descripción	El usuario pulsará el botón de Crear PDF situado al comienzo del informe y se abrirá una nueva página que contendrá el informe completo en PDF. Esta página permite guardarlo en local, aumentarlo o reducirlo.
Excepciones	Ninguna

Saber más	
Precondiciones	El usuario debe haber analizado con anterioridad un fichero XML y encontrarse en la página de informes
Poscondiciones	Se abrirá una nueva página con información relacionada al error o

	acierto asociado
Actores	Usuario
Descripción	Cuando el usuario tenga dudas de un cierto fallo o quiera saber más sobre un acierto podrá hacer clic en el enlace situado a la derecha en cada fila. Este enlace contiene información relacionada que le ayudará a entender mejor lo que se está analizando y cómo solucionarlo en caso de que sea un error.
Excepciones	Ninguna

<u>Descargar ejemplos</u>	
Precondiciones	El usuario debe encontrarse en la página de inicio
Poscondiciones	Se descargará a su carpeta de descargas, un archivo comprimido con varios ejemplos que puede probar
Actores	Usuario
Descripción	Algunos usuarios pueden desear probar la herramienta para obtener una visión inicial de lo que hace y para qué sirve. En la página de inicio, hay un enlace que cuando el usuario lo pulsa, descarga en un archivo ZIP, varios ejemplos de interfaces Android con típicos fallos.
Excepciones	Ninguna

5.4 Diagrama de Clases

5.4.1 Modelo de clases del sistema

A continuación se detalla el modelo en el que se reflejan las principales clases del sistema:

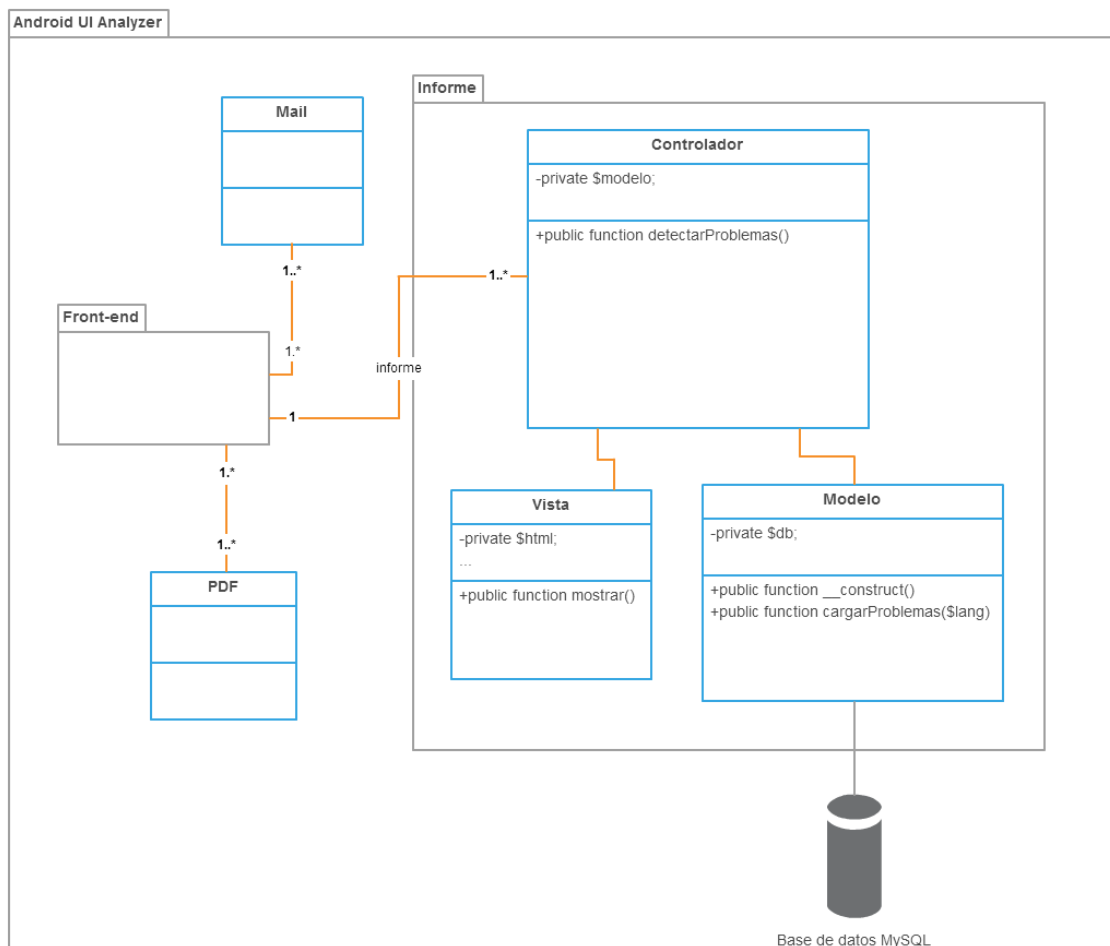


Ilustración 9: Diagrama de clases del análisis

Este diagrama contiene las principales clases del sistema que conformarán el núcleo del mismo. Por una parte, todos los fallos que se comprueban en el informe, los analizará la clase del Controlador a través de diferentes métodos para cada uno. Además, esta clase se podrá apoyar en otra clase adicional, que contenga métodos genéricos.

Para que se pueda procesar el fichero XML, primero se subirá dicho fichero, a través de la página de inicio. Todas las páginas están agrupadas en el paquete Front-end. Entonces el controlador llama al modelo y cargará de la base de datos la información de cada fallo o acierto.

Finalmente la vista se encargará de mostrar todo el informe y las opciones disponibles.

Por otro lado, existirán clases para la generación del PDF y el envío de correo para el formulario de contacto. Estas clases serán escogidas entre las más usadas en la comunidad de desarrolladores y así evitar rehacer código.

5.5 Análisis de Interfaces de Usuario

La interfaz general de la web, seguirá un patrón en común, ya que el framework de CSS lo permite y de esta forma todas las páginas mantienen un estilo muy parecido. Este diseño será orientado al usuario, y pensado para que se relacione con las interfaces de Android.

En todas las pantallas o interfaces, se han incluido los principales aspectos de usabilidad y accesibilidad como por ejemplo:

- La opción del menú superior siempre indicará en que página está situado el usuario.
- Los idiomas serán definidos con sus siglas identificativas dependiendo del país.
- Son interfaces claras sin ningún tipo de elemento que pueda despistar o producir una sensación desagradable al usuario.
- En ningún momento ocurrirá ninguna acción que el usuario no se pueda esperar.

5.5.1 Mapa de navegación

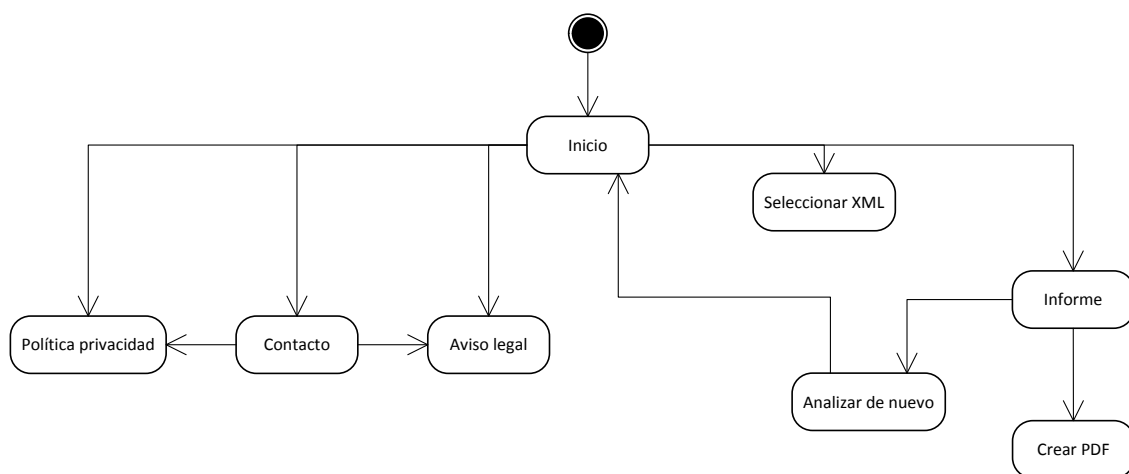


Ilustración 10: Mapa de navegación

5.5.2 Descripción de las Interfaces

Todas las imágenes que se muestran a continuación son versiones preliminares, y se han desarrollado como prototipos para su adaptación final.

5.5.2.1 Inicio

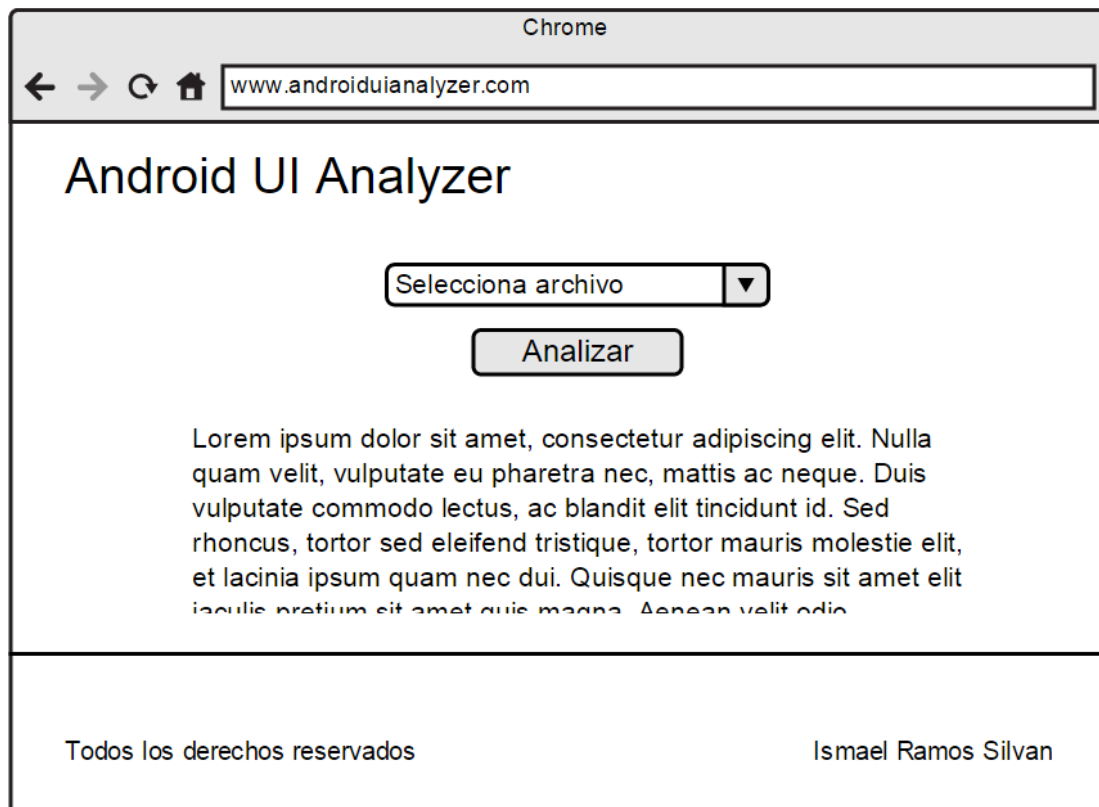


Ilustración 11: Prototipo de la página de inicio

Esta es la página principal, desde la que se podrán comenzar a subir ficheros XML y ver el informe, descargar los ejemplos, traducir a los idiomas, o navegar hacia otras páginas como contacto, aviso legal o la política de privacidad.

Si el usuario elige subir un fichero XML, se le mostrará un cuadro de dialogo como por ejemplo:



Ilustración 12: Prototipo de seleccionar archivo

5.5.2.2 Informe

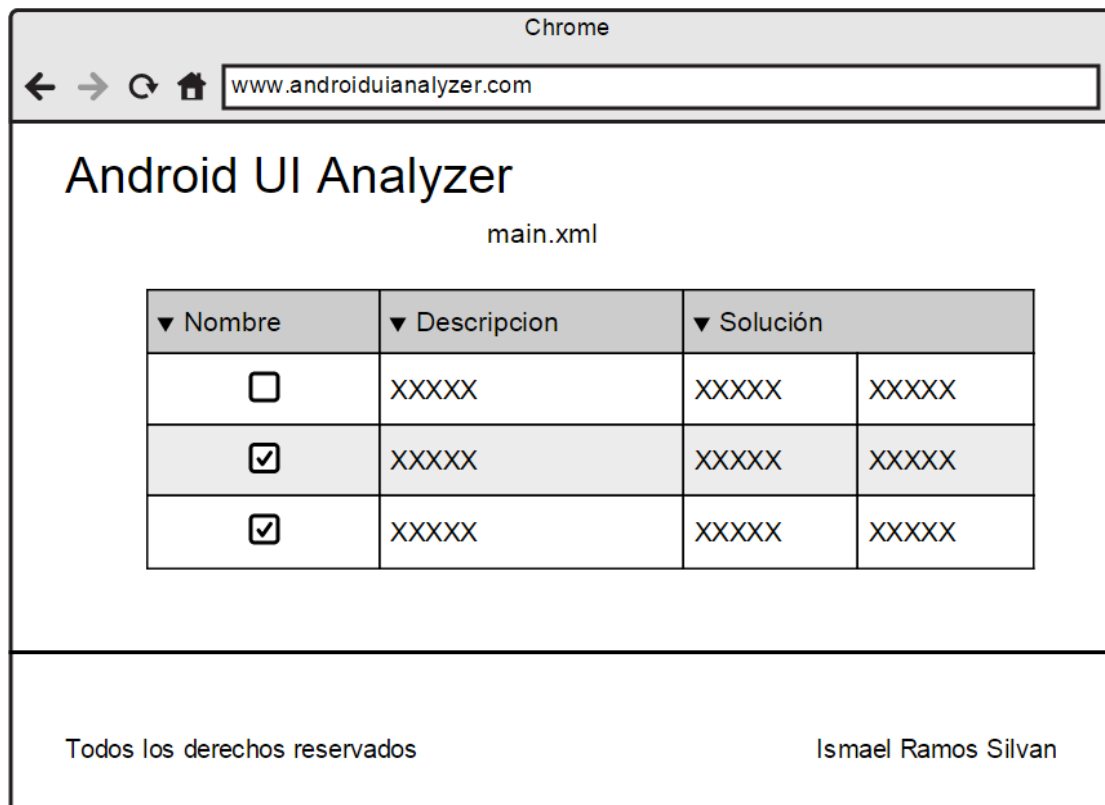


Ilustración 13: Prototipo del informe

Esta página será la encargada de mostrar todos los errores encontrados, dependiendo de la interfaz que se haya analizado. En la parte superior, se añadirán las diferentes opciones como “Mostrar todos” o “Mostrar solo fallos” que permitirá al usuario intercambiar o filtrar lo que desea ver. También se situará una opción para “Analizar de nuevo” y que vuelva a analizar el fichero desde la página de inicio, además de “Crear PDF” cuya página se muestra a continuación:

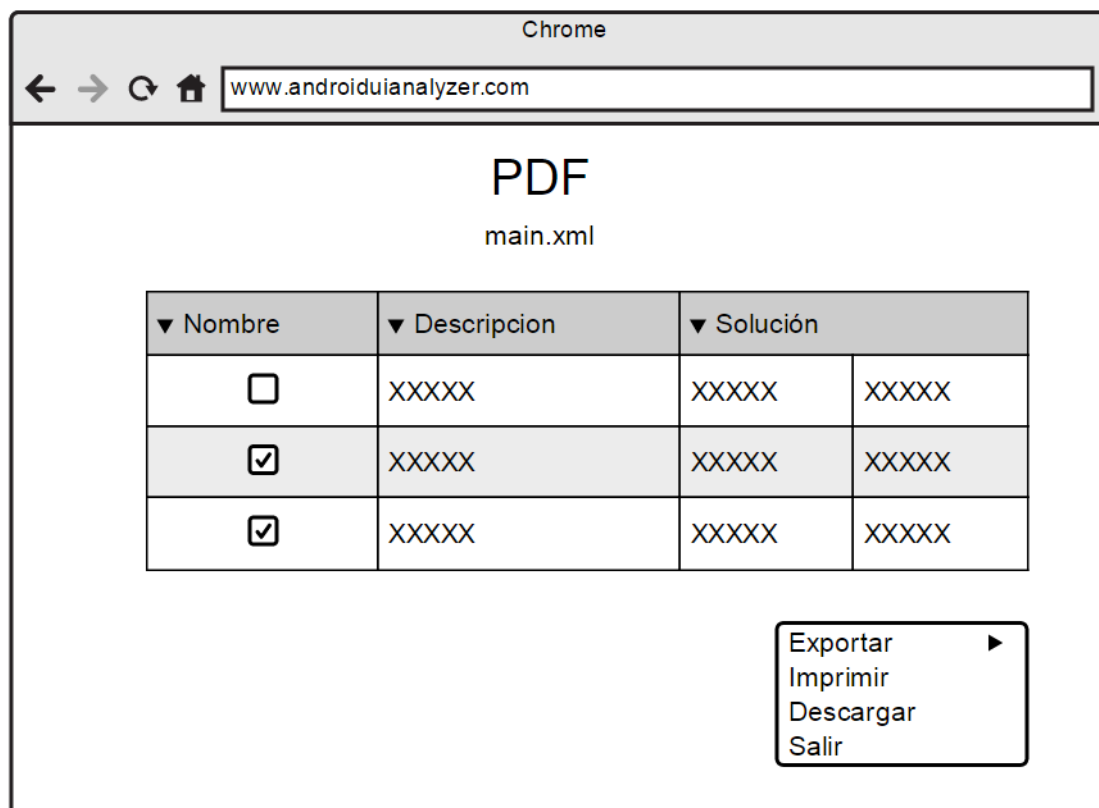
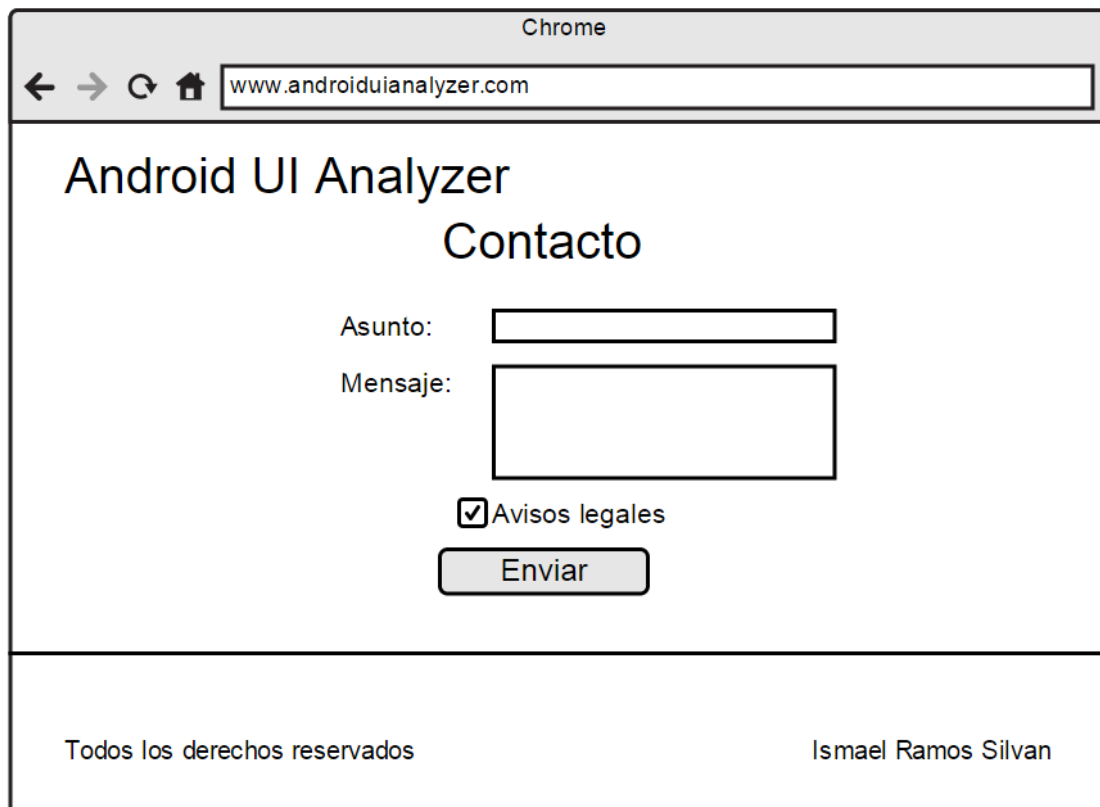


Ilustración 14: Prototipo del PDF

En cada elemento del informe se implementará un enlace para “Saber más” el cual abrirá la página de la guía de Google correspondiente a ese elemento que se está analizando. Esto permite al usuario tener más detalles y revisar esa parte en concreto.

Por otra parte, al final de la página de informes, existirá una barra de progreso, para informar al usuario de cuántos fallos ha tenido y de lo que le falta para tenerlo todo correcto y acorde a las buenas prácticas.

5.5.2.3 Contacto



The image shows a web browser window titled "Chrome" with the address bar containing "www.androiduianalyzer.com". The main content area displays the title "Android UI Analyzer" and a sub-section "Contacto". Below this, there is a form with two input fields: "Asunto:" and "Mensaje:". Below the "Mensaje:" field is a checkbox labeled "Avisos legales" which is checked. At the bottom of the form is a button labeled "Enviar". The footer of the page contains the text "Todos los derechos reservados" on the left and "Ismael Ramos Silvan" on the right.

Ilustración 15: Prototipo de la página de contacto

Esta pantalla contendrá un formulario para que lo rellene el usuario, y pueda ponerse en contacto si tiene alguna duda o sugerencia.

Cuando el usuario realiza todos los pasos y pulsa "Enviar", aparecerá un mensaje avisándole de si ha ocurrido algún error o finalmente se envió el mensaje. Para evitar recibir spam se hará uso de un Captcha.

Al igual que en los informes, también se informará al usuario de lo que le falta para completar el formulario mediante una barra de progreso.

5.5.2.4 Avisos legales

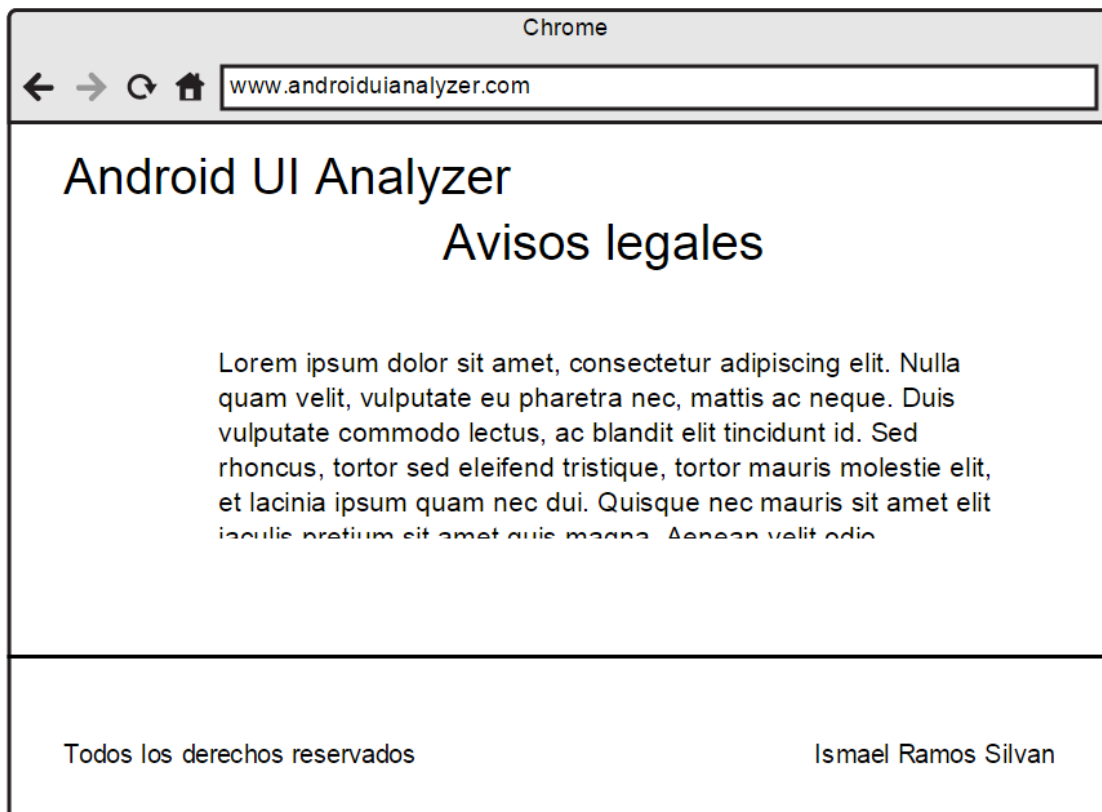


Ilustración 16: Prototipo de los avisos legales

Se tratará de la página que informa al usuario de los distintos avisos legales que debe saber si hace uso de la web y de sus servicios.

5.6 Análisis del Plan de Pruebas

Para el plan de pruebas se realizarán de distintos tipos los cuales se explican a continuación.

5.6.1 Pruebas funcionales

Estas pruebas consistirán en la ejecución y revisión de distintos resultados para distintas entradas a través de pruebas de caja negra. Serán las encargadas de comprobar las funcionalidades de la aplicación, y de verificar que hace lo que debe en cada situación. Por lo tanto, validarán que la aplicación cumple con los requisitos establecidos y los objetivos principales.

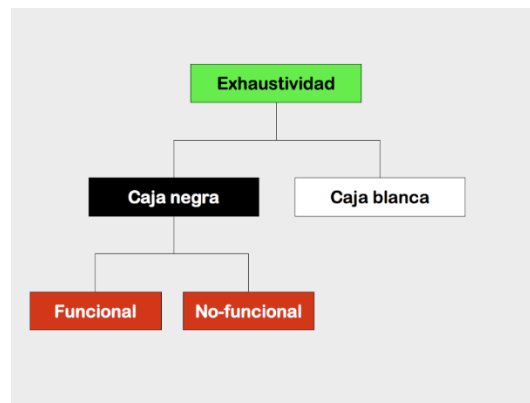


Ilustración 17: Distintos tipos de pruebas

5.6.2 Pruebas de rendimiento

Este tipo de pruebas se realizarán con herramientas que midan el tiempo de carga de la web. Servirán también para verificar que se cumplen ciertos objetivos relacionados con la eficiencia de la web y que fueron establecidos al comienzo.

5.6.3 Pruebas de Usabilidad

Entre estas pruebas se tomarán un grupo de distintos tipos de usuario que sean o no desarrolladores de Android y se les propondrán una serie de tareas sobre la herramienta. Tras esto, se elaborará un cuestionario orientado a analizar posibles mejoras y recibir información adicional del sistema y nuevas opiniones y sugerencias de estos usuarios. La versión de la aplicación con la que realizarán el cuestionario ya será un versión que disponga de los elementos básicos además del informe.

Posteriormente cuando se analicen los cuestionarios, se recopilarán las distintas respuestas y se revisarán aquellos criterios que puedan mejorarse o se implementarán mejoras si son estrictamente necesarias y demandadas.



5.6.4 Pruebas de accesibilidad

Al igual que para la usabilidad, también serán importantes las pruebas de accesibilidad. Para ello se hará uso de distintos navegadores, de herramientas que automaticen el proceso así como validadores. Todo será detallado en capítulos posteriores.

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

6.1.1 Entorno tecnológico de implantación

6.1.1.1 Equipo hardware

Para poder hacer uso de la aplicación no hace falta ningún requisito especial, sino que cualquier equipo informático que sea capaz de ejecutar un navegador será capaz de usar esta herramienta.

Para ello, lo que si es necesario es una conexión a Internet, para que el usuario pueda entrar y hacer uso de la herramienta web.

6.1.2 Dominio

Para que la web sea accesible y se indexe en los principales buscadores, se ha elegido un dominio y desplegado en la siguiente URL:

www.androiduianalyzer.com

Esta URL también ha sido escogida pensando en el posicionamiento futuro de la página, pues cada palabra tiene gran importancia para el algoritmo de los principales buscadores.

6.1.3 Diagrama de clases

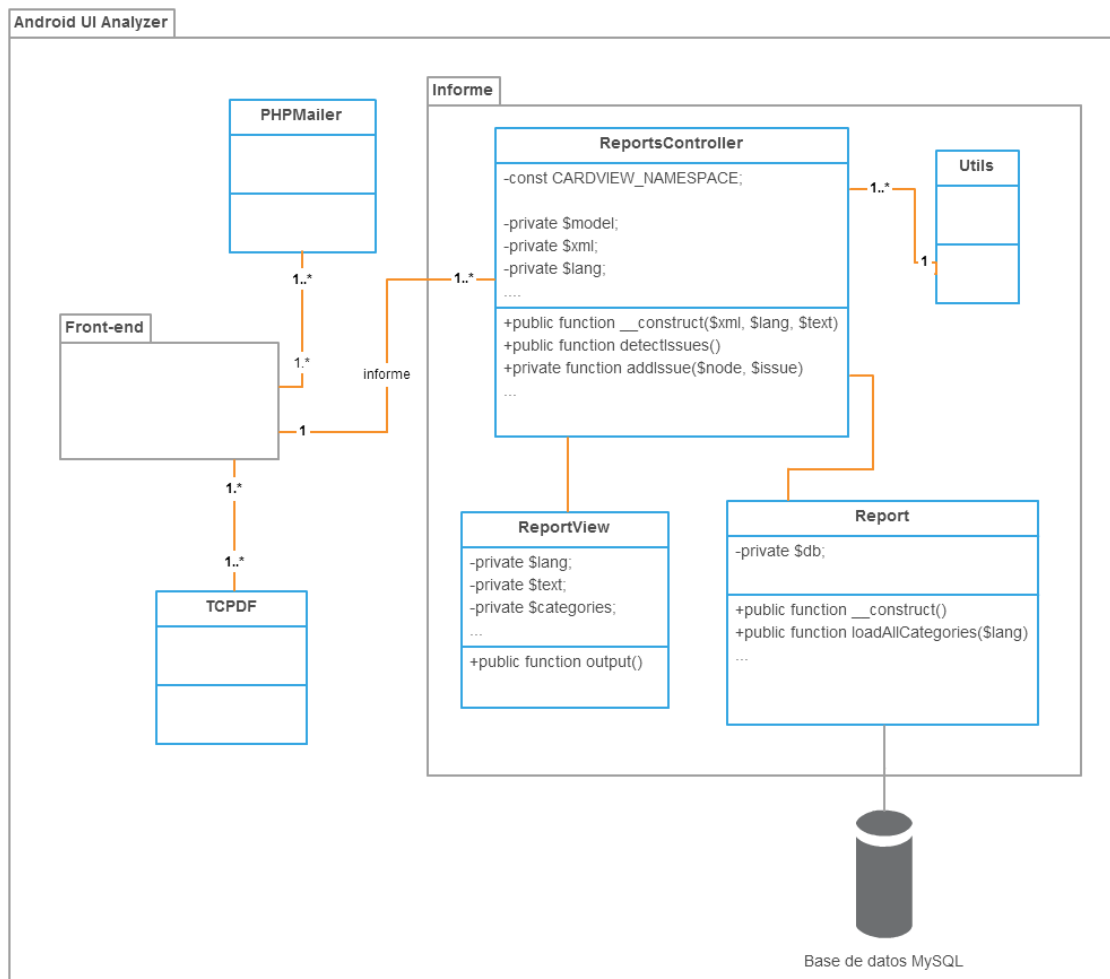


Ilustración 18: Diagrama de clases del diseño

6.1.4 Patrones de diseño implementados

Para brindar una solución a problemas de software que ya han ido surgiendo con el paso de los años, se utilizan patrones de diseño. Estos patrones permiten resolver problemas de una forma segura y que ya ha sido probada por el resto de desarrolladores de software. Esto es así, debido a que la solución que brindan a un determinado problema posee ciertas características que se deben cumplir y que aseguran que el problema no se repetirá y quedará solucionado. Con este motivo se ha implementado el patrón de diseño MVC, que se explica a continuación.

6.1.4.1 MVC

Se trata de un patrón archiconocido que trata de separar los datos y la lógica de negocio de una aplicación, de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. El MVC está formado por tres componentes distintos que son el modelo, la vista y el controlador.”

Un buen ejemplo del mundo real podría ser un coche. En él, existen dos vistas, la interior y la exterior. El conductor sería el controlador que es el que maneja todo, y los distintos componentes del coche como los frenos, ruedas o puertas serían el Modelo.

La implantación en este proyecto se ha hecho con la siguiente organización y relación de clases. Por un lado, está la clase “Report” que sería el modelo. Esta clase además es la encargada de establecer la conexión con la base de datos e interactuar con ella, mediante objetos PDO.

El controlador sería “ReportsController”, y es el encargado de llamar al modelo cuando sea necesario, y ejecutar los métodos que analizan el XML de la interfaz Android.

Por último, la vista se encuentra en la clase “ReportView” que contiene el código HTML y PHP necesarios para mostrar el informe.

6.1.5 Diagrama de Despliegue

La página web y todo el código desarrollado, ha sido desplegado en un servidor web profesional bajo un dominio específico. Además se ha contratado un hosting para almacenar todo el contenido en dicho servidor web. Otra ventaja es disponer de un servidor de correo y de bases de datos.

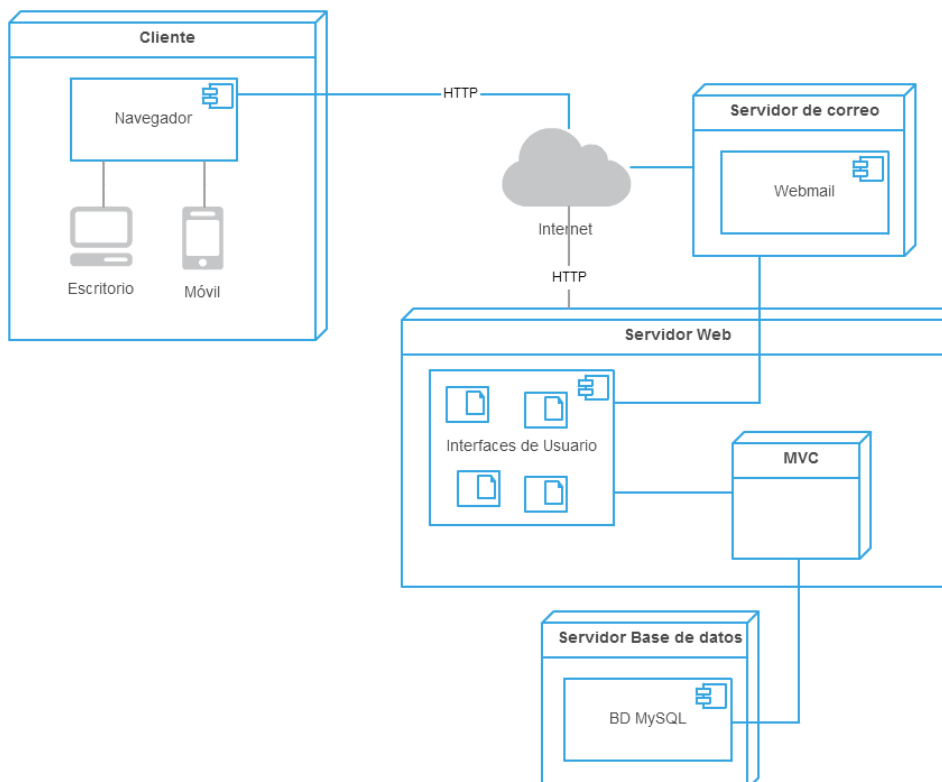


Ilustración 19: Diagrama de despliegue

6.1.5.1 *Servidor Web*

El servidor web contratado es una máquina Linux, con soporte para PHP en su versión 5. Actualmente los límites de transferencia de datos y de almacenamiento en disco, son suficientemente altos y se puede asegurar que no habrá ningún problema con ellos.

6.1.5.2 *Servidor de Base de datos*

Debido a que ha sido necesaria el uso de una base de datos, el servicio de hosting también proporciona soporte para ello. Existe por lo tanto un servidor de base de datos al que se conecta el servidor web. La base de datos es MySQL con una versión 5.1. El límite de bases de datos también es suficiente para este proyecto y no supondrá problema alguno.

6.1.5.3 *Servidor de correo*

Para que el usuario pueda ponerse en contacto y al rellenar el formulario se envía un email es necesario disponer de un servidor de correo. Esto también nos lo facilita el hosting y además nos proporciona un dominio de correo (info@androiduianalyzer.com).

6.1.6 **Guías de diseño comunes**

El proyecto contiene una serie de carpetas para las cuales se ha buscado un nombre descriptivo. Todo el código desarrollado está escrito en inglés y estos nombres también. A continuación se listan una serie de elementos que son comunes para todas las páginas, y que tienen que ver con el código implementado y su estructura:

- Se han creado páginas para los errores 404 y 500 llamadas de esa forma.
- El fichero robots.txt excluye las páginas de error y la del informe. Por otro lado establece la ruta para el sitemap.
- Se ha generado un sitemap.xml con la distribución de las páginas para los buscadores.
- La carpeta “js” contiene los ficheros javascript que utiliza la aplicación.
- Para no repetir código, se han creado ficheros aparte como por ejemplo “create-pdf.php” que se sitúan todos en la carpeta “includes”. Esta carpeta también contiene la cabecera, el pie, el menú o los scripts que se incluyen para todas las páginas. De esta manera se puede modificar desde un solo fichero todas las páginas a la vez, sin hacer cada cambio en cada página.
- La carpeta “examples” contiene ejemplos de interfaces Android. Esta es la carpeta que se recorre y se comprime cuando el usuario descarga los ejemplos.
- Las carpetas “css” e “img” contienen los estilos y las imágenes necesarios para el procesamiento de la página.
- El fichero .htaccess contiene directivas para elegir las páginas de error, restringir el acceso a la raíz del proyecto y de la propia manipulación de sí mismo. También se activa la compresión mediante gZip y el módulo deflate así como los periodos en los que se activa la cache para cada tipo de fichero (html, js, css, etc.).

Por último, establece una serie de condiciones de reescritura para proteger la web frente a bots conocidos y catalogados como malignos.

6.1.7 Reescritura de URLs

Para que la aplicación web se indexe correctamente y para cumplir con las buenas prácticas de URLs amigables, se ha hecho uso del módulo “mod_rewrite” que proporciona Apache. De esta forma se cambian las URLs para que sigan la forma:

<http://www.androiduianalyzer.com/idioma/página>

Así pues un ejemplo válido sería:

<http://www.androiduianalyzer.com/es/home>

Por otro lado, el uso de variables GET en las URL supone un gran fallo y algo catalogado como mala práctica tal y como Google lo ha explicado y detallado⁴. Esto ocurre cuando el usuario traduce una página, ya que el idioma se transmite a través de variables GET. Toda la reescritura se realiza a través del fichero .htaccess, con las directivas:

```
#Redirección de non-www/www
RewriteCond %{HTTP_HOST} !^www\. [NC]
RewriteRule ^(.*)$ http://www.%{HTTP_HOST}/$1 [R=301,L]

#Redirección /example?lang=language to /language/example
RewriteCond %{THE_REQUEST} /+([^\?]+)?(?:\.php)?\?lang=([^&]+) [NC]
RewriteRule ^/%2/%1? [R=301,L,NE]

#Remove extensions
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME}.php -f
RewriteRule ^(.+)?/?$ $1.php [L]

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME}.html -f
RewriteRule ^(.+)?/?$ $1.html [L]

#Redirect /language/example to /example?lang=language
RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule . - [L]

RewriteRule ^([^\?]+)/([^\?]+) $2?lang=$1 [L,QSA]
```

⁴ <https://support.google.com/webmasters/answer/182192>

6.2 Fallos analizados

Como complemento a lo que se verá en el diseño detallado del sistema, esta sección resume todos los fallos que han sido analizados en el informe generado. La siguiente lista muestra el nombre del problema, una descripción, la solución a dicho problema y si fuera oportuno un ejemplo de código para su explicación. Todas las soluciones que se verán a continuación, están en la guía de Android Developers.

6.2.1 Formato incorrecto

Para que la herramienta pueda analizar sin problemas la interfaz XML, es necesario que el fichero que haya subido el usuario tenga formato XML y de verdad sea una interfaz Android.

La herramienta analiza mediante la librería SimpleXML dicho formato, y también mira el namespace que se usa para ver si coincide con el de Android.

Formatos incorrectos:

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools" >

  <Button
    android:id="@+id/button1"
    anue" />

</RelativeLayout>
```

6.2.2 Uso de "wrap_content" y "match_parent"

Cuando el desarrollador establece como ancho o alto de algún componente algo diferente a estas dos opciones, la interfaz no se adaptará correctamente a distintos tamaños de pantalla.

En vez de usar valores hardcoded, es una buena práctica usar "wrap_content" y "match_parent" y permitir que los componentes sean más grandes o pequeños dependiendo de la pantalla.

Mala práctica:

```
<Button android:id="@+id/categorybutton"
  android:background="@drawable/button_bg"
  android:layout_height="match_parent"
  android:layout_weight="0"
  android:layout_width="120dp"
  style="@style/CategoryButtonStyle"/>
```

Más información en:

<https://developer.android.com/training/multiscreen/screensizes.html#TaskUseWrapMatchPar>

6.2.3 Usa RelativeLayout

Muchos desarrolladores hacen uso de LinearLayout para situar mejor los componentes, anidando varios incluso si es necesario. Este es un mal uso y es preferible usar siempre que se pueda RelativeLayout tal y como recomienda Google.

Mala práctica:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Name:"
        />
    </LinearLayout>
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Address:"
        />
    </LinearLayout>
    <Button android:id="@+id/save"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Save"/>
</LinearLayout>
```

Más información:

<https://developer.android.com/training/multiscreen/screensizes.html#TaskUseRelativeLayout>

6.2.4 Utiliza píxeles independientes de la densidad

Este fallo puede tener un símil en el desarrollo web como puede ser usar píxeles en vez de unidades relativas como em o tantos por ciento. En Android, tampoco es recomendable usar la unidad px para nada. En su lugar se recomienda usar dp y sp.

Mala práctica:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="12px"
/>
```

Más información:

<https://developer.android.com/training/multiscreen/screendensities.html#TaskUseDP>

6.2.5 Añade descripciones de contenido

Un fallo muy importante en cuanto a la accesibilidad, es no rellenar el atributo `contentDescription`. Este atributo aparece en pantalla, cuando un usuario activa las opciones de accesibilidad. Por eso es muy importante que no esté vacío, para aquellos usuarios con alguna deficiencia visual.

Mala práctica:

```
<ImageButton
  android:id="@+id/add_note_button"
  android:src="@drawable/add_note"/>
```

Más información:

<https://developer.android.com/guide/topics/ui/accessibility/apps.html#label-ui>

6.2.6 Diseño de navegación direccional

Hay que tener en cuenta la gran cantidad de dispositivos y no todos son iguales. Algunos tienen diferentes mecanismos para interactuar con las interfaces, como joysticks, D-Pad o teclas. Para que todos funcionen correctamente y en todos los casos, es recomendable definir el atributo `focusable` y otros como `nextFocusDown`, `nextFocusUp`, `nextFocusRight`, y `nextFocusLeft`.

Más información:

<https://developer.android.com/training/accessibility/accessible-app.html#focus>

6.2.7 Separar el diseño del contenido

Cuando una aplicación usa distintos estilos y se entremezclan, produce un efecto de rechazo en el usuario y no es bueno para la usabilidad de la misma.

Una práctica recomendada para los desarrolladores, es agrupar los estilos de la aplicación. Esto permite tener controlados todos desde un solo sitio.

Mala práctica:

```
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:textColor="#00FF00"
  android:typeface="monospace"
  android:text="@string/hello" />
```

Más información:

<https://developer.android.com/guide/topics/ui/themes.html>

6.2.8 Usa compound drawables

Puede ser bastante común que se dé la siguiente situación; tener un elemento LinearLayout que contiene un ImageView y un TextView. Esta situación se puede hacer más eficiente usando el método `setCompoundDrawableWithIntrinsicBounds()` y sustituyendo el layout por un simple TextView. Al mejorar la eficiencia, se mejora de algún modo la usabilidad, pues la interfaz tarda menos en cargar y esto también es algo que hay que tener presente.

Más información:

[http://developer.android.com/reference/android/widget/TextView.html#setCompoundDrawablesWithIntrinsicBounds\(int, int, int, int\)](http://developer.android.com/reference/android/widget/TextView.html#setCompoundDrawablesWithIntrinsicBounds(int, int, int, int))

6.2.9 Optimizar mezclando

Otra mala práctica bastante común, es usar el siguiente código o uno análogo:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"

        android:scaleType="center"
        android:src="@drawable/golden_gate" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dip"
        android:layout_gravity="center_horizontal|bottom"

        android:padding="12dip"

        android:background="#AA000000"
        android:textColor="#ffffffff"

        android:text="Golden Gate" />

</FrameLayout>
```

Que da como resultado:



Tal y como se explica en el enlace, esto se puede simplificar y mejorar el rendimiento de forma notable por:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"

        android:scaleType="center"
        android:src="@drawable/golden_gate" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="20dip"
        android:layout_gravity="center_horizontal|bottom"

        android:padding="12dip"

        android:background="#AA000000"
        android:textColor="#ffffffff"

        android:text="Golden Gate" />

</FrameLayout>
```

La explicación se puede consultar en el siguiente enlace:

<http://android-developers.blogspot.com.es/2009/03/android-layout-tricks-3-optimize-by.html>

6.2.10 Crear listas y tarjetas

Las nuevas versiones de Android también traen nuevos componentes debido a los requisitos que van exigiendo los usuarios. Android se está adaptando a lo que ellos necesitan y también orienta sus componentes para que los desarrolladores trabajen menos.

Para crear listas o tarjetas disponemos de dos nuevos componentes como son, RecyclerView que soporta diferentes tipos de diseño y ofrece mejoras de rendimiento, además de CardView que permite mostrar información dentro de tarjetas con un aspecto coherente y ordenado.

La herramienta detecta si se utilizan componentes antiguos (ListView) y propone que se cambien por los nuevos.

Más información:

<http://developer.android.com/training/material/get-started.html#ListsCards>

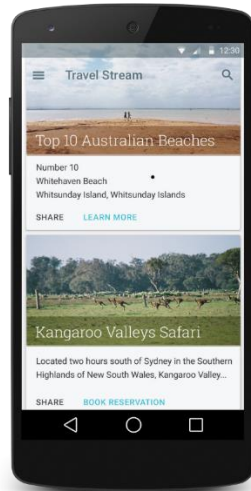
6.2.11 Crear fichas con estilo de Material Design

La nueva versión de Android, trae un nuevo lenguaje de diseño como es Material Design. Además, las aplicaciones que hacen uso de Fichas (Cards) reciben muy buenas críticas por parte de los usuarios, ya que la información queda muy bien organizada.

Además de su uso, Google ofrece a los desarrolladores unas recomendaciones para definir su estilo correctamente. Estas son:

- Definir el atributo `cardCornerRadius`. Por ejemplo: `card_view:cardCornerRadius="4dp"`
- Usar el atributo `card_view:cardBackgroundColor`

Si esto se cumple, el desarrollador podría fácilmente generar lo siguiente:



6.2.12 Tamaño de letra

Una de los fallos “imperdonables” para un desarrollador, es usar un tamaño de texto demasiado pequeño, y que los usuarios no puedan leer sin esfuerzo los textos de la aplicación Android.

La herramienta comprueba que todos estén por encima de un valor mínimo (11) e informa en caso contrario.

Mala práctica:

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="12sp"
/>
```

Más información:

http://developer.android.com/reference/android/widget/TextView.html#attr_android:textSize

6.2.13 Cadenas literales

Este problema se trata de una mala práctica y un inconveniente para una aplicación internacionalizada, lo que se traduce en una mala accesibilidad. Básicamente, se trata de no usar nunca valores literales para el texto.

Por ejemplo:

```
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="Hola" />
```

En su lugar se debería usar:

```
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="@string/hello" />
```

Se puede consultar más información en:

<http://developer.android.com/guide/topics/resources/string-resource.html>

6.2.14 El texto de los botones deberían ser verbos

Por motivos de usabilidad, los usuarios generalmente entienden mejor cuando un botón es igual a una acción y por lo tanto hay que usar verbos, que cuando no ocurre esto.

Por ese motivo, la herramienta avisa al desarrollador, para que compruebe que esto ocurre así. Sería demasiado complejo analizar esto programáticamente.

Más información:

<http://developer.android.com/design/building-blocks/dialogs.html>

6.2.15 El texto dinámico probablemente debe ser seleccionable

Algo que afecta directamente a la usabilidad de la aplicación Android, es que si un TextView se utiliza para mostrar datos, el usuario podría querer copiar los datos y pegarlos en otro lugar. Si esto no se permite, el usuario se frustrará.

Para permitirlo, el TextView debe especificar el atributo `textIsSelectable="true"`. Si esto no es así, se avisa al desarrollador.

Más información:

http://developer.android.com/reference/android/widget/TextView.html#attr_android:textIsSelectable

6.2.16 Menús de Android

6.2.16.1 Falta el título del ítem del menú

Es importante que siempre se defina "android:title" para cada ítem si se trata de un menú. Google marca como obligatorio declarar siempre el atributo para el título.

Por ejemplo:

```
<item android:id="@+id/action_search"
      android:title="@string/action_search"
      yourapp:showAsAction="ifRoom" />
```

Más información:

<http://developer.android.com/guide/topics/ui/actionbar.html#ActionItems>

6.2.16.2 Falta el icono del ítem del menú

En este caso, el atributo "Android:icon" no es obligatorio, pero se recomienda también su uso encarecidamente por motivos de usabilidad.

Por ejemplo:

```
<item android:id="@+id/action_search"
      android:icon="@drawable/ic_action_search"
      android:title="@string/action_search"
      yourapp:showAsAction="ifRoom" />
```

Más información:

<http://developer.android.com/guide/topics/ui/actionbar.html#ActionItems>

6.2.17 Atributo inputType desaparecido

Proporcionar el atributo "android:inputType" para campos de texto permite optimizar por ejemplo el teclado que se muestra al usuario. Esto se considera un fallo de usabilidad si no se usa.

```
<EditText
  android:id="@+id/email_address"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:hint="@string/email_hint"
  android:inputType="textEmailAddress" />
```

Más información:

<http://developer.android.com/guide/topics/ui/controls/text.html>

6.2.18 Sobrescribiendo parámetros del layout

Por último, cabe mencionar que este problema me lo he encontrado yo mismo como desarrollador Android, y el problema de la interfaz viene cuando se usa el tag include. Se trata de un problema que tiene que ver con redefinir las dimensiones en un tag include, ya que si no

se tiene en cuenta lo siguiente, la interfaz dará problemas e incluso la aplicación puede cerrarse.

Cuando se tiene el siguiente código:

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:orientation="vertical">

  ...

  <include
    android:id="@+id/buttons_override"
    android:background="#ff0000"
    android:layout_width="fill_parent"
    layout="@layout/buttons"/>

</LinearLayout>
```

Se puede ver como se sobrescribe el layout mediante `layout="@layout/buttons"`. Cuando un desarrollador quiere sobrescribir otros atributos como el fondo por ejemplo, debe sobrescribir siempre el ancho y el alto (`layout_width` y `layout_height`).

El ejemplo anterior, se solucionaría de la siguiente forma:

```
<include
  android:id="@+id/buttons_override"
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  layout="@layout/buttons"/>
```

Todo está explicado en el siguiente enlace:

<http://stackoverflow.com/questions/2631614/does-android-xml-layouts-include-tag-really-work>

6.3 Diseño de la Base de Datos

6.3.1 Descripción del SGBD Usado

Para la implementación de este proyecto ha sido necesario usar una base de datos. Dado que MySQL es una opción muy extendida y usada por grandes compañías y que además dispone de gran documentación, ha sido la opción elegida.

La versión que ofrece el hosting es la 5.1. Esta versión proporciona todo lo necesario con respecto al soporte de PDO con lo que no existirá ningún problema.

6.3.2 Integración del SGBD en Nuestro Sistema

Como ya se dijo anteriormente, la base de datos se integra mediante la clase modelo de la aplicación que es "Report". Esta es la encargada de crear la conexión con la base de datos usando los datos de acceso que proporciona el hosting.

Si la aplicación fuese desplegada en localhost sólo habría que cambiar dichos datos, y no sería necesario ningún cambio más.

6.3.3 Diagrama E-R

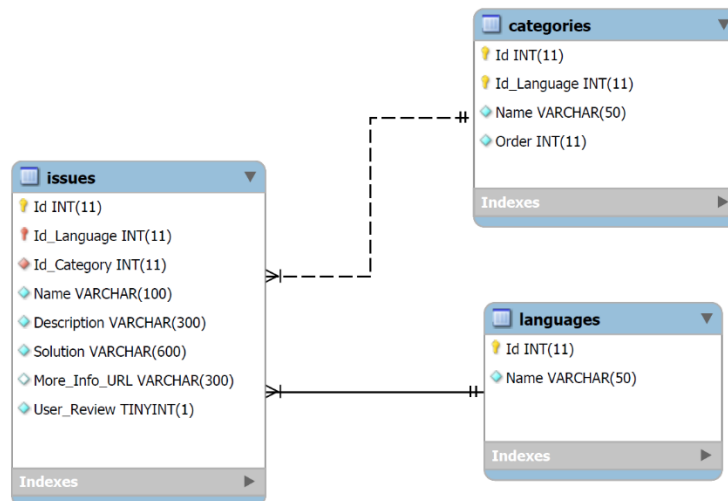


Ilustración 20: Diagrama E-R

La base de datos contiene 3 tablas.

Una llamada "categories" para almacenar las categorías en las que se divide el informe. Esta tabla tiene un ID para la categoría, otro ID para el idioma como clave foránea y el nombre de la categoría traducido en ese idioma. Los dos IDs forman la clave primaria.

Otra tabla llamada “languages” que contiene los idiomas (inglés y español). Simplemente tiene un ID que además es clave primaria y un nombre para identificar dicho idioma.

Por último, la tabla “issues” es la más importante ya que es la que guarda información relativa a cada problema que se analiza en el informe. Esta tabla contiene tres IDs uno para identificar el problema, otro para el idioma (clave foránea) y otro para identificar la categoría (clave foránea). Además de estos, cada “issue” tiene un nombre, una descripción, una solución, un enlace por si el usuario necesita conocer más del problema. Existen algunos de ellos que el usuario debe revisar ya que no es posible determinar con toda seguridad que sea un fallo. Por este motivo se ha creado un campo más en la tabla llamado “User_Review” como booleano.

A continuación se muestran de la base de datos algunos ejemplos de fallos que se detectan y lo que contiene la tabla “issues”:

Malformed file	The format is not XML.	Please, upload a file in XML	http://developer.android.com/guide/t	0
Formato incorrecto	El formato del fichero no es XML.	Por favor, sube un archivo en	http://developer.android.com/guide/t	0
Use "wrap_content" and "match_parent"	To ensure that your layout is flexible and adapts to different screen sizes, you should use "wrap_content" and "match_parent" for the width and height	Use "wrap_content" and "match_parent" size values instead of hard-coded sizes.	https://developer.android.com/training/multiscreen/screensizes.html#TaskUseWrapMatchPar	0
Uso de "wrap_content" y "match_parent"	Para asegurarse de que el diseño es flexible y se adapta a diferentes tamaños de pantalla, se debe utilizar "wrap_content" y "match_parent" para	Usa "wrap_content" y "match_parent" en vez de valores numéricos para definir tamaños.	https://developer.android.com/training/multiscreen/screensizes.html#TaskUseWrapMatchPar	0
Use RelativeLayout	RelativeLayout allows you to specify your layout in terms of the spacial relationships between components.	Substitute your layout to RelativeLayout.	https://developer.android.com/training/multiscreen/screensizes.html#TaskUseRelativeLayout	1
Usa RelativeLayout	RelativeLayout te permite diseñar la interfaz teniendo en cuenta las relaciones espaciales entre los	Sustituye la que tengas por RelativeLayout.	https://developer.android.com/training/multiscreen/screensizes.html#TaskUseRelativeLayout	1
Use Density-independent Pixels	Defining layout dimensions with pixels is a problem because different screens have different pixel densities, so the same number of pixels may correspond to different physical sizes on different	When specifying dimensions, always use either "dp" or "sp" units never "px".	https://developer.android.com/training/multiscreen/screendensities.html#TaskUseDP	0
Utiliza píxeles independientes de la densidad	Definir dimensiones con píxeles es un problema porque existen diferentes pantallas que tienen diferentes densidades de píxeles, por lo que el mismo número de píxeles puede	Al especificar las dimensiones, usa siempre "dp" o "sp" nunca "px".	https://developer.android.com/training/multiscreen/screendensities.html#TaskUseDP	0

Ilustración 21: Contenido de la tabla issues

6.4 Diseño detallado del sistema

Dada la gran cantidad de clases y ficheros que intervienen en el sistema no se ha elaborado un diagrama con todos ellos. Esto no impide explicar de forma detallada el diseño del sistema. Para ello, se irán viendo de forma progresiva las clases y archivos PHP incluidos de los distintos subsistemas de la aplicación.

6.4.1 Subsistema del informe

6.4.1.1 Subida del fichero XML mediante *upload-file.php*

Para poder analizar el fichero XML que haya elegido el usuario, primero se debe subir mediante AJAX. Cuando el usuario pulsa en el botón Analizar XML se realiza una petición POST mediante el archivo *upload-file.php* que realiza el proceso de subida al servidor. El formulario utiliza la función “*ajaxSubmit*” que provee el plugin de jQuery llamado *jQuery Form*.

Además antes de realizar la subida del fichero, se valida el campo para el fichero que ha elegido el usuario y se comprueba que sea un fichero XML y pese menos de 5Mb. Por motivos de privacidad, una vez que el fichero ha sido procesado, se borra del servidor sin dejar ningún tipo de rastro.

6.4.1.2 Clase *Report*

Report
-private \$db;
+public function __construct() +public function loadAllCategories(\$lang) +public function loadIssues(\$lang) +public function loadFormatCategorie(\$lang) +private function executeQuery(\$query)

Ilustración 22: Clase *Report*

Esta clase es el Modelo en el patrón MVC. Es la encargada de establecer la conexión con la base de datos y de interactuar con ella. Sólo tiene un atributo llamado “db” que es un objeto de tipo PDO (objetos que sirven para poder acceder a bases de datos). Todos los métodos que se verán en esta clase, utilizan sentencias SQL para cargar la información. Dichos métodos son:

- `public function __construct()`

Este es el constructor de la clase y en él se inicializa el atributo “db” con un objeto PDO usando los datos que proporciona el hosting para la conexión con la base de datos.

- `public function loadAllCategories($lang)`

Este método se encarga de cargar todas las categorías en las que está dividido el informe generado. Recibe el idioma que se muestra en la página. Devuelve un *array* con cada una en dicho idioma.

- `public function loadIssues($lang)`

Para generar el informe, este método carga todos los elementos que se analizan en un determinado idioma. Devuelve un *array* con todos ellos. Cada elemento tiene, un nombre, descripción, categoría, solución en caso de ser erróneo y el enlace para saber más.

- `public function loadFormatCategorie($lang)`

Cuando el usuario sube un fichero con un formato incorrecto, solo se carga la categoría asociada y no todas. También recibe el idioma, y devuelve un *array* que la contiene.

- `private function executeQuery($query)`

Este método es llamado por los anteriores, y es el encargado de preparar la *query* en SQL, ejecutarla y devolver el resultado. Dado que todos los anteriores métodos lo usaban, se extrajo como un método aparte.

6.4.1.3 Clase ReportsController

ReportsController
<pre>-const ANDROID_NAMESPACE = "http://schemas.android.com/apk/res/android"; -const CARDVIEW_NAMESPACE = "http://schemas.android.com/apk/res-auto"; -private \$allIssues; -private \$model; -private \$xml; -private \$lang; -private \$text; -private \$issuesDetected = array(); -private \$allElementsAffected = array();</pre>
<pre>+public function __construct(\$xml, \$lang, \$text) +public function detectIssues() +private function addIssue(\$node, \$issue) +private function analyzeNodesNonRecursively() +private function malformedFile() +private function useRelativeLayout() +private function compoundDrawable() +private function optimizeByMargin() +private function analyzeNodesRecursively(\$nodes) +private function useWrapContentMatchParent(\$nodes) +private function useDensityIndependentPixels(\$nodes) +private function useContentDescription(\$nodes) +private function useFocusable(\$nodes) +private function separateIntoStyle(\$nodes) +private function useElevation(\$nodes) +private function listsAndCards(\$nodes) +private function checkCardAttributes(\$nodes) +private function checkFontSizes(\$nodes) +private function checkStringsValues(\$nodes) +private function checkActionVerbs(\$nodes) +private function checkSelectableText(\$nodes) +private function checkItemTitle(\$nodes) +private function checkItemIcon(\$nodes) +private function checkInputType(\$nodes) +private function overridingParams(\$nodes)</pre>

Ilustración 23: Clase ReportsController

Esta es la clase principal de la aplicación. Su objetivo principal, es analizar mediante los métodos que se explican a continuación los posibles errores del fichero XML.

Como atributos contiene: dos constantes, que almacenan los namespaces de Android (el general y para el componente CardView). Otros atributos que contiene son, el modelo, el xml parseado, el idioma, y dos arrays adicionales. Estos arrays sirven para saber qué problemas de entre todos contiene realmente la interfaz XML analizada (“issuesDetected”) y dado que los

ficheros XML se basan en una jerarquía recursiva y Android también, es posible que haya varios nodos afectados por el mismo problema (“allElementsAffected”).

Los métodos que se explican a continuación, son los más importantes de todo el proyecto, pues como se ha dicho, forman el núcleo de la aplicación:

- `public function __construct($xml, $lang)`

Este es el constructor de la clase y en él se inicializa el atributo para el modelo con un objeto Report nuevo, además de recibir el objeto XimpleXML y el idioma.

- `public function detectIssues()`

Este método comprueba el formato del fichero y en caso de que sea correcto, comienza a analizar todos los nodos. Como ya se mencionó algunos problemas necesitan analizarse de forma recursiva y como se verá más adelante se ha separado en dos métodos distintos.

Si el formato es incorrecto, no se analiza nada, se carga la categoría de formato incorrecto para mostrarle al usuario únicamente ese fallo.

Finalmente se preparan todos los arrays que contienen los problemas detectados, los nodos afectados y se llama a la vista.

- `private function addIssue($node, $issue)`

Este método es llamado por los demás que comprueban cada problema, y añade el nodo y el problema en cuestión a los array que usa la vista posteriormente. Para hacer esto, se usa el ID del componente de Android.

- `private function analyzeNodesRecursively($nodes)`

Dado que hay problemas que necesitan que se analice de forma recursiva todos los nodos XML, éste método se encarga de llamar a dichos métodos y de esta forma existe una separación entre lo que es recursivo y lo que no.

- `private function analyzeNodesNonRecursively()`

Éste es el método que llama a los que no necesitan recursión. Además, este método y el anterior comprueban que la interfaz XML que ha subido el usuario no sea un menú de Android. Los menús en Android son Interfaces especiales, y no se necesita comprobar ciertos criterios en estas interfaces.

A continuación se detallan los métodos que no usan recursión:

- `private function malformedFile()`

Método que comprueba el objeto XML y detecta si es vacío. En este caso significa que hubo un error de procesamiento al parsear el fichero mediante la librería SimpleXML, y esto equivale a que ha habido un error de formato en el fichero.

- `private function useRelativeLayout()`

En la gran mayoría de los casos, las interfaces en Android se deben diseñar usando RelativeLayout en vez de LinearLayout. Este método requiere revisión por parte del usuario pues no en todos los casos es así.

- `private function compoundDrawable()`

Según la guía de desarrollo de Google, siempre que hay un elemento LinearLayout que contiene un ImageView y un TextView se debe cambiar y usar el componente compound drawable. Este método detecta el layout usado y comprueba si existen esos dos componentes en cuyo caso avisa de la posible mejora.

- `private function optimizeByMergin()`

Otra buena práctica que se recomienda es optimizar la interfaz me modo que si el elemento raíz es un FrameLayout y no aporta un atributo para el fondo u otros atributos adicionales, se puede sustituir por el componente merge de Android. El método comprueba que esto sea así y lo marca como fallo en tal caso.

A continuación se detallan los métodos que sí requieren recursión:

- `private function useWrapContentMatchParent($nodes)`

Para asegurar un diseño flexible y que se adapte a cualquier dispositivo, este método comprueba que los atributos `layout_width` y `layout_height`, para la altura y ancho de los componentes, contengan exactamente o `wrap_content` o `match_parent`.

- `private function useDensityIndependentPixels($nodes)`

Al igual que en la web, no se deben usar pixeles en las dimensiones. En su lugar, se deben usa `dp` o `sp`.

- `private function useContentDescription($nodes)`

Para los componentes `ImageButton`, `ImageView` y `CheckBox`, es necesario establecer el atributo `contentDescription`. Además para el componente `EditText` es necesario rellenar el atributo `hint`.

- `private function useFocusable($nodes)`

La aplicación Android debería soportar más métodos de navegación que la pantalla táctil. Muchos dispositivos Android vienen con hardware de navegación diferente

como un D-Pad, teclas, o un trackball. Por este motivo es importante que para cada componente, se establezca el atributo “focusable” a true, además de definir los siguientes cuatro: nextFocusDown, nextFocusUp, nextFocusRight y nextFocusLeft.

- `private function separateIntoStyle($nodes)`

Este método comprueba el número de atributos de un nodo. Si es superior a 10, se recomienda que se usen estilos de Android para no sobrecargar el código y mejorar la eficiencia. Además de esta forma separamos el estilo del contenido, un patrón como se ha visto muy usado.

- `private function useElevation($nodes)`

Para las nuevas versiones de Android que usan Material Design, es necesario definir el atributo “elevation” para los componentes RelativeLayout, LinearLayout, FrameLayout, GridLayout y DrawerLayout. Si no es así para alguno se marca para revisión.

- `private function listsAndCards($nodes)`

Android Lollipop también trae nuevos componentes más eficientes para las listas. En vez de usar ListView se debería de usar RecyclerView. También está disponible CardView. Estos componentes son una versión mejorada de los anteriores.

- `private function checkCardAttributes($nodes)`

Si el usuario está usando CardView en sus interfaces, es obligatorio que defina los atributos cardElevation, cardCornerRadius y cardBackgroundColor. Este método se encarga de comprobarlo y de informar en caso contrario.

- `private function checkFontSizes($nodes)`

Comprueba que el tamaño de cualquier texto sea superior a 11 puntos. Un tamaño inferior se puede decir que es demasiado pequeño.

- `private function checkStringsValues($nodes)`

Por motivos de internacionalización, y para que las interfaces siempre estén bien traducidas, todas las cadenas de texto deben estar referenciadas. Esto evita que se introduzcan strings en el propio código, y todos tengan que estar en el correspondiente fichero en la carpeta res (resources). Para ello el método comprueba que todos los textos empiecen por “@string/”.

- `private function checkActionVerbs($nodes)`

Las buenas prácticas también recomiendan que el texto de las acciones de los botones sean verbos. Dado que esto es muy difícil comprobarlo programáticamente si se encuentra un botón se marca para su revisión por parte del usuario.

- `private function checkSelectableText($nodes)`

Si un `TextView` se utiliza para mostrar datos, el usuario podría querer copiar los datos y pegarlos en otro lugar. Para permitir esto, el `TextView` debe especificar el atributo `textIsSelectable`. Aun así esto debe revisarlo el usuario en cada caso particular.

- `private function checkItemTitle($nodes)`

En caso de que la interfaz sea un menú de Android es necesario que defina el atributo para el título.

- `private function checkItemIcon($nodes)`

Además del título, es opcional definir el atributo para el icono. No es obligatorio pero es recomendable para cada item.

- `private function checkInputType($nodes)`

Proporcionar el atributo `inputType` para campos de texto permite optimizar por ejemplo el teclado que se muestra al usuario. Este método comprueba que esté definido.

- `private function overridingParams($nodes)`

En caso de que la interfaz este usando un componente mediante `include`, y quiera sobrescribir algún atributo, debe definir siempre el ancho y el alto (`layout_width` y `layout_height`). Si no es posible que en buena parte de los dispositivos la interfaz deje de funcionar en versiones anteriores a la última.

6.4.1.4 Clase *ReportView*

ReportView
<pre>-private \$lang; -private \$text; -private \$categories; -private \$issuesDetected; -private \$issues; -private \$allElementsAffected;</pre>
<pre>+public function __construct(\$lang, \$text, \$categories, \$issuesDetected, \$issues, \$allElementsAffected) +public function output()</pre>

Ilustración 24: Clase *ReportView*

Una vez que toda la lógica ha sido procesada, la vista se encarga de pintarlo en pantalla generando el informe a partir de los fallos detectados, las categorías y demás.

- `public function output()`

Este es el método encargado de retornar el código html procesado. Primero comienza generando el código para los botones superiores. A continuación elabora cada una de las tablas con su categoría, que contienen cada fallo detectado en distintas filas. Cada columna contiene, el nombre, la descripción, la solución o el enlace para más información. Al final de todo, crea el contenido del PDF y la barra de progreso.

6.4.1.5 *Archivo create-pdf.php y pdf/create.php*

En el momento que se analiza el XML y se crea el informe, también se guarda en sesión una variable con el contenido, que servirá después para generar el PDF. De esto se encarga el archivo `create-pdf.php`

Por lo tanto, cuando el usuario pulsa sobre el botón Crear PDF en la página del informe, se genera una petición post que recoge el fichero situado en `pdf/create.php` que es el que realmente usa la librería TCPDF.

Una vez que el PDF es creado, se abre en una nueva pestaña para que lo visualice el usuario y lo guarde si quiere.

6.4.2 Otros archivos y funciones

6.4.2.1 *Archivo language.php*

Para permitir la internacionalización de la página se ha diseñado la base de datos y todas las páginas pensando en esta característica. Es el fichero `language.php` el que se encarga de detectar el idioma del usuario, y cargarlo de entre los disponibles. Como ya se dijo en capítulos anteriores, solo hay disponibles dos idiomas, pero cabe la posibilidad de añadir cualquier otro de forma sencilla y sin cambiar la estructura existente. Todo se hace a través de la sesión del usuario.

Los ficheros `es.php` y `en.php` son los que contienen las cadenas de texto traducidas. Cuando un usuario cambia de idioma, se carga el fichero correspondiente y se realizan peticiones a la base de datos para cargar el informe en otro idioma si fuese necesario

6.4.2.2 *Plugin de jQuery para las cookies*

La privacidad es algo muy importante y que hay que cuidar de cara al usuario. Para avisar de la política de privacidad y los avisos legales, se ha utilizado un plugin de jQuery llamado "cookieBar". Este plugin permite mostrar una barra en la parte superior, cuando se entra por primera vez o todavía no se ha aceptado la política. En caso de que el usuario deniegue dicha

política y la rechace, no se ejecutará ningún código para el análisis y toma de datos estadísticos.

6.4.2.3 *Formulario de contacto*

Como se explicó anteriormente, el usuario dispone de un formulario para contactar en caso de cualquier problema, o para resolver cualquier duda o proponer una sugerencia. Este formulario contiene una serie de campos todos ellos validados, además de un Captcha y de un aviso de términos legales.

La librería usada para el Captcha se llama ReCaptcha y según la Wikipedia “es una extensión que se utiliza para reconocer texto presente en imágenes. Emplea por tanto la prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano para, a su vez, mejorar la digitalización de textos”. Esta es una forma de garantizar que no se reciben correos de spam. Además los lugares donde aparece el correo de contacto, están procesados para ofuscarlo mediante CSS y que ningún bot malicioso pueda leerlo sin más.

6.4.2.4 *Archivo validate.php*

Para la validación de datos cuando el usuario envía un email, se utiliza el método más conocido como es “htmlspecialchars()”. Este método comprueba caracteres especiales que puedan poner en peligro la seguridad. Además también se hace uso de métodos como trim() o stripslashes() para quitar las barras de un string con comillas escapadas.

```
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
```

Una vez comprobados los campos, se envía el email mediante la función PHPMailer(). Los datos de configuración del servidor los proporciona el hosting y la conexión se realiza mediante SMTP.

En caso de que se produzca algún problema con el Captcha se retorna al formulario y se informa al usuario. Ninguno de los campos rellenos se pierde, pues se transfieren por variables de sesión.

6.5 Diagramas de interacción de los escenarios del subsistema

6.5.1 Subsistema del informe

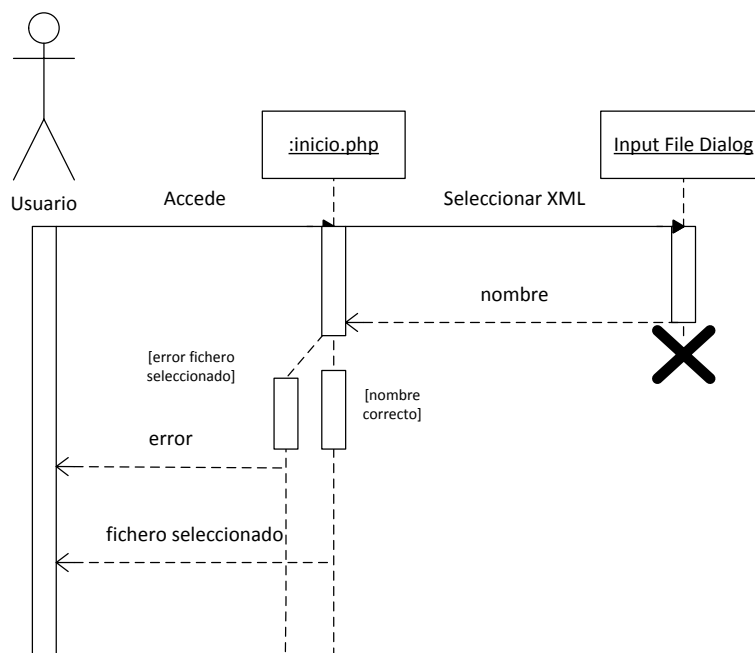


Ilustración 25: Diagrama de interacción del caso de uso seleccionar XML

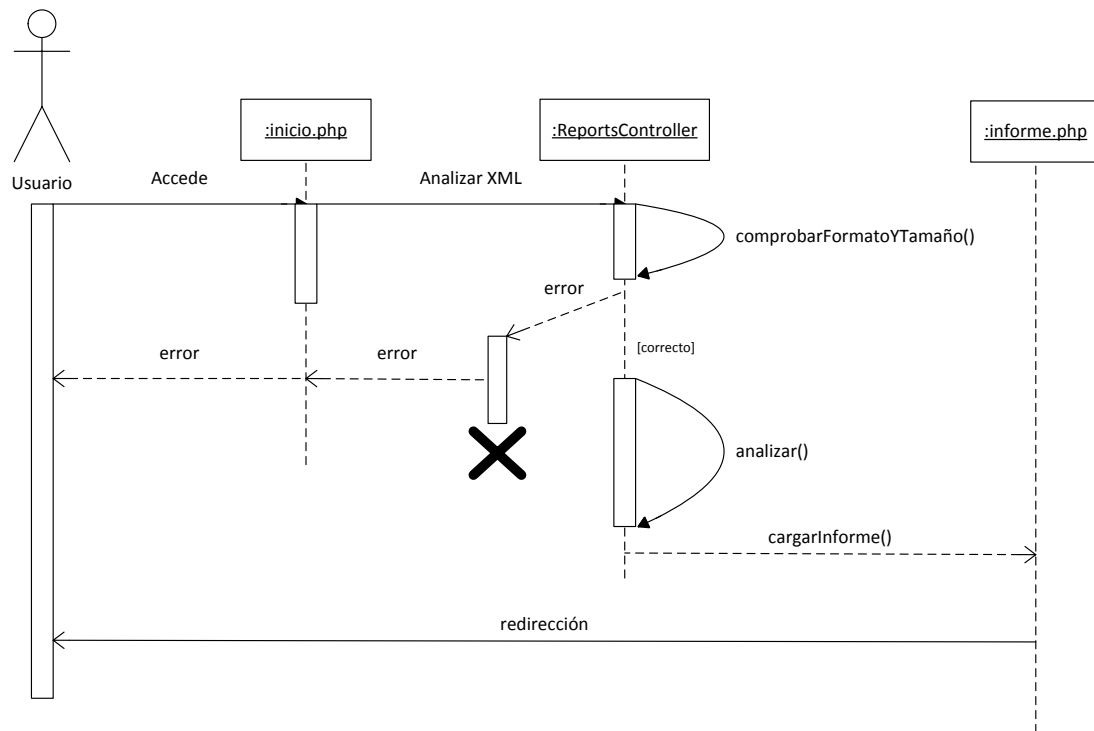


Ilustración 26: Diagrama de interacción del caso de uso Analiza XML

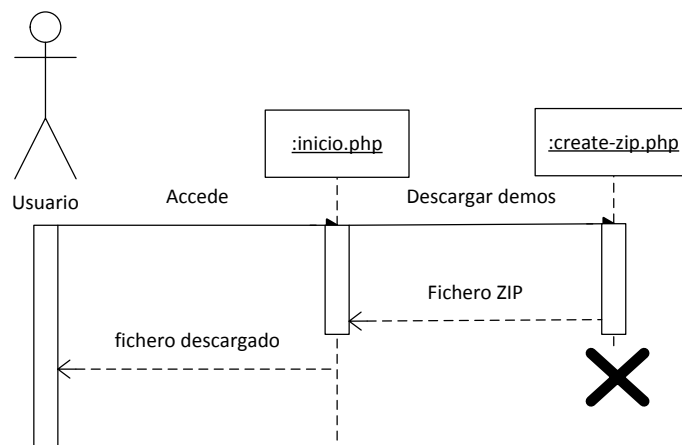


Ilustración 27: Diagrama de interacción del caso de uso Descargar ejemplos

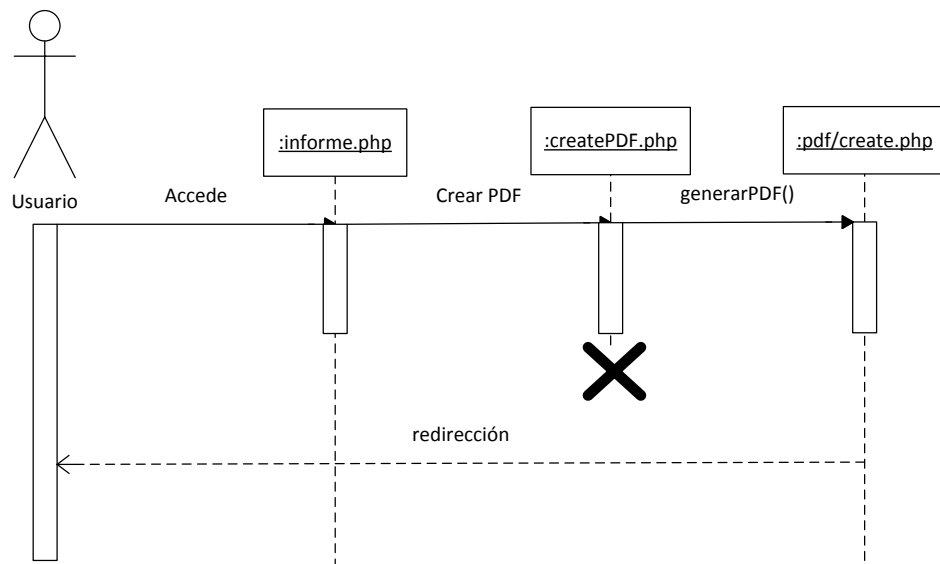


Ilustración 28: Diagrama de interacción del caso de uso Crear PDF

6.6 Herramientas y Programas Usados para el Desarrollo

6.6.1 PHPStorm de JetBrains (IntelliJ)

Este ha sido el entorno de desarrollo usado para llevar a cabo todo el desarrollo del proyecto. Se trata posiblemente del editor de PHP más potente que existe actualmente. La versión utilizada ha sido la 8.0.3.

Es un editor muy potente, que permite trabajar con múltiples entornos y tiene infinidad de parámetros de configuración y personalización. En su comparación con Eclipse (PDT), destaca con gran superioridad en aspectos importantes como la usabilidad.

Se ha utilizado para desarrollar todo el código, pudiendo trabajar a través de FTP.

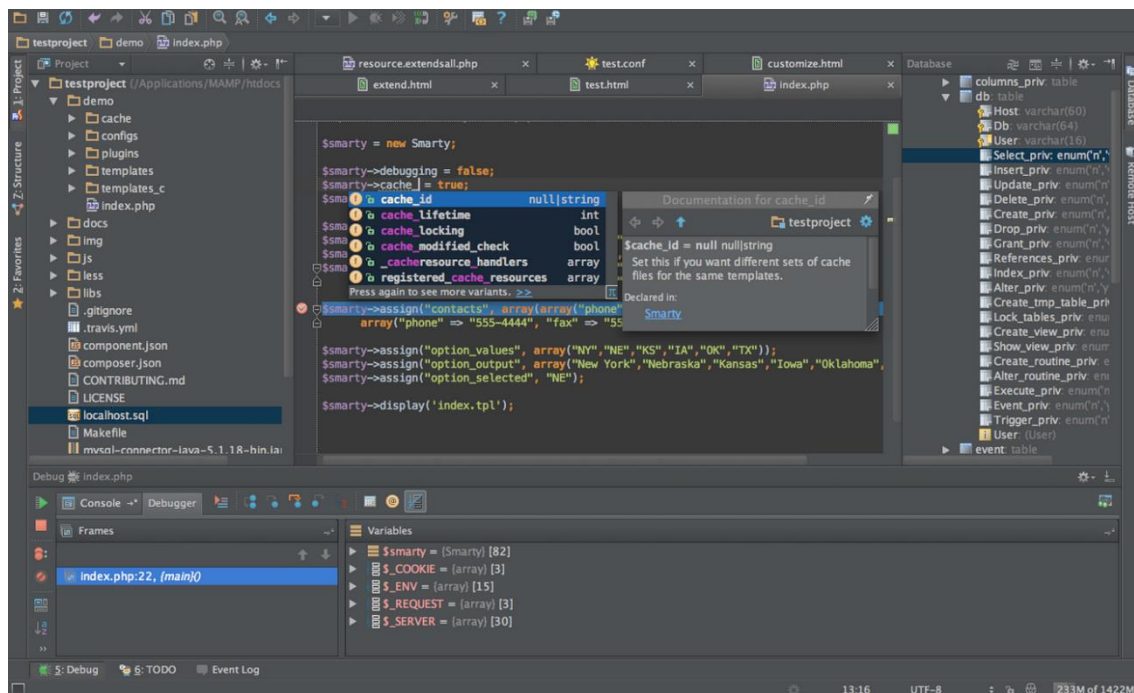


Ilustración 29: Entorno PHPStorm

6.6.2 MySQL Workbench de Oracle

Para toda la gestión y administración de la base de datos se ha utilizado este programa, también muy conocido y con grandes funcionalidades. Se ha utilizado para toda la creación de la base de datos, y para generar el diagrama con todas las tablas.

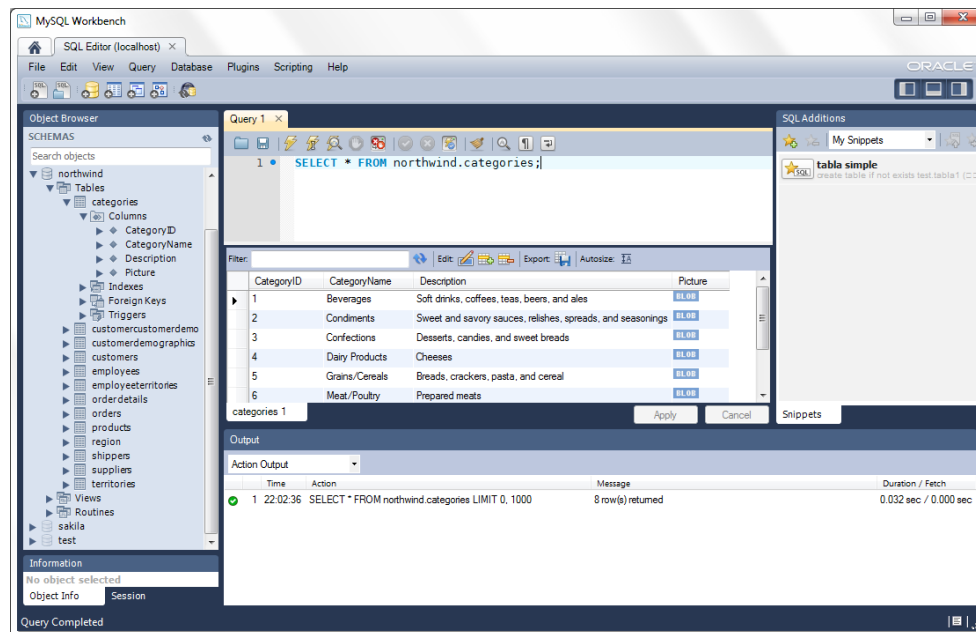


Ilustración 30: Entorno MySQL Workbench

6.6.3 Filezilla

Se trata de un cliente para la gestión de archivos mediante FTP. Se ha utilizado principalmente para subir y bajar ficheros del servidor de hosting donde se encuentra toda la web. La versión utilizada ha sido la 3.10.3.

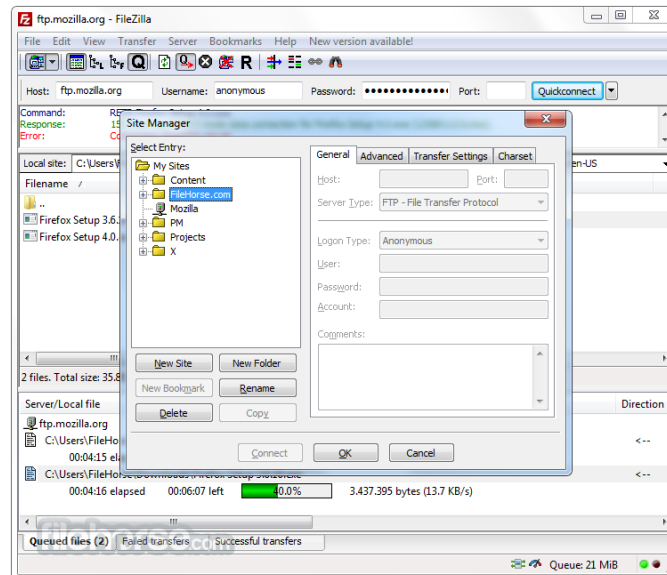


Ilustración 31: Entorno Filezilla

6.6.4 Herramientas para Webmasters de Google

Dado que la web está disponible en Internet se ha hecho uso de estas herramientas gratuitas, que según Wikipedia, “permiten a los creadores de páginas web comprobar el estado de la indexación de sus sitios en internet por el buscador y optimizar su visibilidad”. Estas herramientas permiten:

- Enviar y comprobar un mapa del sitio (sitemap).
- Comprobar y ajustar la frecuencia de indexación, y ver con qué frecuencia Googlebot visita un sitio determinado.
- Producir y comprobar un archivo robots.txt
- Enumerar los enlaces de páginas internas y externas a ese sitio.
- Ver qué palabras clave en las búsquedas de Google han llevado a ese sitio, y la tasa de clic en las palabras de esa lista.
- Ver estadísticas sobre cómo google indexa el sitio, y los errores que haya podido encontrar.
- Fijar un dominio preferido (por ejemplo, hacer que ejemplo.com tenga preferencia sobre www.ejemplo.com), lo que determinará cómo la URL del sitio aparezca en los resultados de las búsquedas.

Además, es posible enlazar ésta herramienta de Google con Google Analytics para medir de mejor manera y unificada, las estadísticas del sitio web.

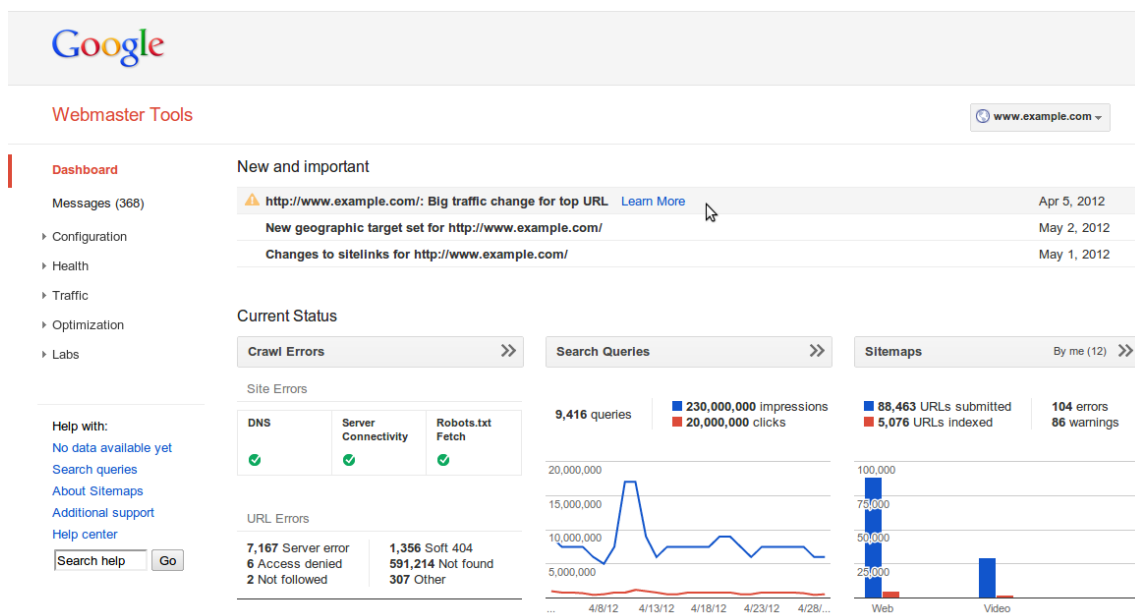


Ilustración 32: Ejemplo de Google Webmasters Tools

6.6.5 Google Analytics

Para llevar a cabo un seguimiento detallado de los usuarios que visitan la página, con qué dispositivos lo hacen y gran cantidad de datos adicionales, se ha hecho uso de Google Analytics. Insertando un simple código de Javascript se pueden obtener cantidad de datos útiles para el desarrollador y analizarlos mediante informes.

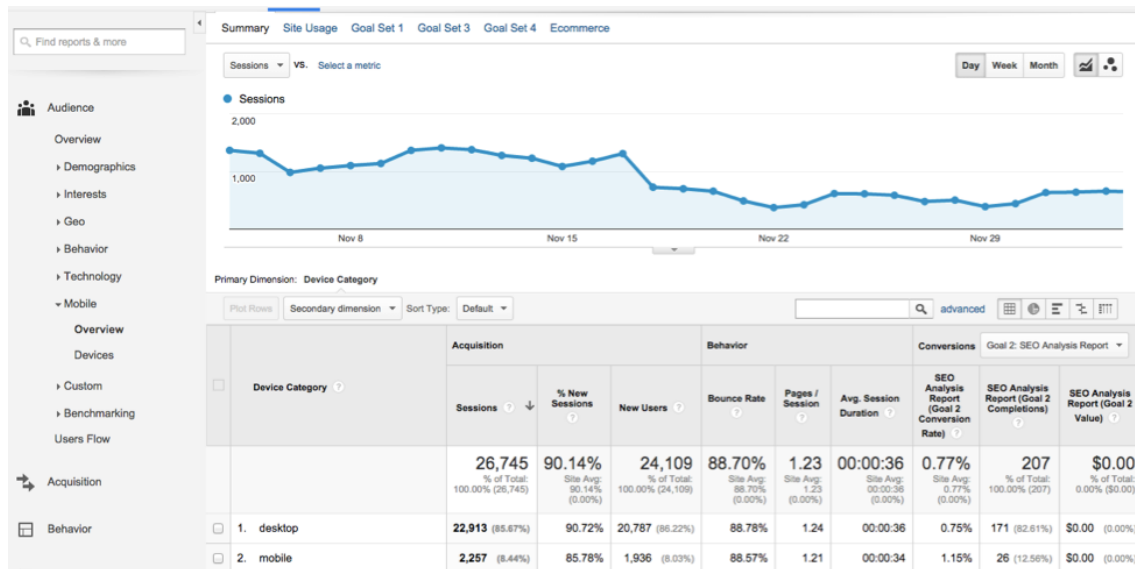


Ilustración 33: Ejemplo de Google Analytics

Capítulo 7. Desarrollo de las Pruebas

7.1 Preparación de las pruebas

7.1.1 Pruebas funcionales

Para la realización de estas pruebas, primero se explicarán las diferentes clases de equivalencia que se van a aplicar en la realización de las mismas. Las funciones principales que han sido escogidas pertenecen a los principales casos de uso.

7.1.1.1 Función Seleccionar XML

Clase de equivalencia	Valores válidos		Valores inválidos	
Longitud del nombre del XML que se introduce para seleccionar.	Longitud > 0		Longitud = 0	
	V01	1	I01	0
	V02	>1		
Contenido del campo donde se introduce el nombre del XML que se quiere seleccionar.	Cadena de caracteres que no contenga ninguno de los siguientes: " < > : & ' "		Cadena de caracteres que contenga alguno de los siguientes: " < > : & ' "	
	V03	Cualquier combinación válida	I02	"
			I03	<
			I04	>
			I05	:
			I06	
			I07	&
			I08	'
Extensión del archivo seleccionado.	Fichero compatible		Fichero incompatible	
	V04	Un fichero que tenga la extensión de XML	I09	Fichero con extensión distinto a XML
			I10	Fichero inexistente

7.1.1.2 Función Analizar XML

Clase de equivalencia	Valores válidos		Valores inválidos	
Formato del contenido del fichero XML que ha sido seleccionado	Formato válido		Formato inválido	
	V05	Formato XML de Android	I11	Formato no XML
			I12	Formato XML pero no Android
			I13	Formato corrupto
			I14	Fichero vacío
Tamaño del fichero XML.	Inferior a 5MB		Superior a 5MB	
	V06	Cualquier combinación válida	I15	Fichero de 6MB

7.1.2 Pruebas de Usabilidad

Como ya se comentó durante la fase de Análisis, se realizan pruebas a dos tipos de usuarios para luego mostrarles el cuestionario que se describe más adelante.

Los usuarios que se han usado para estas pruebas, son alumnos de la Universidad de Oviedo, así como otros desarrolladores de Android compañeros. Dentro de todos ellos, ha habido desarrolladores de Android y otros que no, aunque la gran mayoría (un 80%) sí lo eran.

Dentro de las funciones que se pedían realizar están:

- Buscar la herramienta a través de los principales navegadores
- Subir un fichero XML y analizarlo
- Descargar ejemplos de interfaces de Android
- Traducir la página a otro idioma
- Ponerse en contacto
- Cambiar las opciones de visualización del informe
- Guardar en PDF

A continuación se muestran las distintas preguntas del cuestionario que los usuarios han realizado:

- ¿Se identifican bien los objetivos de la web?
- En caso afirmativo, ¿para qué dirías que sirve?
- ¿Te parece la URL clara y fácil de recordar?
- ¿Y las URL internas?
- En caso negativo, ¿has tenido algún problema? ¿Cuál? También puedes proponer URLs.

- ¿Con qué navegador trabajas?
- Puntúa del 0 al 10 el aspecto de la web.
- ¿Cambiarías algo de dicho aspecto? ¿El qué?
- ¿Crees que falta alguna sección fundamental y que no aparece? ¿Reordenarías alguna otra?
- Puntúa del 0 al 10 la facilidad de uso de la web. (0-muy difícil, 10-muy sencilla)
- ¿Qué es lo que te pareció más difícil de entender? Escribe lo que te desconcertó al verlo.
- ¿Se muestra claramente la identidad del sitio a través de todas las páginas?
- ¿Probaste los ejemplos? ¿Has tenido dificultades con ellos? ¿Cuáles?
- Puntúa del 0 al 10 si la estructura general del sitio web está orientada al usuario.
- Puntúa del 0 al 10 si crees que el aspecto general se corresponde con los objetivos, características, contenidos y servicios que ofrece el sitio web
- ¿Te parece el título de las páginas correcto?
- En caso negativo, escribe los que cambiarías o el por qué.
- ¿Los enlaces son fácilmente reconocibles como tales?
- ¿Has tenido algún problema con el tamaño de los textos?
- ¿Te ha sido difícil traducir la página? ¿Has encontrado alguna traducción mal? ¿Cuál?
- ¿Te parecen suficientes los nombres, descripciones y soluciones de cada fallo? ¿Echas de menos algún campo más?
- ¿Has identificado bien lo que son fallos y lo que no?
- ¿Qué te gustaría que se mejorase (cualquier aspecto)?

Además del cuestionario anterior, se pasará una guía de usabilidad, propuesta por Yusef Hasan Montero en el siguiente enlace:

<http://www.nosolousabilidad.com/articulos/heuristica.htm>

7.1.3 Pruebas de Rendimiento

Para la realización de la pruebas de rendimiento se ha elegido una herramienta online, que mide la velocidad de carga, y parámetros relacionados con la eficiencia en general y el SEO.

Esta herramienta se llama GTMetrix y es un analizador de páginas web muy conocido en la comunidad de desarrolladores. Además hace uso de más herramientas como se verá más adelante.

7.2 Implementación de las pruebas

7.2.1 Pruebas funcionales

A continuación se muestran los resultados de las distintas pruebas funcionales de caja negra que se han llevado a cabo:

7.2.1.1 Función Seleccionar XML

Caso de prueba	P01
Parámetros a introducir	Nombre: ""
Salida esperada	La aplicación no permitirá introducir un nombre vacío. No ocurrirá nada.
Resultado obtenido	El esperado.
Valores límite cubiertos	I001

Caso de prueba	P02
Parámetros a introducir	Nombre: "a" (fichero existente)
Salida esperada	La aplicación seleccionará el fichero XML.
Resultado obtenido	El esperado.
Valores límite cubiertos	V001, V006, V007

Caso de prueba	P03
Parámetros a introducir	Nombre: "main_activity" (fichero existente)
Salida esperada	La aplicación seleccionará el fichero XML.
Resultado obtenido	El esperado.
Valores límite cubiertos	V001, V006, V007

Caso de prueba	P04
Parámetros a introducir	Nombre: "cpu"-1"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I002

Caso de prueba	P05
Parámetros a introducir	Nombre: "cpu<<1"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I003

Caso de prueba	P06
Parámetros a introducir	Nombre: "cpu>>1"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I004

Caso de prueba	P07
Parámetros a introducir	Nombre: "cpu:ms"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I005

Caso de prueba	P08
Parámetros a introducir	Nombre: "cpu 21"

Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I006

Caso de prueba	P09
Parámetros a introducir	Nombre: "cpu&21"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I007

Caso de prueba	P10
Parámetros a introducir	Nombre: "cpu`21"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo válido.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P11
Parámetros a introducir	Nombre: "activity_main.xml"
Salida esperada	La aplicación seleccionará el fichero XML.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P12
Parámetros a introducir	Nombre: "asd.xml" (fichero inexistente)
Salida esperada	La aplicación informará al usuario que debe escoger un fichero que exista.

Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P13
Parámetros a introducir	Nombre: "imagen.jpg"
Salida esperada	La aplicación informará al usuario que debe escoger un archivo con extensión XML.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

7.2.1.2 Función Analizar XML

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero con formato XML pero no es una interfaz de Android)
Salida esperada	La aplicación generará el informe asociado informando de que no se trata de una interfaz Android y no mostrando nada más.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero con formato corrupto de XML)
Salida esperada	La aplicación generará el informe asociado pero informará de que el fichero contiene un formato no XML.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero vacío)

Salida esperada	La aplicación generará el informe asociado pero informará de que no es una interfaz Android.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero de 2MB con interfaz Android)
Salida esperada	La aplicación generará el informe asociado sin problema alguno.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero de 6MB)
Salida esperada	La aplicación generará el informe asociado pero informará de que no es una interfaz Android.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

Caso de prueba	P00
Parámetros a introducir	Nombre: "Android.xml" (fichero vacío)
Salida esperada	La aplicación generará el informe asociado pero informará de que no es una interfaz Android.
Resultado obtenido	El esperado.
Valores límite cubiertos	V003, I008

7.2.2 Pruebas de Usabilidad

Estas pruebas consistían en la elaboración de un cuestionario y de tareas por parte de los usuarios. Los resultados de este cuestionario se adjuntan en el anexo 3 de este documento.

A continuación se ve reflejada la tabla con la guía de usabilidad antes mencionada

Criterios	¿Cumplido?
Generales	
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	SI
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	SI
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web? El diseño de la página de inicio debe ser diferente al resto de páginas y cumplir la función de 'escaparate' del sitio.	SI
¿La estructura general del sitio web está orientada al usuario? Los sitios web deben estructurarse pensando en el usuario, sus objetivos y necesidades. La estructura interna de la empresa u organización, cómo funciona o se organiza no interesan al usuario.	SI
¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web? Ciertas combinaciones de colores ofrecen imágenes más o menos formales, serias o profesionales.	SI
¿Es coherente el diseño general del sitio web? Se debe mantener una coherencia y uniformidad en las estructuras y colores de todas las páginas. Esto sirve para que el usuario no se desoriente en su navegación.	SI
¿Es reconocible el diseño general del sitio web? Cuánto más se parezca el sitio web al resto de sitios web, más fácil será de usar.	SI
¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza? Las fechas que se muestren en la página deben corresponderse con actualizaciones, noticias, eventos...no con la fecha del sistema del usuario.	No
Identidad e Información	
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	SI
El Logotipo, ¿es significativo, identificable y suficientemente visible?	SI
El eslogan o <i>tagline</i> , ¿expresa realmente qué es la empresa y qué servicios ofrece?	No
¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?	SI
¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)	SI
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	SI
En artículos, noticias, informes... ¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	SI

<u>Lenguaje y Redacción</u>	
¿El sitio web habla el mismo lenguaje que sus usuarios? Se debe evitar usar un lenguaje corporativista. Así mismo, hay que prestarle especial atención al idioma, y ofrecer versiones del sitio en diferentes idiomas cuando sea necesario.	SI
¿Emplea un lenguaje claro y conciso?	SI
¿Es amigable, familiar y cercano? Es decir, lo contrario a utilizar un lenguaje constantemente imperativo, mensajes crípticos, o tratar con "desprecio" al usuario.	SI
¿1 párrafo = 1 idea? Cada párrafo es un objeto informativo. Trasmite ideas, mensajes...Se deben evitar párrafos vacíos o varios mensajes en un mismo párrafo.	SI
<u>Rotulado</u>	
Los rótulos, ¿son significativos? Ejemplo: evitar rótulos del tipo "haga clic aquí".	SI
¿Usa rótulos estándar? Siempre que exista un "estándar" comúnmente aceptado para el caso concreto, como "Mapa del Sitio" o "Acerca de..."	SI
¿Usa un único sistema de organización, bien definido y claro? No se deben mezclar diferentes. Los sistemas de organización son: alfabético, geográfico, cronológico, temático, orientado a tareas, orientado al público y orientado a metáforas.	SI
¿Utiliza un sistema de rotulado controlado y preciso? Por ejemplo, si un enlace tiene el rótulo "Quiénes somos", no puede dirigir a una página cuyo encabezamiento sea "Acerca de"	SI
El título de las páginas, ¿Es correcto? ¿Ha sido planificado? Relacionado con la capacidad para poder buscar y encontrar el sitio web.	SI
<u>Estructura y Navegación</u>	
La estructura de organización y navegación, ¿Es la más adecuada? Hay varios tipos de estructuras: jerárquicas, hipertextual, facetada,...	SI
En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?	SI
En el caso de ser puramente hipertextual, ¿Están todos los clúster de nodos comunicados? Aquí se mide la distancia entre nodos.	N/A
¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos,...)? Los enlaces no sólo deben reconocerse como tales, sino que su caracterización debe indicar su estado, y ser reconocidos como una unidad	SI
En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística? No se deben superar los 7±2 elementos, ni los 2 o, como mucho, 3 términos por elemento.	SI
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace? Relacionado con el nivel de significación del rótulo del enlace, aunque también con: el uso de globos de texto, información contextual, la barra de estado del navegador,...	SI

¿Se ha controlado que no haya enlaces que no lleven a ningún sitio? Enlaces que no llevan a ningún sitio: Los enlaces rotos, y los que enlazan con la misma página que se está visualizando (por ejemplo enlaces a la "home" desde la misma página de inicio)	SI
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación? ...como <i>breadcrumbs</i> , enlaces a la página de inicio,...recuerde que el logo también es recomendable que enlace con la página de inicio.	SI
Las imágenes enlace, ¿se reconocen como clicables? ¿Incluyen un atributo 'title' describiendo la página de destino? En este sentido, también hay que cuidar que no haya imágenes que parezcan enlaces y en realidad no lo sean.	SI
¿Se ha evitado la redundancia de enlaces?	SI
¿Se ha controlado que no haya páginas "huérfanas"? Páginas huérfanas: que aún siendo enlazadas desde otras páginas, éstas no enlacen con ninguna.	SI
<u>Layout de la Página</u>	
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central)	SI
¿Se ha evitado la sobrecarga informativa? Esto se consigue haciendo un uso correcto de colores, efectos tipográficos y agrupaciones para discriminar información. Los grupos diferentes de objetos informativos de una página deben ser 7±2.	SI
¿Es una interfaz limpia, sin ruido visual?	SI
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	SI
¿Se hace un uso correcto del espacio visual de la página? Es decir, que no se desaproveche demasiado espacio con elementos de decoración, o grandes zonas en "blanco", y que no se adjudique demasiado espacio a elementos de menor importancia.	SI
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página? (La jerarquía visual se utiliza para orientar al usuario)	SI
¿Se ha controlado la longitud de página? Se debe evitar en la medida de lo posible el <i>scrolling</i> . Si la página es muy extensa, se debe fraccionar.	SI
<u>Búsqueda (si es necesario, por la extensión del sitio, incorporar un buscador interno)</u>	
¿Se encuentra fácilmente accesible? Es decir: directamente desde la home, y a ser posible desde todas las páginas del sitio, y colocado en la zona superior de la página.	SI
¿Es fácilmente reconocible como tal?	SI
¿Permite la búsqueda avanzada? (siempre y cuando, por las características del sitio web, fuera de utilidad que la ofreciera)	No
¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?	N/A
¿La caja de texto es lo suficientemente ancha?	N/A
¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?	N/A

<i>Elementos Multimedia</i>	
¿Las fotografías están bien recortadas? ¿Son comprensibles? ¿Se ha cuidado su resolución?	SI
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario? (prestar especial atención a usuarios de otros países y culturas)	SI
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	SI
¿Se ha evitado el uso de animaciones cíclicas?	SI
<i>Ayuda</i>	
Si posee una sección de Ayuda, ¿Es verdaderamente necesaria? Siempre que se pueda prescindir de ella simplificando los elementos de navegación e interacción, debe omitirse esta sección.	No
En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"? La zona de la página más normal para incluir el enlace a la sección de Ayuda, es la superior derecha.	N/A
¿Se ofrece ayuda contextual en tareas complejas? (transferencias bancarias, formularios de registro...)	No
Si posee FAQs, ¿Es correcta tanto la elección como la redacción de las preguntas? ¿Y las respuestas?	No
<i>Accesibilidad</i>	
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	Si
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?	Si
¿Existe un alto contraste entre el color de fuente y el fondo?	Si
¿Incluyen las imágenes atributos 'alt' que describan su contenido?	Si
¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla? Se debe prestar atención a: <i>JScript</i> , <i>CSS</i> , tablas, fuentes...	Si
¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar <i>plugins</i> adicionales?	Si
¿Se ha controlado el peso de la página? Se deben optimizar las imágenes, controlar el tamaño del código <i>JScript</i> ...	Si
¿Se puede imprimir la página sin problemas? Leer en pantalla es molesto, por lo que muchos usuarios preferirán imprimir las páginas para leerlas. Se debe asegurar que se puede imprimir la página (no salen partes cortadas), y que el resultado es legible.	Si
<i>Control y Retroalimentación</i>	
¿Tiene el usuario todo el control sobre el interfaz? Se debe evitar el uso de ventanas pop-up, ventanas que se abren a pantalla completa, banners intrusivos...	Si
¿Se informa constantemente al usuario acerca de lo que está pasando? Si el usuario tiene que esperar hasta que se termine una operación, se debe mostrar un mensaje indicándole y que debe esperar, con el tiempo de espera estimado o una barra de progreso.	Si

¿Se informa al usuario de lo que ha pasado? Por ejemplo, cuando un usuario valora un artículo o responde a una encuesta, se le debe informar de que su voto ha sido procesado correctamente.	Si
Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema? Siempre es mejor intentar evitar que se produzcan errores a tener que informar al usuario del error.	Si
¿Posee el usuario libertad para actuar? NO restringir la libertad del usuario: Uso de animaciones que no pueden ser "saltadas", páginas en las que desaparecen los botones de navegación, no impida al usuario poder usar el botón derecho de su ratón...	Si
¿Se ha controlado el tiempo de respuesta? Esto tiene que ver con el peso de cada página (accesibilidad) y tiene relación con el tiempo que tarda el servidor en finalizar una tarea y responder. El tiempo máximo que esperará un usuario son 10 segundos	Si
<i>Aclaraciones</i>	
¿Se ha evaluado adecuadamente la orientación del usuario? (Donde estoy, como volver, que he visitado, que va a pasar)	Si
¿Se ha usado correctamente la publicidad?	N/A

7.2.3 Pruebas de Accesibilidad

7.2.3.1 Criterios generales

Las pruebas que se van a realizar para este proyecto, se harán sobre los dos principales navegadores que los usuarios han usado, esto es, Google Chrome y Mozilla Firefox.

Además, las páginas que han sido seleccionadas para dichas pruebas, han sido la de inicio, la página del informe, y la de contacto. El motivo es que la página principal de la aplicación no podía faltar, la del informe contiene múltiples tablas y la de contacto contiene un formulario.

Dado que la aplicación utiliza un framework de CSS que es responsive y se adapta a todas las resoluciones, la página se adapta perfectamente y no es necesario el uso de herramientas automáticas para comprobarlo.

7.2.3.2 Pruebas automáticas

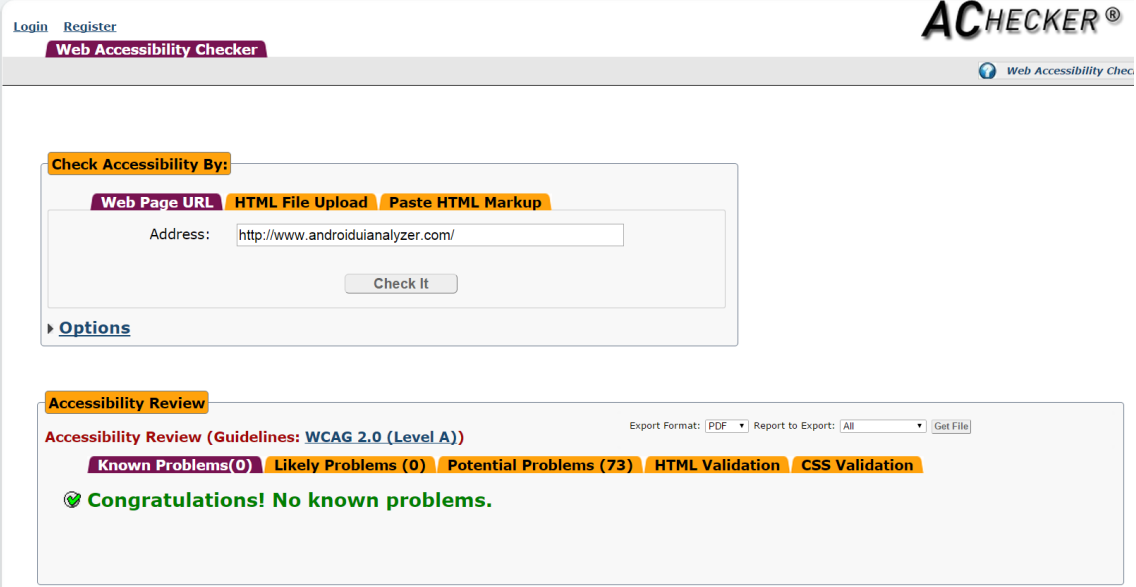
Para la realización de pruebas automáticas de accesibilidad se ha usado la herramienta AChecker⁵. Las pruebas se han establecido para un nivel de Accesibilidad A en la versión 2.0 de las WCAG. Las páginas que se han podido analizar, corresponden a la de inicio y la de contacto en este caso, ya que la página del informe se genera a partir de una acción por parte del usuario como es subir un XML. Por tanto en este caso, la herramienta de accesibilidad no tiene

⁵ <http://achecker.ca/checker/index.php>

sentido ya que la URL para el informe no contendrá lo mismo que cuando se analiza previamente.

Para el nivel A, los resultados han sido:

Página de Inicio:



The screenshot shows the ACHECKER Web Accessibility Checker interface. At the top, there are links for 'Login' and 'Register', and the 'Web Accessibility Checker' logo. The main content area is divided into two sections. The first section, 'Check Accessibility By:', has three tabs: 'Web Page URL', 'HTML File Upload', and 'Paste HTML Markup'. The 'Web Page URL' tab is active, showing an 'Address:' field with the URL 'http://www.androiduianalyzer.com/' and a 'Check It' button. Below this is an 'Options' link. The second section, 'Accessibility Review', shows the 'Export Format' set to 'PDF' and 'Report to Export' set to 'All'. It displays the following statistics: 'Known Problems (0)', 'Likely Problems (0)', and 'Potential Problems (73)'. Below these are links for 'HTML Validation' and 'CSS Validation'. A green checkmark icon is followed by the text 'Congratulations! No known problems.'

Ilustración 34: Resultados nivel A de la página de inicio

Página de contacto:



The screenshot shows the ACHECKER Web Accessibility Checker interface. At the top, there are links for 'Login' and 'Register', and the 'Web Accessibility Checker' logo. The main content area is divided into two sections. The first section, 'Check Accessibility By:', has three tabs: 'Web Page URL', 'HTML File Upload', and 'Paste HTML Markup'. The 'Web Page URL' tab is active, showing an 'Address:' field with the URL 'http://www.androiduianalyzer.com/contact' and a 'Check It' button. Below this is an 'Options' link. The second section, 'Accessibility Review', shows the 'Export Format' set to 'PDF' and 'Report to Export' set to 'All'. It displays the following statistics: 'Known Problems (0)', 'Likely Problems (0)', and 'Potential Problems (69)'. Below these are links for 'HTML Validation' and 'CSS Validation'. A green checkmark icon is followed by the text 'Congratulations! No known problems.'

Ilustración 35: Resultados nivel A de la página de contacto

Esto supone un nivel de accesibilidad A. Los problemas potenciales de todas las pruebas se han examinado, y la conclusión es que no son realmente problemas. Se trata de elementos que la

herramienta pide revisar por si acaso. Además, la página de informe se ha comprobado manualmente y sección por sección siguiendo los mismos patrones que para el resto de páginas, con lo que se podría asegurar que cumple también el nivel A, y como veremos a continuación el nivel AA.

Una vez subido se ha subido al nivel de accesibilidad AA, este nivel ha dado varios problemas que se explican más adelante, pero cabe decir de antemano, que se puede considerar este nivel como superado también.

Página de inicio:

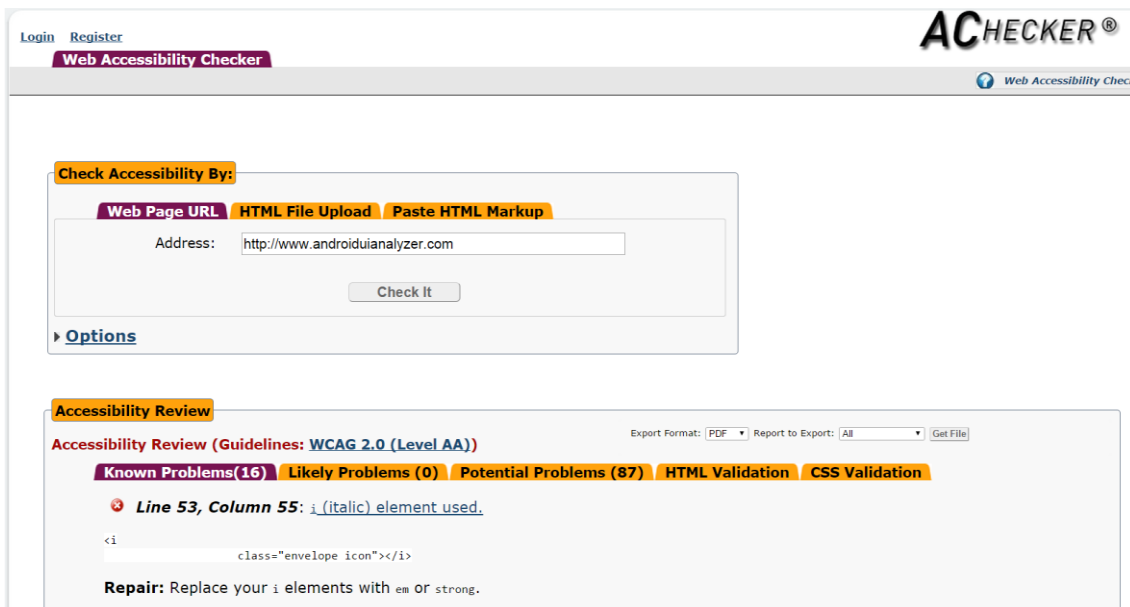
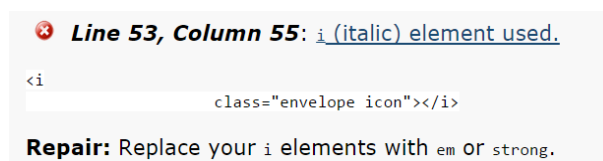


Ilustración 36: Resultados nivel AA

La herramienta encuentra 16 elementos con problemas. Muchos de ellos son repetidos, y en realidad solo hay de dos tipos distintos.

El primero es el siguiente:



Esto surge debido al uso de los iconos, que ofrece el framework, y que todos los demás frameworks como Bootstrap o Foundation usan. Se considera este fallo como un error del analizador de accesibilidad, y se espera que en un futuro esto ya no ocurra y que se adapten a los nuevos tiempos.

El otro problema que surgió, viene al detectar el contraste entre el fondo y el texto. Según la herramienta en este caso todo aparece en negro, cuando en realidad no es así.

La herramienta dice lo siguiente:

❗ **Line 156, Column 5:** [The contrast between the colour of text and its background for the element is not sufficient to meet WCAG2.0 Level AA.](#)

```
<h2 class="ui inverted header">All rights reserved &reg;. 2015</h2>
```

Repair: Use a colour contrast evaluator to determine if text and background colours provide a contrast ratio of 4.5:1 for standard text, or 3:1 for larger text. Change colour codes to produce sufficient contrast. <http://www.w3.org/TR/UNDERSTANDING-WCAG20/visual-audio-contrast-contrast.html#visual-audio-contrast-contrast-resources-head>

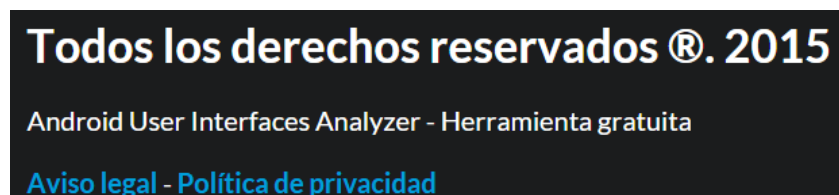
Fixed size example: **color contrast example**

Real size example (18 points): **color contrast example**

CSS rules for the element:

```
External CSS (http://www.androiduianalyzer.com/css/global.css):
    footer{
        background-color:#181C1D;
        margin-bottom:-2em;
    }
```

La realidad es:



Por lo tanto, este tipo de errores no se han aceptado como tales y por este motivo, se ha considerado que el nivel de AA también es alcanzado.

Por último, mencionar que se han probado otras herramientas como TAW⁶, obteniendo los mismos resultados.

7.2.3.3 Checklist del WCAG 1.0

La siguiente tabla se trata del *checklist* que el WCAG proporciona para verificar las pautas de accesibilidad de la aplicación web. Cada punto contiene antes un hiperenlace que va directamente a la web del WCAG para proporcionar explicaciones adicionales sobre el mismo. A continuación se muestran los resultados para este proyecto:

Puntos de verificación Prioridad 1:

En general (Prioridad 1)	Sí	No	N/A
1.1 Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). <i>Esto incluye:</i> imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, <i>GIFs</i> animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.	X		
2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.	X		

⁶ <http://www.tawdis.net/>

4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).	X		
6.1 Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.	X		
6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.	X		
7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.	X		
14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.	X		
Y si utiliza imágenes y mapas de imagen (Prioridad 1)	Sí	No	N/A
1.2 Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor.			X
9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.			X
Y si utiliza tablas (Prioridad 1)	Sí	No	N/A
5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.	X		
5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.			X
Y si utiliza marcos ("frames") (Prioridad 1)	Sí	No	N/A
12.1 Titule cada marco para facilitar su identificación y navegación.	X		
Y si utiliza "applets" y "scripts" (Prioridad 1)	Sí	No	N/A
6.3 Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, <i>applets</i> u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.	X		
Y si utiliza multimedia (Prioridad 1)	Sí	No	N/A
1.3 Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda visual de una presentación multimedia.			X
1.4 Para toda presentación multimedia dependiente del tiempo (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación.			X
Y si todo lo demás falla (Prioridad 1)	Sí	No	N/A
11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.			X

Puntos de verificación Prioridad 2:

En general (Prioridad 2)	Sí	No	N/A
---------------------------------	-----------	-----------	------------

<u>2.2</u> Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan el suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro [Prioridad 2 para las imágenes. Prioridad 3 para los textos].	X		
<u>3.1</u> Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.	X		
<u>3.2</u> Cree documentos que estén validados por las gramáticas formales publicadas.	X		
<u>3.3</u> Utilice hojas de estilo para controlar la maquetación y la presentación.	X		
<u>3.4</u> Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.	X		
<u>3.5</u> Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.	X		
<u>3.6</u> Marque correctamente las listas y los ítems de las listas.	X		
<u>3.7</u> Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.			X
<u>6.5</u> Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa.	X		
<u>7.2</u> Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado).	X		
<u>7.4</u> Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica.	X		
<u>7.5</u> Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad.	X		
<u>10.1</u> Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.	X		
<u>11.1</u> Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas.	X		
<u>11.2</u> Evite características desaconsejadas por las tecnologías W3C.	X		
<u>12.3</u> Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.	X		
<u>13.1</u> Identifique claramente el objetivo de cada vínculo.	X		
<u>13.2</u> Proporcione metadatos para añadir información semántica a las páginas y sitios.	X		
<u>13.3</u> Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).			X
<u>13.4</u> Utilice los mecanismos de navegación de forma coherente.	X		
Y si utiliza tablas (Prioridad 2)	Sí	No	N/A
<u>5.3</u> No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión	X		

alineada).			
5.4 Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.			X
Y si utiliza marcos ("frames") (Prioridad 2)	Sí	No	N/A
12.2 Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco.			X
Y si utiliza formularios (Prioridad 2)	Sí	No	N/A
10.2 Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente.	X		
12.4 Asocie explícitamente las etiquetas con sus controles.	X		
Y si utiliza "applets" y "scripts" (Prioridad 2)	Sí	No	N/A
6.4 Para los <i>scripts</i> y <i>applets</i> , asegúrese de que los manejadores de eventos sean independientes del dispositivo de entrada.			X
7.3 Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas.	X		
8.1 Haga los elementos de programación, tales como <i>scripts</i> y <i>applets</i> , directamente accesibles o compatibles con las ayudas técnicas	X		
9.2 Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.			X
9.3 Para los "scripts", especifique manejadores de evento lógicos mejor que manejadores de eventos dependientes de dispositivos.			X

Puntos de verificación Prioridad 3:

En general (Prioridad 3)	Sí	No	N/A
4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento.	X		
4.3 Identifique el idioma principal de un documento.		X	
9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.	X		
9.5 Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario.		X	
10.5 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos.	X		
11.3 Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).	X		
13.5 Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.	X		
13.6 Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.	X		
13.7 Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.			X

<u>13.8</u> Localice la información destacada al principio de los encabezamientos, párrafos, listas, etc.	X		
<u>13.9</u> Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas).			X
<u>13.10</u> Proporcione un medio para saltar sobre un <i>ASCII art</i> de varias líneas.		X	
<u>14.2</u> Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.	X		
<u>14.3</u> Cree un estilo de presentación que sea coherente para todas las páginas.	X		
Y si utiliza imágenes o mapas de imagen (Prioridad 3)	Sí	No	N/A
<u>1.5</u> Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.			X
Y si utiliza tablas (Prioridad 3)	Sí	No	N/A
<u>5.5</u> Proporcione resúmenes de las tablas.		X	
<u>5.6</u> Proporcione abreviaturas para las etiquetas de encabezamiento.		X	
<u>10.3</u> Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten correctamente los textos contiguos, proporcione un texto lineal alternativo (en la página actual o en alguna otra) para <i>todas</i> las tablas que maquetan texto en paralelo, en columnas de palabras.		X	
Y si utiliza formularios (Prioridad 3)	Sí	No	N/A
<u>10.4</u> Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.	X		

7.2.4 Pruebas de Rendimiento

Para la realización de estas pruebas, se ha hecho uso de otra herramienta online, que comprueba entre otras muchas cosas, la velocidad de carga de la página. Esta herramienta como ya se comentó, se llama GTMetrix, y hace uso a su vez de otras dos herramientas como son Google Page Speed ⁷y Yahoo YSlow⁸.

Los resultados obtenidos son:

Para la página de inicio y el apartado de Page Speed:

⁷ <https://developers.google.com/speed/pagespeed/?csw=1>

⁸ <http://yslow.org/>

GTmetrix [Login](#) | [Sign Up](#)

Home Features Recommendations Top 1000 FAQ API Locations Contact GTmetrix PRO

Latest Performance Report for: <http://www.androiduianalyzer.com/> [Download PDF](#)

Report generated: Sun, May 31, 2015, 4:25 AM -0700
 Test Server Region: Vancouver, Canada
 Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.8

Summary

Page Speed Grade: **A** (95%)[↑] YSlow Grade: **B** (88%)[↑] Page load time: 1.37s
 Total page size: 422KB
 Total number of requests: 18

Breakdown

Page Speed | YSlow | Timeline | History

RECOMMENDATION	GRADE	TYPE	PRIORITY
Inline small CSS	B (80)	↓ CSS	High
Avoid a character set in the meta tag	B (85)	↓ Content	High
Avoid CSS @import	B (85)	↓ CSS	Medium
Leverage browser caching	B (88)	↑ Server	High
Specify a cache validator	B (89)	↓ Server	High
Defer parsing of JavaScript	A (91)	↑ JS	High
Optimize images	A (95)	↑ Images	High
Serve scaled images	A (95)	↑ Images	High
Minify HTML	A (95)	↓ Content	High
Specify a character set early	A (95)	↓ Content	High
Specify image dimensions	A (95)	↑ Images	High
Minify CSS	A (99)	↑ CSS	High
Minify JavaScript	A (99)	↑ JS	High
Avoid bad requests	A (100)	↓ Content	High
Avoid landing page redirects	A (100)	↓ Server	High
Enable gzip compression	A (100)	↑ Server	High
Enable Keep-Alive	A (100)	↓ Server	High
Inline small JavaScript	A (100)	↓ JS	High
Minimize redirects	A (100)	↑ Content	High
Minimize request size	A (100)	↓ Content	High
Optimize the order of styles and scripts	A (100)	↑ CSS/JS	High
Put CSS in the document head	A (100)	↓ CSS	High
Remove query strings from static resources	A (100)	↑ Content	High
Serve resources from a consistent URL	A (100)	↓ Content	High
Specify a Vary: Accept-Encoding header	A (100)	↑ Server	High
Combine images using CSS sprites	A (100)	↑ Images	Medium
Prefer asynchronous resources	A (100)	↓ JS	Medium

I got my scores; what now?
 Start optimizing your site! But before you do:
 > Understand the recommendations. They are meant to be generic, best practices; not everything will apply to your site.
 > Rules are sorted in order of impact upon score. Optimizing rules at the top of the list can greatly improve your overall score.
 Read our [FAQ](#) for more info.

Need help?
 Gossamer Threads can handle all of your speed optimization requirements.
 Feel free to [contact us](#) if you have any questions.

Contact | Terms of Service | © 1995-2015 Gossamer Threads, Inc.

Ilustración 37: Resultados de rendimiento de Google Page Speed

La gran mayoría de los apartados están sumamente optimizados y la página carga a una gran velocidad de 1,37 segundos.

El apartado de YSlow muestra lo siguiente:

GTmetrix Login | Sign Up

Home Features Recommendations Top 1000 FAQ API Locations Contact GTmetrix PRO

Latest Performance Report for: <http://www.androiduianalyzer.com/> Download PDF

Report generated: Sun, May 31, 2015, 4:25 AM -0700
 Test Server Region: Vancouver, Canada
 Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.8

Summary

Page Speed Grade: **A** (95%)[†] YSlow Grade: **B** (88%)[†] Page load time: 1.37s
 Total page size: 422KB
 Total number of requests: 18

Breakdown

Page Speed YSlow Timeline History

RECOMMENDATION	GRADE	TYPE	PRIORITY
Use a Content Delivery Network (CDN)	F (20)	Server	Medium
Add Expires headers	D (87)	Server	High
Make fewer HTTP requests	F (72)	Content	High
Minify JavaScript and CSS	D (60)	CSS/JS	Medium
Use cookie-free domains	D (60)	Cookie	Low
Reduce DNS lookups	A (6)	Content	Low
Avoid empty src or href	A (100)	Content	High
Compress components with gzip	A (100)	Server	High
Avoid URL redirects	A (100)	Content	Medium
Make AJAX cacheable	A (100)	JS	Medium
Put CSS at the top	A (100)	CSS	Medium
Remove duplicate JavaScript and CSS	A (100)	CSS/JS	Medium
Put JavaScript at bottom	A (100)	JS	Medium
Avoid AlphasLoader filter	A (100)	CSS	Medium
Avoid HTTP 404 (Not Found) error	A (100)	Content	Medium
Reduce the number of DOM elements	A (100)	Content	Low
Do not scale images in HTML	A (100)	Images	Low
Use GET for AJAX requests	A (100)	JS	Low
Avoid CSS expressions	A (100)	CSS	Low
Reduce cookie size	A (100)	Cookie	Low
Make favicon small and cacheable	A (100)	Images	Low
Configure entity tags (ETags)	A (100)	Server	Low
Make JavaScript and CSS external	(n/a)	CSS/JS	Medium

I got my scores; what now?
 Start optimizing your site! But before you do:
 > Understand the recommendations. They are meant to be generic, best practices; not everything will apply to your site.
 > Rules are sorted in order of impact upon score. Optimizing rules at the top of the list can greatly improve your overall score.
 Read our [FAQ](#) for more info.

Need help?
 Gossamer Threads can handle all of your speed optimization requirements.
 Feel free to [contact us](#) if you have any questions.

Contact | Terms of Service | © 1995-2015 Gossamer Threads, Inc.

Ilustración 38: Resultados de rendimiento de YSlow

En esta ocasión, la herramienta propone varios cambios de los cuales los más importantes se explican a continuación:

- Use a Content Delivery Network (CDN⁹): este aspecto no se ha llevado a cabo ya que supone un coste adicional que no se ha planteado ni reflejado en el presupuesto y se ha considerado innecesario para este proyecto. El uso de un CDN podría ser de gran

⁹ Tal y como refleja la Wikipedia, un CDN “es una red superpuesta de computadoras que contienen copias de datos, colocados en varios puntos de una red con el fin de maximizar el ancho de banda”.

utilidad si el proyecto fuese mucho mayor y utilizase una cantidad de recursos externos mucho mayor.

- Minify Javascript and CSS: esta sugerencia viene porque algunos recursos externos no están minificados. Pero esto no se puede solucionar a menos que lo hagan los propietarios de dichos recursos. Alguno de ellos es por ejemplo, las librerías de Google Analytics.
- Use cookie-free domains: al igual que la de usar un CDN, esta sugerencia se basa en servir todo el contenido estático desde un servidor adicional. Esto también incrementaría el coste del proyecto y al igual que se ha dicho previamente, se ha considerado innecesario.

El informe que genera esta herramienta también permite ver el timeline de carga de la web. Este, se muestra a continuación:

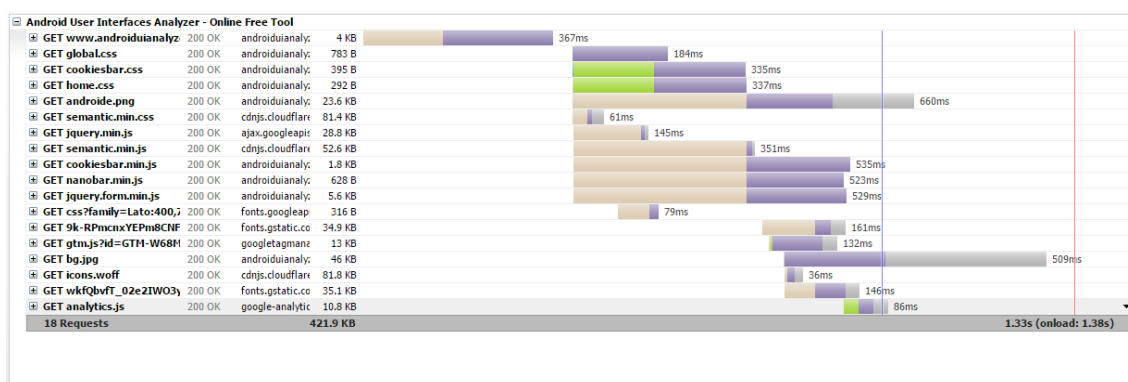


Ilustración 39: Timeline de carga de la página

Como conclusión general, se puede decir que la página tiene un tiempo de carga muy bajo, y está bastante optimizada.

Capítulo 8. Manuales del Sistema

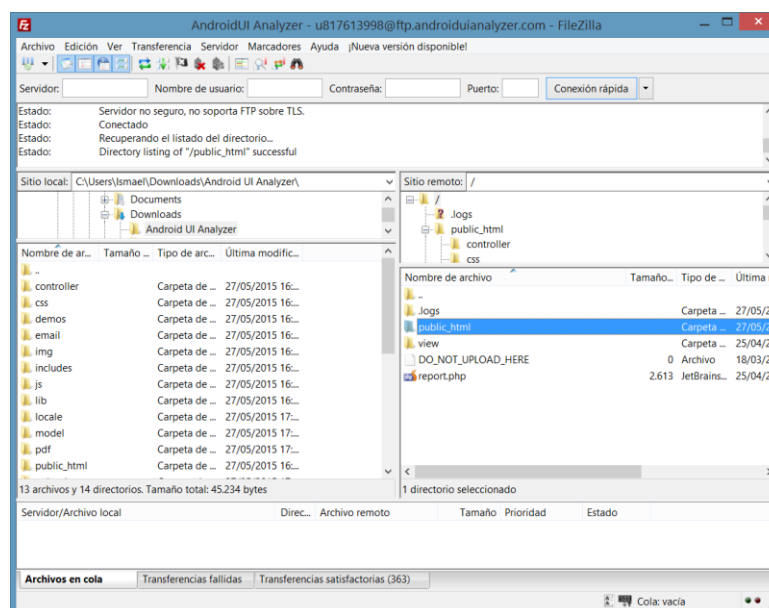
8.1 Manual de Instalación

Para la instalación de este proyecto se pueden seguir dos vías. Ya que sea desarrollado en un entorno semi-profesional con un servidor de hosting y bajo un dominio en concreto, a continuación se explica este primer método. El segundo, consistiría en desplegarlo todo en *localhost*.

8.1.1 Instalación en un servicio de Hosting

Este proceso es relativamente sencillo y simple. Para llevar a cabo esta instalación, lo primero será contratar un servicio de Hosting y disponer de los datos de acceso, típicamente mediante FTP.

Posteriormente, se deben subir todos los ficheros adjuntados con esta documentación al servidor usando un cliente de FTP como Filezilla, tal y como se muestra en la siguiente imagen:

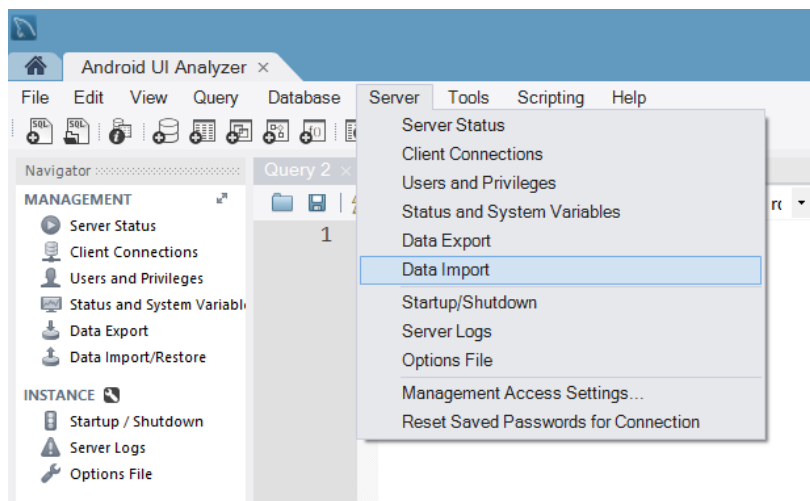


Una vez subidos todos los ficheros se debe configurar la base de datos y la conexión a la misma. Para ello y, una vez obtenidos los datos de acceso a dicha BD, se deben editar en el fichero Report.php, como se muestra a continuación:

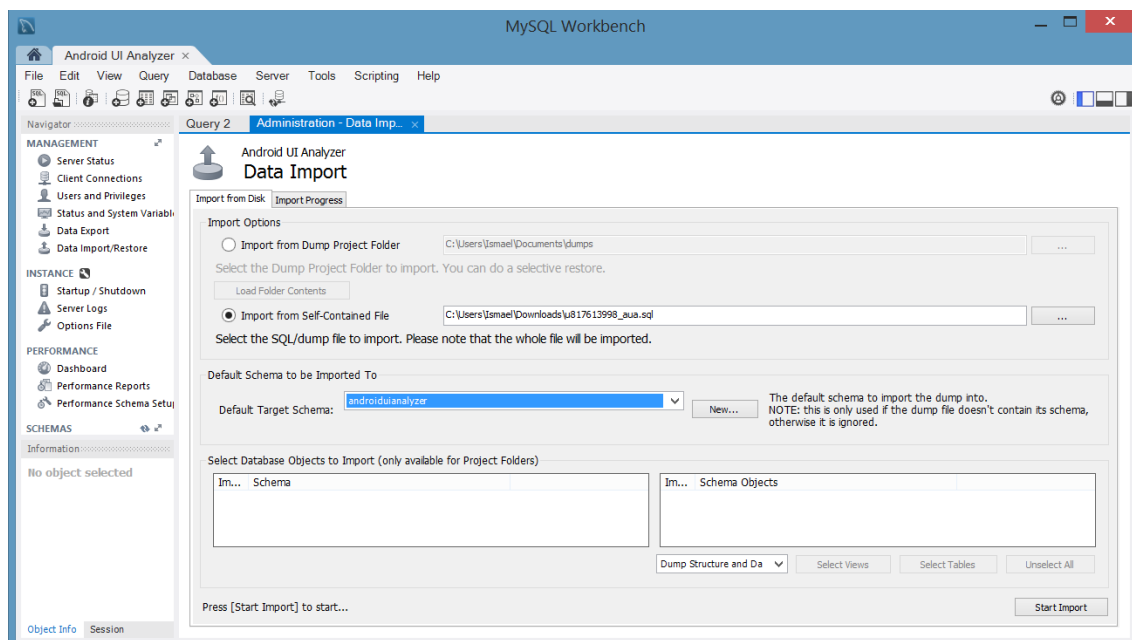
```
public function __construct ()
{
    try {
        $this->db = new PDO('mysql:host=mysql.hostinger.es;dbname=u817613998_aua', 'u817613998_isma', 'XXXXXX');
    } catch (PDOException $e) {
        $this->db = null;
        die ();
    }
}
```

Por otra parte se debe importar la base de datos mediante el fichero SQL que también se dispone junto con los demás archivos. El proceso de importación de dicha base de datos, puede variar según la herramienta que se use. En este caso se muestra el proceso para la herramienta MySQL Workbench.

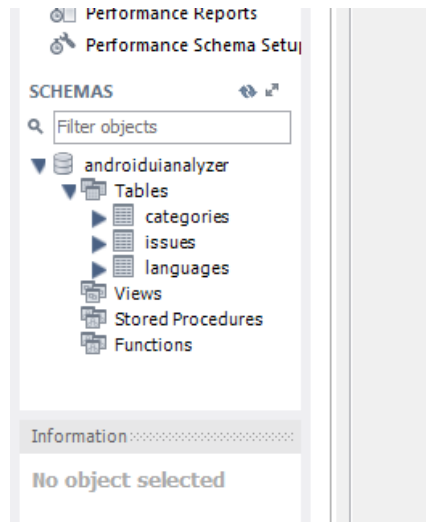
Una vez abierta la conexión con la base de datos, elegimos desde la pestaña Server, Data Import:



Acto seguido, se abrirá un asistente para comenzar la importación de las tablas y los datos necesarios. Hay que elegir la segunda opción y seleccionar el fichero sql que contiene la base de datos completa. Una vez hecho esto, también hay que escoger la base de datos a la que se quiere importar todo, en este caso androiduianalyzer y pulsar en Start Import:



Una vez hecho esto, la base de datos estará funcionando:



Para que la página comience a funcionar sin problemas, solo queda un paso, y es que hay que modificar el fichero .htaccess y cambiar las partes donde aparezca el dominio por el que se haya escogido si es distinto de www.androiduianalyzer.com:



MIW

Esta herramienta está desarrollada como trabajo fin de máster para la Universidad de Oviedo, más concretamente para el Máster de Ingeniería Web.



Código

Con esta herramienta, podrás analizar cualquier **interfaz de usuario en Android** y, asegurarte de cumplir las [buenas prácticas](#) que Google recomienda.



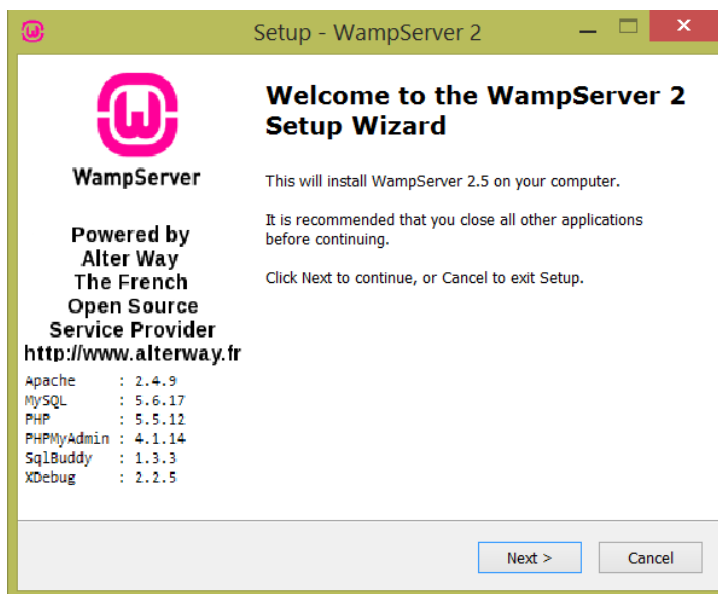
Informes

Elabora un informe con los aspectos a mejorar de las **interfaces** que estás diseñando. Esta **herramienta** está dirigida a desarrolladores Android.

8.1.2 Instalación en localhost

Para poder desplegar la web en localhost, es necesario hacer uso de una herramienta que proporcione PHP y MySQL. En este caso se verán herramientas para Windows como son WAMP Server o XAMPP. Cualquiera de estas dos opciones es perfectamente válida. Para este manual, se utilizará WAMP Server en su versión de 64bits.

Lo primero será instalar la propia herramienta. Para ello solo hace falta seguir el asistente sin ningún paso que merezca la pena mencionar, más allá que pulsar en Siguiente hasta terminar:



Una vez instalado iniciaremos el servidor haciendo doble clic sobre el icono creado. A continuación hay que copiar todos los archivos a la carpeta situada en “C:\wamp\www”, si se desea tener la página funcionando en la raíz de localhost.

Una vez hecho esto, los pasos para instalar la base de datos son completamente análogos a los vistos anteriormente, a excepción de escoger la base de datos que en este caso será localhost.

Por último también se debe cambiar el dominio que aparece en el fichero .htaccess. Una vez hecho esto la web ya estará disponible, puesto que los módulos necesarios ya estarán activados en la configuración de Apache y PHP.

8.2 Manual de Usuario

En este manual se explican las características y el uso principal de la herramienta. Además se establece el uso y la manera de proceder a la hora de utilizar la aplicación.

8.2.1 Página de inicio



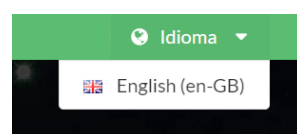
The screenshot shows the homepage of the Android UI Analyzer tool. At the top, there is a green navigation bar with 'Android UI Analyzer' and 'Contacto' on the left, and 'Idioma' with a dropdown arrow on the right. The main content area has a dark background with the title 'Herramienta de análisis de Interfaces de Usuario Android'. Below the title, it says 'Con esta herramienta podrás analizar la accesibilidad y usabilidad de cualquier interfaz en Android.' There is a file selection input field with the text 'Seleccionar archivo' and 'Ningún archivo seleccionado', and a green button labeled 'Analizar Fichero XML'. To the right of the text is a large green Android robot icon. Below this, there are three columns of information: 1. 'MIW' with a graduation cap icon, stating the tool was developed for a master's thesis at the University of Oviedo. 2. 'Código' with a code icon, stating the tool helps analyze user interfaces in Android to ensure compliance with Google's best practices. 3. 'Informes' with a document icon, stating it helps improve interfaces by generating reports. Each column has a blue button: '+ info' for MIW, and 'Descargar demos' for Informes. At the bottom, there is a dark footer with 'Todos los derechos reservados ©. 2015' and 'Android UI Analyzer' on the left, and 'Proyecto Fin Máster' and 'Desarrollado por Ismael Ramos' on the right. There are also links for 'Aviso legal - Política de privacidad' and 'Herramienta gratuita'.

Ilustración 40: Página de inicio

Desde esta página el usuario puede seleccionar un fichero XML que contenga una interfaz Android y pulsar en Analiza XML.

La página también incluye secciones para explicar dónde y porqué se ha desarrollado, una descripción básica de para qué sirve y un botón para descargar ejemplos de interfaces.

Todas las páginas disponen en la parte superior de un menú con el resto de enlaces a otras páginas y la opción para cambiar el idioma.



8.2.2 Página del informe

Informe - activity_main.xml

Mostrar todos Analizar de nuevo Crear PDF

Soportando Múltiples Tamaños de Pantalla

Estado	Elementos afectados	Nombre	Descripción	Solución
✘ Fallo	TextView (id:textView1), android.support.v7.widget.CardView (id:card_view), ImageView (id:imageView2), Button (id:button2)	Uso de "wrap_content" y "match_parent"	Para asegurarse de que el diseño es flexible y se adapta a diferentes tamaños de pantalla, se debe utilizar "wrap_content" y "match_parent" para el ancho y la altura de los componentes.	Usa "wrap_content" y "match_parent" en vez de valores numéricos para definir tamaños. Saber más

Soportando Diferentes Densidades

Estado	Elementos afectados	Nombre	Descripción	Solución
✘ Fallo	Button (id:button1), Button (id:button2)	Utiliza píxeles independientes de la densidad	Definir dimensiones con píxeles es un problema porque existen diferentes pantallas que tienen diferentes densidades de píxeles, por lo que el mismo número de píxeles puede corresponder a diferentes tamaños físicos en diferentes dispositivos.	Al especificar las dimensiones, usa siempre "dp" o "sp" nunca "px". Saber más

45% **Sigue trabajando!**

Android User Interfaces Analyzer - Herramienta gratuita Proyecto Fin Máster
 Aviso legal - Política de privacidad Desarrollado por [Ismael Ramos](#)

Ilustración 41: Página del informe

Esta página es la que se encarga de mostrar el informe. En la parte central se encuentran las distintas tablas con los fallos y encima de cada una la categoría a la que pertenecen.

Cada tabla tiene una cabecera que explica las diferentes columnas. Si se pulsa sobre Saber más, se abrirá una nueva pestaña en el navegador para conocer más información sobre ese fallo en concreto.

Al comienzo del informe, existen tres opciones, para Mostrar los elementos que son correctos, volver a inicio o crear el PDF.

Por otra parte, al final del mismo, se dispone de una barra de progreso con mensajes que informan del estado actual de la interfaz.

8.2.3 Página de contacto

Android UI Analyzer | Contacto | Idioma

Contacta con nosotros

Si tienes alguna duda o quieres proponer alguna idea puedes contactar a través del siguiente formulario.

Datos personales

Nombre completo * | Email *

Nombre completo | Email válido

Consúltanos lo que quieras

Asunto * | Mensaje *

Asunto que quieras tratar | Escribe lo que quieras. Puedes incluso darnos tus opiniones!

Completa el captcha *

No soy un robot | reCAPTCHA | Privacidad - Condiciones

Acepto los términos y condiciones del [Aviso legal](#) y la [Política de privacidad](#)

Enviar

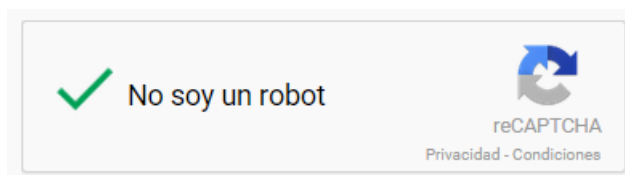
Todos los derechos reservados ©. 2015 | Android UI Analyzer | Proyecto Fin Máster | Desarrollado por [Ismael Ramos](#)

Android User Interfaces Analyzer - Herramienta gratuita | [Aviso legal](#) - [Política de privacidad](#)

Ilustración 42: Página de contacto

Esta es la página mediante la cual el usuario puede ponerse en contacto y resolver cualquier duda o proponer alguna mejora.

El formulario consta de 6 campos obligatorios. Uno de ellos se trata de un Captcha para evitar spam, que se debe resolver antes de enviar el mensaje:



En la parte superior, se ha implementado una barra de progreso que avanza a medida que se va completando el formulario, o retrocede si surge algún problema a posteriori.

8.3 Manual del Programador

8.3.1 Requisitos hardware

Para poder trabajar con fluidez se recomienda un ordenador con las siguientes características:

- Procesador Core 2 Duo a 2GHz o equivalente.
- 2Gb de Memoria RAM.

Además, cualquier configuración hardware que permita tener instalado el servidor para PHP y MySQL (en caso de trabajar en local) servirá sin ningún problema.

8.3.2 Recursos humanos

La persona encargada de mantener este proyecto o realizar modificaciones tendrá que tener las siguientes cualidades o competencias:

- Conocimiento de un lenguaje de programación orientado a objetos. Al menos C++ y en mejor posición C#, Python o Java. Por supuesto el ideal sería PHP.
- Haber desarrollado alguna vez para la web o tener conocimientos medios sobre ello.
- Además de los anteriores, el futuro desarrollador tendrá que conocer patrones de diseño o al menos saber qué son y para qué se utilizan. De esta manera, podrá identificarlos y modificar mejor el código.
- Tener nociones de Refactorización.
- Conocimientos básicos de Android y XML.

8.3.3 Desarrollo futuro

En esta sección se explicarán los pasos básicos para desarrollar posibles ampliaciones como añadir idiomas o detectar nuevos fallos.

8.3.3.1 *Nuevos idiomas*

La inclusión de un nuevo idioma será relativamente sencilla. El primer paso podría ser añadir el idioma a la tabla “languages” de la base de datos, y traducir los distintos “issues” y “categories” a ese nuevo idioma.

Lo siguiente será crear el fichero de localización como ya existen “es.php” y “en.php” con las cadenas traducidas.

Una vez que la base de datos esté preparada y las traducciones de las páginas también, se deberá modificar el archivo “language.php” para que cargue el fichero de localización nuevo en caso de que el usuario lo seleccione:


```

switch ($lang) {
    case "es" :
        include ("locale/es.php");
        break;
    default :
        include ("locale/en.php");
}

```

Por último solo faltaría añadir la opción de ese nuevo idioma al menú, para que el usuario pueda escogerlo.

8.3.3.2 Detectar nuevos fallos

El hecho de añadir un nuevo fallo a detectar también es relativamente fácil. Los pasos se podrían resumir:

1. Crear la categoría necesaria si no vale ninguna de las ya existentes.
2. Crear la nueva entrada en la base de datos con los datos para el nuevo fallo que se quiere detectar. Esto es, nombre, descripción, solución, enlace de saber más y si necesita revisión por parte del usuario.
3. Determinar si necesita recursión o no y crear un nuevo método privado en ReportsController con la programación necesaria para detectar el fallo.
4. Añadir la llamada a ese nuevo método.

8.3.3.3 Modificaciones en la interfaz

En el capítulo de Aspectos teóricos ya se mostró algún ejemplo sobre el framework de CSS que se ha usado (SemanticUI). No obstante si se quisiese modificar la interfaz, se pueden tomar fragmentos disponibles de la web, y que ya vienen generados.

La inclusión, de botones, menús, fichas, listas, tablas o cualquier otro elemento conocido es muy sencilla. Por ejemplo, se podría añadir una barra lateral simplemente añadiendo el código HTML:

```

<div class="ui left vertical inverted labeled icon sidebar menu">
  <a class="item">
    <i class="home icon"></i>
    Home
  </a>
  <a class="item">
    <i class="block layout icon"></i>
    Topics
  </a>
  <a class="item">
    <i class="smile icon"></i>
    Friends
  </a>
</div>

```

Y el código Javascript:

```

$('.left.demo.sidebar')
  .sidebar('toggle')
;

```

Capítulo 9. Conclusiones y Ampliaciones

9.1 Conclusiones

El principal objetivo de este proyecto, era disponer de una herramienta online que fuese capaz de analizar interfaces Android, y que según las prácticas que recomienda Google, se generase un informe con los posibles fallos, aciertos y recomendaciones. Este objetivo ha sido cumplido con satisfacción. Además se han buscado una serie de funcionalidades adicionales que facilitan al usuario la interacción con el informe y con la web en general.

Durante todo el desarrollo del proyecto, he profundizado y adquirido más conocimientos en conceptos como posicionamiento en buscadores, el lenguaje de programación PHP, administración y diseño de bases de datos, así como el uso de frameworks de CSS y de herramientas como Google Analytics.

Cabe mencionar, que los resultados obtenidos en el cuestionario han sido bastante positivos siendo muy útiles para realizar pequeñas mejoras en el diseño del informe como ya se ha explicado.

9.2 Ampliaciones

Entre las posibles ampliaciones, cabría mencionar la siguiente lista:

- Incorporación de más opciones a detectar en las interfaces, a medida que aparezcan nuevas recomendaciones.
- Nuevas opciones de administración y configuración del informe, como intentar reanalizar el archivo con solo pulsar un botón.
- Incorporación de nuevos idiomas.

Estas ampliaciones se pueden implementar cuando el sistema se quede anticuado o si las guías y recomendaciones de Google sufren cambios significativos.

Capítulo 10. Referencias Bibliográficas

10.1 Libros y Artículos

[Lockhart15] Lockhart, Josh. “Modern PHP: New Features and Good Practices”. O’Reilly. 2015. ISBN-13: 978-1491905012.

[Duckett14] Duckett, Jon. “JavaScript and JQuery: Interactive Front-End Web Development”. Wiley. 2014. ISBN-13: 978-1118531648.

10.2 Referencias en Internet

[SemanticUI15] Semantic-Org “SemanticUI”. <http://semantic-ui.com>. 2015.

[jQuery15] jQuery “jQuery Form Plugin ”. <http://jquery.malsup.com/form/#ajaxSubmit>. 2015.

[Hassan08] Hassan Montero, Y. “Guía de Evaluación Heurística de Sitios Web”. <http://www.nosolousabilidad.com/articulos/heuristica.htm>. 2015.

[PHP15] PHP, “Manual de PHP sobre SimpleXML”. <http://php.net/manual/es/book.simplexml.php>. 2015.

[WorkStudio15] Work Studio INC, “Basic Example using SMTP”. <http://phpmailer.worxware.com/?pg=examplebsmtp>. 2015.

[Google15] Google INC, “Best practices in Android”. <http://developer.android.com/guide/practices/index.html>. 2015

[Edo15] Mauri Edo, “¿Testing funcional o pruebas de caja negra?”. <https://testingfuncional.wordpress.com/2011/03/12/%C2%BFtesting-funcional-o-pruebas-de-caja-negra/>. 2015

[Google15] Google INC, “Guía para principiantes sobre optimización para motores de búsqueda”. https://static.googleusercontent.com/media/www.google.com/es//intl/es/webmasters/docs/guia_optimizacion_motores_busqueda.pdf. 2015

Capítulo 11. Anexos

Los anexos que contiene esta documentación son:

- Anexo 1 – Planificación: contiene información detallada de la planificación.
- Anexo 2 – Presupuesto: contiene el presupuesto de costes y el seguimiento del mismo.
- Anexo 3 – Cuestionarios: información relativa a los cuestionarios que se han realizado para las pruebas de usabilidad.
- Anexo 4 - Código fuente: todo el código que se ha implementado y que conforma el proyecto.