

UNIVERSIDAD DEL VALLE DE GUATEMALA

**Facultad de Ingeniería
Departamento de Computación**



**Desarrollo de plataformas de comunicación y recolección de
datos sobre desastres para la Coordinadora Nacional para la
Reducción de Desastres. Módulo: Aplicación para celulares.**

Jessica Alejandra Canahui Morales

**Guatemala
2013**

Desarrollo de plataformas de comunicación y recolección de datos sobre desastres para la Coordinadora Nacional para la Reducción de Desastres. Módulo: Aplicación para celulares.

UNIVERSIDAD DEL VALLE DE GUATEMALA

**Facultad de Ingeniería
Departamento de Computación**



**Desarrollo de plataformas de comunicación y recolección de
datos sobre desastres para la Coordinadora Nacional para la
Reducción de Desastres. Módulo: Aplicación para celulares.**

Trabajo de Graduación presentado por Jessica Alejandra Canahui
Morales para optar al grado académico de Licenciada en Ingeniería
de Ciencias de la Computación.

**Guatemala
2013**

Vo. Bo.:

(f) _____
Tomás Gálvez Peña

Tribunal:

(f) _____
Douglas Leonel Barrios González

(f) _____
Tomás Gálvez Peña

(f) _____
Rodrigo Arias Palomo

Fecha de aprobación:

PREFACIO

El siguiente informe de Megaproyecto describe el proceso de planteamiento, diseño e implementación de una herramienta para teléfonos móviles inteligentes que disminuya el tiempo de transferencia de datos tomados en poblaciones afectadas por desastres naturales con el propósito de agilizar el proceso de dar respuesta a dichos desastres desde la Sede Central de la Coordinadora Nacional de Reducción de Desastres (CONRED).

El Megaproyecto se dividió en tres fases. La primera fase consistió en la elaboración de un documento de visión, en el cual se indicaron los objetivos, necesidades, requerimientos y riesgos del proyecto; para la elaboración de dicho documento fue necesario realizar investigación sobre sistemas similares y recopilar información sobre las necesidades y requerimientos de CONRED. Esta fase se llevó a cabo durante el segundo semestre del ciclo escolar 2012.

La segunda fase consistió en diseñar la arquitectura tanto de la aplicación Android como de la página web. En esta fase se comenzó con el desarrollo de ambas soluciones. Se evaluaron todas las características del contexto de uso y de los usuarios finales, de manera que fueran tomadas en cuenta en el diseño final de las soluciones. Esta fase se llevó a cabo durante el primer semestre del ciclo escolar 2013, y se contó con el apoyo del departamento de informática y de los delegados de CONRED.

La tercera fase del proyecto consistió en la finalización del desarrollo de los sistemas, así como de las pruebas de control de calidad y de usabilidad. Durante esta fase se hicieron entregas de varias versiones de ambos sistemas, se realizaron capacitaciones sobre el uso de la aplicación Android, y se realizaron pruebas de usabilidad de ambas soluciones.

ÍNDICE

PREFACIO	VI
ÍNDICE.....	VII
LISTA DE CUADROS.....	X
LISTA DE ILUSTRACIONES	XI
I. INTRODUCCIÓN	12
A. Objetivos.....	12
1. General del Megaproyecto.....	12
2. General del Módulo	12
3. Específicos del Módulo	12
B. Justificación	12
II. MARCO TEÓRICO.....	14
A. Sistema Operativo Android.....	14
B. PhoneGap	15
C. Tecnologías web	16
1. HTML	16
a. Etiquetas.....	17
b. Nombres.....	17
c. Atributos	18
d. HTML5	18
2. PHP	18
3. JavaScript.....	19
a. jQuery	19
b. JSON.....	19

c. JSON2HTML.....	22
1) Transformaciones	23
2) Funcionamiento	23
4. CSS	23
III. METODOLOGÍA	25
A. Obtención de requerimientos	25
B. Elaboración de Casos de Uso	25
C. Creación de prototipo.....	25
D. Investigación y Diseño.....	25
E. Implementación	25
IV. DISEÑO.....	26
A. Requerimientos y Casos de Uso	26
1. Objetivos.....	26
2. Criterios de Éxito	26
3. Características Principales	26
B. Arquitectura de la aplicación	27
1. Plataforma de desarrollo móvil.....	27
2. Diseño del motor de la aplicación.....	28
3. Sistema en la base de datos central	30
4. Diagrama general del sistema y sus componentes	31
5. Diseños de las bases de datos	32
a. SQLite	32
b. MySQL	32
B. Características de la aplicación	33

1. Ingreso de datos nuevos	33
2. Guardado de información recolectada	33
3. Envío de información recolectada	33
4. Modificación de datos previamente guardados.....	34
5. Mantenimiento	34
6. Extensibilidad	35
V. RESULTADOS Y DISCUSIÓN	36
A. Pantalla Inicial	36
B. Nuevo Formulario	37
C. Formularios Guardados.....	37
D. Ingreso de datos	38
E. jhEngine	40
F. Servicio para envíos pendientes	44
G. Procesos del servidor de la base de datos central.....	46
VI. CONCLUSIONES	49
VII. RECOMENDACIONES.....	50
VIII. Bibliografía	51
IX. APÉNDICE.....	53
A. Formulario de Evaluación de Daños y Análisis de Necesidades (EDÁN) para Android ..	53
B. Formulario de Evaluación Preliminar de Situación	54
C. Formulario de Evaluación Preliminar de Habitabilidad.....	57
D. Formulario de Evaluación Rápida de Daño en Edificaciones.....	60
E. Evaluación Rápida de Centros Educativos	61
X. GLOSARIO	62

LISTA DE CUADROS

Tabla 1. Características soportadas por PhoneGap en las principales plataformas móviles.....	15
--	----

LISTA DE ILUSTRACIONES

Figura 1. Ejemplo de marcado HTML.....	17
Figura 2. Resultado visualizado del marcado HTML de ejemplo	17
Figura 3. Sintaxis de un objeto JSON.....	20
Figura 4. Sintaxis de un arreglo JSON	20
Figura 5. Sintaxis de un valor JSON.....	21
Figura 6. Sintaxis de una cadena JSON.....	21
Figura 7. Sintaxis de un valor numérico JSON.....	22
Figura 8. Diagrama general de la arquitectura de la aplicación.....	31
Figura 9. Diseño de la base de datos SQLite local para uso de la aplicación.....	32
Figura 10. Diseño de la bse de datos MySQL para almacenaje de la información en la sede central.....	33
Figura 11. Pantalla inicial de la aplicación.	36
Figura 12. Pantalla para iniciar a llenar un formulario en blanco.....	37
Figura 13. Pantalla de carga de un formulario previamente llenado y guardado en la base de datos del teléfono.....	38
Figura 14. Diagrama de flujo que describe el proceso de guardado/envío de un formulario lleno o parcialmente lleno.....	39
Figura 15. Pantalla de ingreso de datos de la aplicación.	40
Figura 16. Diagrama de clases para el motor jhEngine.	41
Figura 17. Objeto <template> correspondiente a un formulario sencillo en jhEngine.	42
Figura 18. Despliegue obtenido a partir de la plantilla de transformación del ejemplo básico de la Figura 17.	43
Figura 20. Diagrama de llamadas entre los componentes involucrados en el inicio del servicio de envío.....	44
Figura 21. Código del archivo ping.php en el servidor central.....	46
Figura 22. Código del archivo submit.php en el servidor central	48
Figura 23. Formulario de Evaluación de Daños y Análisis de Necesidades (EDÁN) para Android	53

I. INTRODUCCIÓN

A. Objetivos

1. General del Megaproyecto

Desarrollo e implementación de plataformas que agilizan el proceso de recolección y envío de datos para la Coordinadora Nacional para la Reducción de Desastres (CONRED).

2. General del Módulo

Desarrollar una aplicación para teléfonos inteligentes que permita a los delegados departamentales de CONRED recopilar datos de desastres en el lugar donde ocurren y enviar la información recolectada a la Sede Central por medio de la conexión a Internet del teléfono en cuanto dicha conexión se haga disponible.

3. Específicos del Módulo

- Implementar un sistema que disminuya el tiempo de envío de datos desde los lugares donde se reporten desastres y la Sede Central de CONRED.
- Implementar una aplicación que maneje efectivamente cualquier suceso no esperado durante los procesos de llenado y envío de información.
- Implementar un sistema usable que facilite el llenado y envío de reportes y formularios a usuarios poco experimentados con teléfonos inteligentes.
- Implementar el sistema como una plataforma que facilite al personal del Departamento de Sistemas de CONRED el mantenimiento e implementación de formularios y funcionalidades adicionales.

B. Justificación

El territorio de Guatemala, debido a su posición geográfica, geológica y tectónica, está clasificado como uno de los países a nivel mundial con un alto potencial de múltiples amenazas naturales. Su situación social, económica, deterioro ambiental y de desarrollo general y su alta vulnerabilidad provocan que una gran parte de la población, su infraestructura y servicios estén expuestos a diferentes riesgos, que pueden resultar en desastres.

Al ocurrir un desastre en el territorio nacional CONRED debe llenar varios formularios que reúnen información sobre la situación en las áreas afectadas. Dicha información es vital para la toma de decisiones sobre la respuesta al evento y su organización.

Actualmente los formularios se llenan a mano en el lugar del desastre y al llegar el delegado a una sede, la información se transcribe y se envía a la sede central.

Este método puede tomar más tiempo de lo que se consideraría conveniente, por lo que se retrasa el proceso de respuesta al desastre. El nuevo sistema presenta varias ventajas. En primer lugar, no habrá que transcribir el contenido, ya que se tomará directamente en el teléfono. También se podrá enviar la información desde cualquier lugar donde el teléfono pueda conectarse a internet, lo cual ya no hace necesario el tener que estar en la sede departamental, y hace posible el envío directamente desde el sitio del evento.

II. MARCO TEÓRICO

A. Sistema Operativo Android

“Android es un sistema operativo para dispositivos móviles como teléfonos inteligentes y tabletas basado en el núcleo Linux. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google, usando diversos conjuntos de herramientas de software de código abierto para dispositivos móviles.

Android implementa una arquitectura en la que cualquier aplicación puede obtener acceso a las capacidades del dispositivo móvil. Por ejemplo, una aplicación puede llamar una o varias de las funcionalidades básicas de dichos dispositivos, tales como realizar llamadas, enviar mensajes de texto, o utilizar la cámara, facilitando a los desarrolladores crear experiencias más ricas y con más coherencia para los usuarios.” (Molina Rivera, Sandoval Cardona, & Toledo Franco, 2012)

“La base de la arquitectura Android es el kernel Linux 2.6, el cual soporta seguridad, gestión de la memoria, gestión de procesos, pila de red y modelo de controlador de dispositivo.

Un conjunto de librerías C/C++ es utilizado por varios componentes del sistema Android. Consisten de la librería estándar C (libc), librerías multimedia incluyendo MPEG4, H.264, MP3, JPG y PNG, LibWebCore como motor de navegación web, motor gráfico 2D SGL, librerías de gráficos 3D, FreeType para despliegue de fuentes y SQLite, un motor ligero de bases de datos relacionales.

En tiempo de ejecución, Android incluye un conjunto de librerías básicas que proveen la funcionalidad del lenguaje de programación Java. La máquina virtual Dalvik provee un ambiente de ejecución para aplicaciones Android en Java. Cada aplicación de Android corre en su propio proceso, con su propia instancia de la máquina virtual.

Los desarrolladores de aplicaciones usualmente acceden a la capa del marco de trabajo de aplicación a través de varios APIs, mientras desarrollan programas con Java y XML. El marco de trabajo de aplicación es una plataforma de desarrollo de software libre que incluye administración de vistas, proveedores de contenido, administradores de recursos, gestores de notificaciones y gestores de actividades. Muchos componentes reutilizables son publicados en el marco de trabajo, y el desarrollador puede reemplazar los componentes.” (Yoon, 2012)

B. PhoneGap

“PhoneGap es un marco de trabajo de código abierto para el desarrollo de aplicaciones híbridas multi-plataforma utilizando HTML, CSS y Javascript, usando APIs de abstracción para interactuar con el hardware del dispositivo. Por eso, las aplicaciones desarrolladas utilizando PhoneGap pueden ser publicadas en múltiples plataformas con pocos o ningún cambio al código base.” (Intel Corporation, 2012)

	iPhone / iPhone 3G	iPhone 3GS y más reciente	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7+8
Acelerómetro	Si	Si	Si	Si	Si	Si	Si
Cámara	Si	Si	Si	Si	Si	Si	Si
Brújula	No	Si	Si	No	No	Si	Si
Contactos	Si	Si	Si	Si	Si	No	Si
Archivos	Si	Si	Si	Si	Si	No	Si
Geolocalización	Si	Si	Si	Si	Si	Si	Si
Multimedia	Si	Si	Si		No	No	Si
Red	Si	Si	Si	Si	Si	Si	Si
Notificaciones (Alertas)	Si	Si	Si	Si	Si	Si	Si
Notificaciones (Sonido)	Si	Si	Si	Si	Si	Si	Si
Notificaciones (Vibración)	Si	Si	Si	Si	Si	Si	Si
Almacenamiento	Si	Si	Si	Si	Si	Si	Si

	Symbian	Bada
Acelerómetro	Si	Si
Cámara	Si	Si
Brújula	No	Si
Contactos	Si	Si
Archivos	No	No
Geolocalización	Si	Si
Multimedia	No	No
Red	Si	Si
Notificaciones (Alertas)	Si	Si
Notificaciones (Sonido)	Si	Si
Notificaciones (Vibración)	Si	Si
Almacenamiento	No	No

Tabla 1. Características soportadas por PhoneGap en las principales plataformas móviles.

“El código de PhoneGap fue contribuido a la Apache Software Foundation (ASF) bajo el nombre *Apache Cordova*, y pasó a ser proyecto de máximo nivel en Octubre del 2012. El proyecto se mantendrá como libre y de código abierto bajo la Licencia Apache, Versión 2.0.” (Adobe Systems, Inc., 2013)

C. Tecnologías web

1. HTML

“HTML (Lenguaje de Marcado de HiperTexto, por sus siglas en inglés) es el lenguaje que describe la estructura de páginas web. Entre otras cosas, HTML le da a los autores los medios para:

- Publicar documentos en línea con encabezados, texto, tablas, listas, fotografías, etc.
- Obtener información en línea por medio de vínculos de hipertexto, con el click de un botón.
- Diseñar formularios para realizar transacciones con servicios remotos, para uso en búsquedas de información, reservaciones, órdenes de productos, etc.
- Incluir hojas de cálculo, clips de video y audio y otras aplicaciones directamente en sus documentos.

Con HTML, los autores describen la estructura de las páginas utilizando *marcado*. Los *elementos* del lenguaje etiquetan piezas de contenido tales como “párrafo”, “lista”, “tabla”, etc.” (World Wide Web Consortium (W3C), 2012)

“Una instancia HTML es como un archivo de texto, excepto que algunos de los caracteres son interpretados como marcado. El marcado da estructura al documento.

La instancia representa una jerarquía de elementos. Cada elemento tiene un nombre, algunos atributos y, en ciertos casos, contenido. La mayoría de los elementos son representados en el documento como una etiqueta de inicio, que da el nombre y atributos, seguido por el contenido, seguido por la etiqueta de fin.

A continuación en la Figura 1 se muestra una porción de marcado HTML, y en la Figura 2 se muestra el despliegue que dicho ejemplo tiene en un navegador web.

```
<html>
  <head>
    <title>A sample HTML instance</title>
  </head>
  <body>
    <h1>An Example of Structure</h1>
    <p>Here's a typical paragraph.</p>
    <ul>
      <li>Item one has an <a name="anchor">anchor</a></li>
      <li>Here's item two.</li>
    </ul>
  </body>
</html>
```

Figura 1. Ejemplo de marcado HTML

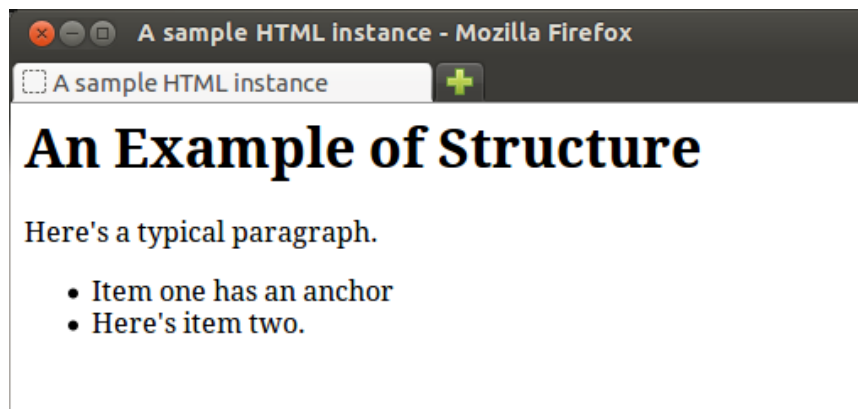


Figura 2. Resultado visualizado del marcado HTML de ejemplo

a. Etiquetas

Cada elemento empieza con una etiqueta, y cada elemento no vacío termina con una etiqueta. Las etiquetas de inicio se delimitan con “<” y “>”, y las etiquetas de fin se delimitan con “</” y “>”.

b. Nombres

El nombre de un elemento sigue inmediatamente el delimitador de apertura de una etiqueta. Los nombres consisten de una letra seguida por hasta 33 letras, dígitos, puntos o guiones. Los nombres no diferencian entre mayúsculas y minúsculas.

c. Atributos

En una etiqueta de inicio, se permite espacio en blanco y atributos entre el nombre del elemento y el delimitador de cierre. Un atributo consiste de un nombre, un signo de igual, y un valor. Se permite espacio en blanco alrededor del signo de igual.

El valor se especifica en una cadena rodeada por comillas simples o dobles. La longitud de un valor se limita a 1024 caracteres.” (Berners-Lee, 1993)

d. HTML5

HTML5 es la quinta revisión del estándar HTML. Su intención es mejorar el lenguaje con soporte para la más reciente multimedia, mientras su lectura se mantiene fácil para humanos y es consistentemente entendido por computadoras y dispositivos.

HTML5 busca no sólo reemplazar el estándar HTML4, sino también otras especificaciones como XHTML1.

En particular, HTML5 agrega nuevas características sintácticas. Éstas incluyen tanto los nuevos elementos `<video>`, `<audio>`, `<canvas>`, tanto la integración de Gráficos Vectoriales Escalables (SVG, por sus siglas en inglés) y el lenguaje de marcado MathML para fórmulas matemáticas. Estas características están diseñadas para facilitar la inclusión y manejo de contenido multimedia en la web sin tener que recurrir a extensiones y APIs que no están abiertos al público. Otros elementos nuevos, tales como `<section>`, `<article>`, `<header>` y `<nav>` están diseñados para describir semánticamente los documentos. Se han agregado nuevos atributos para el mismo propósito, mientras algunos elementos y atributos han sido removidos. HTML5 también define, con cierto nivel de detalle, el procesamiento requerido para tratar documentos con sintaxis inválida de manera uniforme en todos los navegadores compatibles y otros agentes de usuario. (World Wide Web Consortium (W3C), 2013)

2. PHP

“PHP, que significa por sus acrónimo recursivo en inglés “PHP: Hypertext Preprocessor”, es un lenguaje de scripting de código abierto, ampliamente usado y de propósito general, especialmente adecuado para desarrollo web y puede ser intercalado con HTML. Su sintaxis se basa en la de los lenguajes C, Java y Perl.

El objetivo principal del lenguaje es permitir a los desarrolladores web escribir rápidamente páginas web generadas dinámicamente en el momento de ejecución.

En vez de utilizar comandos para producir HTML, las páginas PHP contienen HTML junto con código incrustado que ejecuta instrucciones del lenguaje. El código se encuentra entre etiquetas especiales de inicio y fin (“<?php” y “?>”).” (The PHP Group, 2013)

3. JavaScript

JavaScript es un lenguaje de programación que como parte de navegadores web, sus implementaciones permiten al código del lado del cliente interactuar con el usuario, controlar el navegador, comunicarse asincrónicamente con diferentes servidores y alterar el contenido del documento que se está desplegando. JavaScript es un lenguaje de programación de alto nivel, dinámico y sin tipos que se adapta bien a estilos de programación funcional y orientado a objetos. (Flanagan, 2011)

a. jQuery

JavaScript tiene un API general intencionalmente simple, pero la implementación de dicho API en los diferentes navegadores que existen ha resultado en incompatibilidades mayores entre navegadores. Muchos programadores encuentran más fácil escribir aplicaciones web utilizando un marco de trabajo de JavaScript o librería de utilidades para simplificar tareas comunes y ocultar las diferencias entre navegadores. Actualmente, uno de las librerías más populares y utilizadas es jQuery. (Flanagan, 2011)

jQuery es una librería multi-navegador de JavaScript designada a simplificar el scripting del lado del cliente de HTML. Su sintaxis está diseñada para hacer más sencillo el proceso de navegar un documento, crear animaciones, manejar eventos y desarrollar aplicaciones Ajax. (The jQuery Foundation, 2013)

Como su nombre lo indica, jQuery se concentra en consultas (en inglés: *queries*). Una consulta típica utiliza un selector CSS para identificar un conjunto de elementos del documento y retorna un objeto que representa esos elementos. Dicho objeto provee varios métodos útiles para operar en los elementos HTML encontrados como un grupo. (Flanagan, 2011)

b. JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Su lectura y escritura es sencilla para humanos y su lectura y generación es fácil para máquinas. Se encuentra basado en un subconjunto del lenguaje de programación JavaScript.

JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son similares a la familia de lenguajes-C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros.

JSON se conforma por dos estructuras:

1. Una colección de pares valor/nombre. En varios lenguajes, esto se denomina objeto, registro, diccionario, hash table o arreglo asociativo.
2. Una lista ordenada de valores. En la mayoría de lenguajes, esto se denomina arreglo, vector, lista o secuencia.

Éstas son estructuras de datos universales. Virtualmente, todos los lenguajes de programación las soportan de una forma u otra.

En JSON, estas estructuras toman las siguientes formas:

Un **objeto** es un set no ordenado de pares de nombre/valor. Un objeto comienza con “{” (llave izquierda) y termina con “}” (llave derecha). Cada nombre es seguido por “:” (dos puntos) y los pares nombre/valor son separados por “,” (coma).

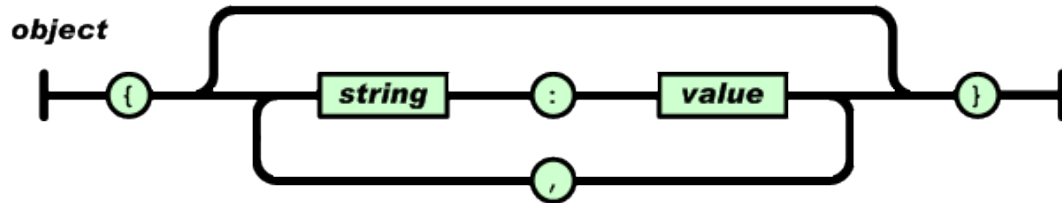


Figura 3. Sintaxis de un objeto JSON

Un **arreglo** es una colección ordenada de valores. Un arreglo empieza con “[” (corchete izquierdo) y termina con “]” (corchete derecho). Sus valores están separados por “,”.

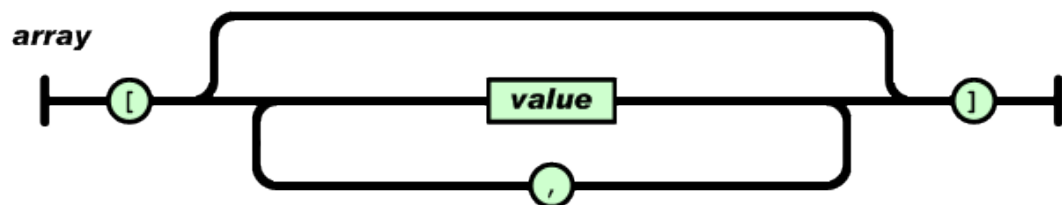


Figura 4. Sintaxis de un arreglo JSON

Un **valor** puede ser una *cadena* en comillas dobles, un número, un valor booleano, *null*, un *objeto* o un *arreglo*. Estas estructuras pueden estar anidadas.

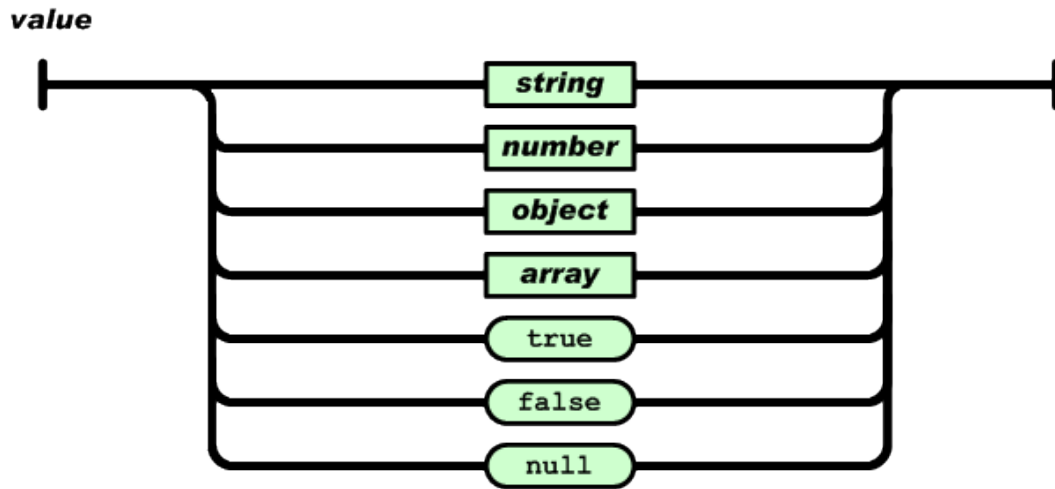


Figura 5. Sintaxis de un valor JSON

Una **cadena** es una secuencia de cero o más caracteres Unicode, envueltos en comillas dobles, utilizando caracteres de escape con diagonal invertida.

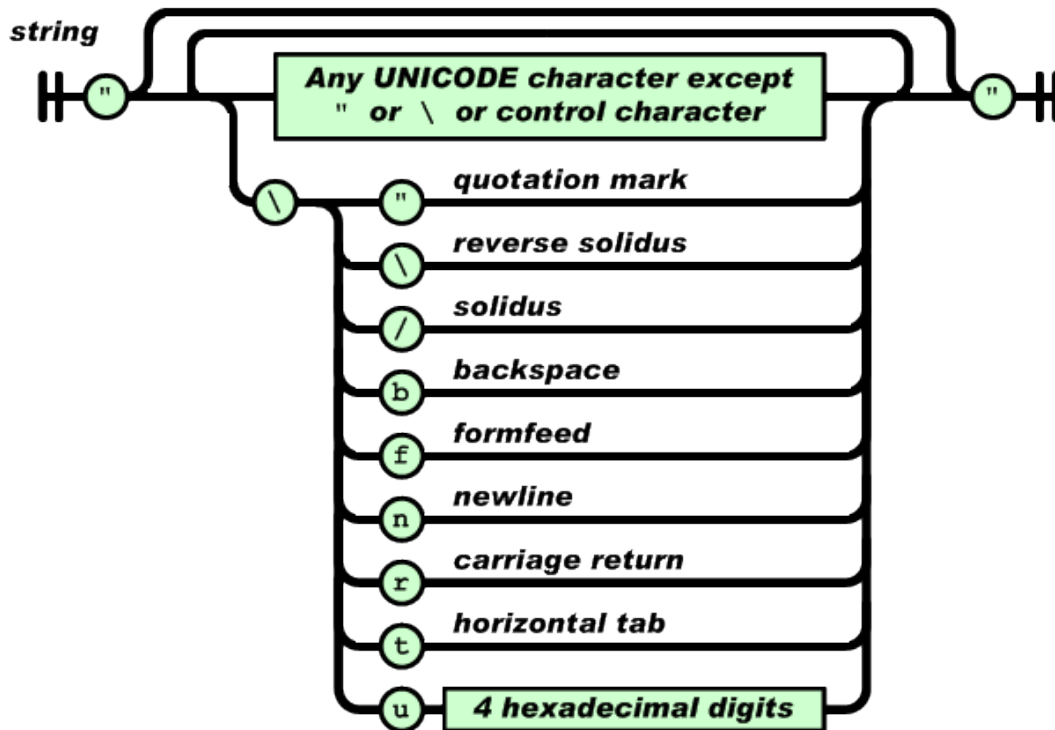


Figura 6. Sintaxis de una cadena JSON

Un número es muy similar a un número en C o Java, con la excepción que los formatos octal y hexadecimal no son utilizados.

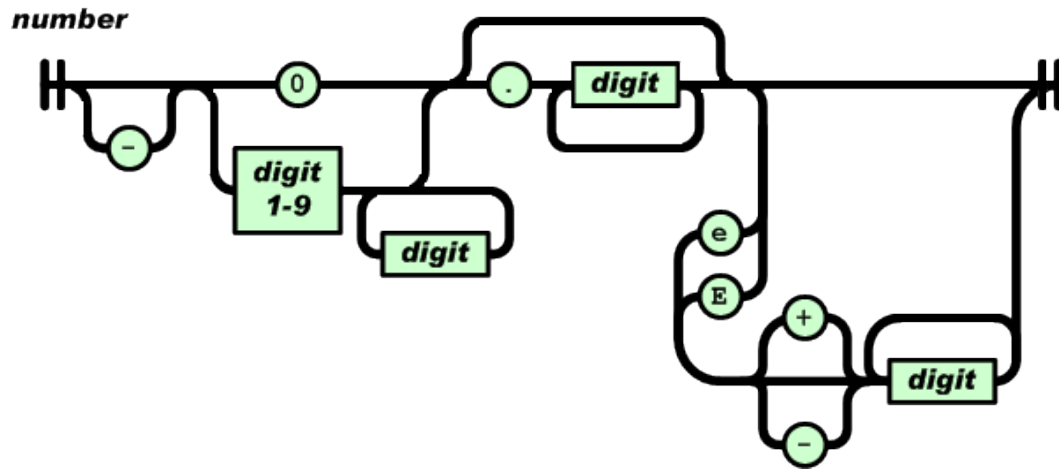


Figura 7. Sintaxis de un valor numérico JSON

Se puede insertar espacio blanco entre cualquier secuencia de estos elementos. (ECMA, 2002)

c. JSON2HTML

json2html es una librería de código abierto que transforma objetos JSON en HTML utilizando una plantilla JSON especificada por el usuario. Hay dos implementaciones de la librería:

- Como un plugin de jQuery
- Como un paquete para uso en un servidor node.js

Ambas implementaciones utilizan la misma gramática de definición de plantillas, por lo que una misma plantilla podría funcionar de igual manera tanto del lado del cliente (jQuery) como del servidor (node.js).

1) Transformaciones

Una transformación especifica una jerarquía de elementos. Cada par nombre-valor del objeto transformador representa los atributos encontrados en el elemento HTML, con la excepción de los siguientes nombres reservados de atributos:

- **tag**, que especifica el tipo de elemento (div, span, etc.)
- **html**, describe el HTML interno contenido en el elemento.
- **children**, especifica los elementos hijos del elemento actual.

Para representar el valor de un elemento del contenedor JSON que se está transformando, se usa la nomenclatura “\${*direccion.del.elemento.deseado*}”.

2) Funcionamiento

JSON2HTML provee una la función \$(selector).json2html(source, transform) cuyo primer parámetro es el JSON que se desea transformar. El segundo parámetro es la plantilla de transformación deseada. El resultado de la transformación se agrega al contenido de cualquier elemento que coincida con el selector indicado (desde donde se hace la invocación de la función). (Moappi Enterprise, 2013)

4. CSS

CSS (Cascading Style Sheets) es el lenguaje para describir la presentación estética de páginas web, incluyendo colores, disposición de elementos y fuentes. Permite adaptar la presentación a diferentes tipos de dispositivos, tales como pantallas, grandes o pequeñas, e impresoras. CSS es independiente de HTML y puede ser utilizado con cualquier lenguaje de marcado basado en XML. La separación de HTML y CSS facilita mantener sitios, compartir hojas de estilos entre varias páginas, y ajustar páginas a diferentes ambientes. Esto es conocido como la separación de estructura y presentación. (World Wide Web Consortium (W3C), 2012)

CSS funciona a base a **reglas**. Una regla de CSS consiste de dos partes principales: Un **selector** y un **bloque**.

En CSS, el estilo aplicado a los elementos del árbol de documento es determinado por búsqueda de patrones. Estos patrones, llamados **selectores**, pueden variar de simples nombres de elementos a patrones contextuales. Si todas las condiciones en el patrón son verdaderas para un elemento, el selector señala ese elemento.

Un **bloque** encierra su contenido entre llaves (“{}”). En el medio puede haber cualquier token, aunque paréntesis, llaves y corchetes deben ocurrir en pares coincidentes y pueden estar anidados. Las comillas simples y dobles también deben ocurrir en pares, y los caracteres en medio son interpretados como una cadena.

Un **bloque declarativo** tiene el mismo inicio y fin que un bloque (llaves “{}”), y en el medio contiene cero o más *declaraciones* separadas por punto y coma “;”.

Una **declaración** consiste de un nombre de propiedad, seguido por dos puntos “:” y un valor para la propiedad.

Una **regla** consiste de un *selector* seguido de un *bloque declarativo*.

Para encontrar el valor de una combinación de elemento/propiedad, los agentes de usuario deben aplicar el siguiente orden:

1. Encontrar todas las declaraciones que aplican al elemento y propiedad en cuestión para el tipo objetivo de despliegue. Se aplican las declaraciones si el selector asociado coincide con el elemento y el medio objetivo coincide con las reglas de media.
2. Ordenar de acuerdo a importancia (normal o !important) y origen (autor, usuario o agente de usuario) en orden ascendente de precedencia:
 - Declaraciones de agente de usuario.
 - Declaraciones normal de usuarios.
 - Declaraciones normal de autor.
 - Declaraciones importantes de autor.
 - Declaraciones importantes de usuario.
3. Ordenar las reglas con la misma importancia y origen por especificidad del selector. Los selectores más específicos sobrescribirán los más generales.
4. Finalmente, ordenar por orden de escritura: si dos declaraciones tienen el mismo peso, origen y especificidad, el último especificado gana. (World Wide Web Consortium (W3C), 2011)

III. METODOLOGÍA

A. Obtención de requerimientos

En esta fase se tuvo varias reuniones con los representantes de CONRED interesados en el proyecto. Se habló principalmente de la situación actual de la Institución, para tener un mejor conocimiento del objetivo general de Megaproyecto y de otras consideraciones a tomar durante su diseño e implementación.

B. Elaboración de Casos de Uso

Una vez determinados los requerimientos se procedió a elaborar los Casos de Uso que daría CONRED a los productos del Megaproyecto. Éstos se establecieron para ayudar a determinar si dichos productos cumplían con los requerimientos del Megaproyecto.

C. Creación de prototipo

Se realizó una prueba de concepto sobre las plataformas propuestas para la aplicación de celulares. Dicha prueba de concepto se mostró a los representantes de CONRED para presentarles un ejemplo de lo que se propuso como respuesta lo obtenido de las fases de obtención de requerimientos y de elaboración de casos de uso.

D. Investigación y Diseño

Se procedió a investigar sobre diferentes plataformas y herramientas para desarrollo y a diseñar el motor de la aplicación con el propósito de simplificar tanto la implementación de dicho motor como la de los formularios individuales requeridos por CONRED, tomando en cuenta factores tales como la extensibilidad y el mantenimiento de la aplicación misma.

E. Implementación

La implementación constó de dos fases principales: la implementación del motor de la aplicación y la implementación de los diferentes formularios de CONRED. En esta etapa se percibió la ayuda que proporcionó el motor a la hora de implementar los formularios, haciendo esta segunda fase más sencilla de implementar y evitando duplicación de código durante la implementación de funcionalidades compartidas entre los formularios.

IV. DISEÑO

A. Requerimientos y Casos de Uso

1. Objetivos

1. Facilitar el procesamiento de la información para que CONRED pueda tomar decisiones en menos tiempo.
2. Disminuir el tiempo de respuesta en caso de una emergencia en alguna comunidad.
3. Utilizar la tecnología para obtener información confiable, que agilice el proceso de evaluación de comunidades que han sido víctimas de desastres naturales.
4. Implementar un sistema usable que facilite el llenado y envío de reportes y formularios a usuarios poco experimentados con teléfonos inteligentes.

2. Criterios de Éxito

Se estableció que el producto se considerará exitoso al cumplir con los siguientes criterios:

1. La utilización de la aplicación disminuyó el tiempo de envío y procesamiento de los datos de las evaluaciones.
2. La aplicación está siendo utilizada por todas las personas encargadas de evaluar las situaciones que se presentan en las comunidades que han sido afectadas por algún desastre natural.
3. El tiempo de respuesta de CONRED hacia las comunidades que necesitan ayuda disminuye.

3. Características Principales

Se estableció que el producto final deberá implementar la siguiente funcionalidad:

1. **Autenticación del usuario.** El sistema reconoce el nombre y región a la que pertenece el usuario.
2. **Formularios:** Los datos relacionados a las evaluaciones solicitadas se ingresan en una plantilla predefinida, en la cual se proporcionan opciones múltiples para hacer su uso más rápido y más fácil. Los formularios solicitados por CONRED son:
 - Evaluación preliminar de situación.
 - Evaluación preliminar de habitabilidad.

- Evaluación preliminar de situación.
- Evaluación rápida de daño en edificaciones.
- Evaluación rápida de centros educativos.

3. Geoposicionamiento. La aplicación es capaz de enviar coordenadas de ubicación obtenidas del sistema de Geoposicionamiento del celular.

4. Imágenes. La aplicación permite el envío de fotografías tomadas por el celular.

5. Almacenamiento temporal de la información. La información ingresada en las plantillas de evaluación será almacenada en la memoria interna de los dispositivos, en caso de que no se cuente con una conexión a internet, para ser enviada cuando se establezca la conexión.

6. Servidor de base de datos. Los datos recopilados en las evaluaciones se almacenan en una base de datos central alojada por la Sede Central de CONRED inmediatamente después de haberse enviado vía internet. CONRED maneja en su sistema central una base de datos de IBM Lotus Notes, la cual funciona bajo el paradigma de una base de datos documental. No obstante, fue indicado durante las reuniones por parte de CONRED que se proveerá para el Megaproyecto un servidor de bases de datos MySQL y que ellos internamente harán la obtención de datos de dicho servidor y la alimentación de su sistema con los datos que ahí se encuentren.

B. Arquitectura de la aplicación

1. Plataforma de desarrollo móvil

Para la distribución del producto, CONRED adquirió para los delegados 30 teléfonos con Sistema Operativo Android 2.3. No obstante, se estableció el deseo de no desarrollar la aplicación sólo para un sistema operativo, ya que es probable que en el futuro se opte por adquirir teléfonos con otra plataforma.

Existen varios marco de trabajos para desarrollo móvil multiplataforma, tales como LibGDX, Corona, Titanium, PlayN y Phonegap. De éstos se eligió Phonegap, en el cual se desarrolla usando tecnologías web como HTML, JavaScript y CSS, ya que es de esperar que con el tiempo la aplicación necesite ser modificada (dado que en CONRED es común que se modifiquen los campos de los formularios), extendida o que se le dé mantenimiento (como en el posible caso de un cambio de servidor de Base de Datos). El personal de CONRED no cuenta con mucha

disponibilidad de tiempo para aprender nuevos lenguajes de programación (como es el caso con Corona, el cual es un marco de trabajo para Lua) ni aprender muchos conceptos y procedimientos específicos de desarrollo móvil. En muchos aspectos, la programación en Phonegap es como en una aplicación web tradicional, por lo que no se requerirá mucho tiempo para entender la lógica de la aplicación y realizar cambios al código.

2. Diseño del motor de la aplicación

Para la implementación de un formulario en el proyecto se pueden identificar 3 procesos principales:

- El despliegue del formulario donde se ingresará la información.
- La obtención de los datos ingresados al formulario.
- El llenado de un formulario con datos previamente guardados.

Esto significa que una implementar aisladamente cada formulario requerido implicaría repetir los tres procesos cinco veces, lo cual tiene varias desventajas, tales como:

- Código repetido con escasas modificaciones entre formularios.
- El código de los formularios ya existentes no estaría lo suficientemente generalizado, lo que implica un proceso más largo de creación para cada formulario, incluso después de haber implementado los formularios de los requerimientos iniciales.
- Al cambiar un campo de un formulario, se tendría que modificar sus respectivos procesos de carga y guardado de datos.
- Cada cambio sobre un campo de un formulario puede implicar hacer un cambio en las bases de datos, tanto la local del dispositivo como la que se encuentra en el servidor central.

Se tomó en cuenta que CONRED tendrá que hacer un proceso de transferencia de datos a partir de una base de datos relacional hacia una documental, por lo que realizar un diseño relacional para modelar los datos del formulario puede significar más tiempo requerido para la obtención y alimentación de datos.

Por estas consideraciones se decidió utilizar ambas bases de datos (del celular y del servidor central) para almacenar la información de los formularios en un formato parseable, en vez de procesar los datos y almacenarlos en un diseño relacional muy detallado. Para generalizar el proceso de cargar, desplegar, guardar y enviar información se planteó un motor de transformaciones biyectivas entre HTML y dicho formato de almacenamiento (cuya elección será discutida en breve).

Este motor tendrá el propósito de obtener la información ingresada en un formulario desplegado por medio de funciones genéricas, y de transformar los datos hacia el formato de almacenamiento. De similar manera, dado un conjunto de datos previamente obtenidos por medio de un formulario, hacer la transformación desde el formato de almacenamiento en que se encuentra hacia su respectivo despliegue de formulario, con los datos ya almacenados. Para crear formulario en blanco, se hará el mismo proceso, con un conjunto de información vacío. Esto también da la posibilidad de establecer valores predeterminados en caso esto sea necesario.

También se consideró que, en la mayoría de los casos, CONRED trabaja bajo condiciones de telecomunicaciones mucho menos que ideales. Esto significa que para tener una mejor posibilidad de que la información se pueda transmitir por medio de internet, el tamaño de los datos deberá ser optimizado. No se puede utilizar un formato que aumente innecesariamente el volumen de los datos.

Para almacenamiento de datos se tomaron en cuenta los formatos XML y JSON. XML cuenta con la ventaja de ser compatible con el lenguaje XSLT (Extensible Stylesheet Language Transformations), que realiza transformaciones entre documentos XML. Ésta es una ventaja muy fuerte, ya que HTML es un lenguaje subconjunto de XML, por lo que HTML es generado directamente por medio de XSLT. De similar manera, se puede parsear HTML (interpretado como XML) y obtener un archivo XML en base a una transformación.

JSON por su parte es un formato más compacto que XML debido a la definición de su sintaxis, que es más abreviada. También provee soporte para tipos de datos nativos, tales como valores numéricos, cadenas, valores booleanos y valores nulos.

No hay un estándar definido para JSON que haga lo que hace XSLT, pero durante la investigación se encontró la librería JSON2HTML, descrita en el marco teórico. Esta librería no cuenta con la flexibilidad de XSLT. Específicamente, el marco de trabajo no obtiene JSON en base a HTML. Con respecto a reservas sobre incluir una librería externa para implementar transformaciones, ningún lenguaje tiene una ventaja sustancial sobre el otro, ya que ninguno está implementado nativamente en JavaScript.

Para tomar la decisión sobre el formato, el factor espacio fue muy influyente, tanto por la baja velocidad de la conexión de datos que se espera manejar como por el hecho de que la aplicación es para dispositivos móviles, por lo que el espacio también es limitado. Finalmente se eligió JSON como el formato de almacenamiento, contando con JSON2HTML y decidiendo implementar un algoritmo para obtener JSON en base a un formulario HTML lleno.

3. Sistema en la base de datos central

No es posible guardar información en una base de datos MySQL utilizando únicamente JavaScript. La forma correcta es por medio de un servicio que recibe una petición HTML y realiza una acción sobre sus parámetros. En este caso, el servicio realizaría una validación de los datos, los almacenaría en la base de datos y retornaría una respuesta a la aplicación. Esta respuesta es importante, ya que por la importancia de los datos que se envían no se puede asumir que han sido recibidos correctamente sin una confirmación.

Como apoyo al otro módulo de aplicación del Megaproyecto, CONRED proveyó también un servidor de PHP, por lo que se decidió aprovechar este recurso y realizar los servicios en dicho lenguaje.

4. Diagrama general del sistema y sus componentes

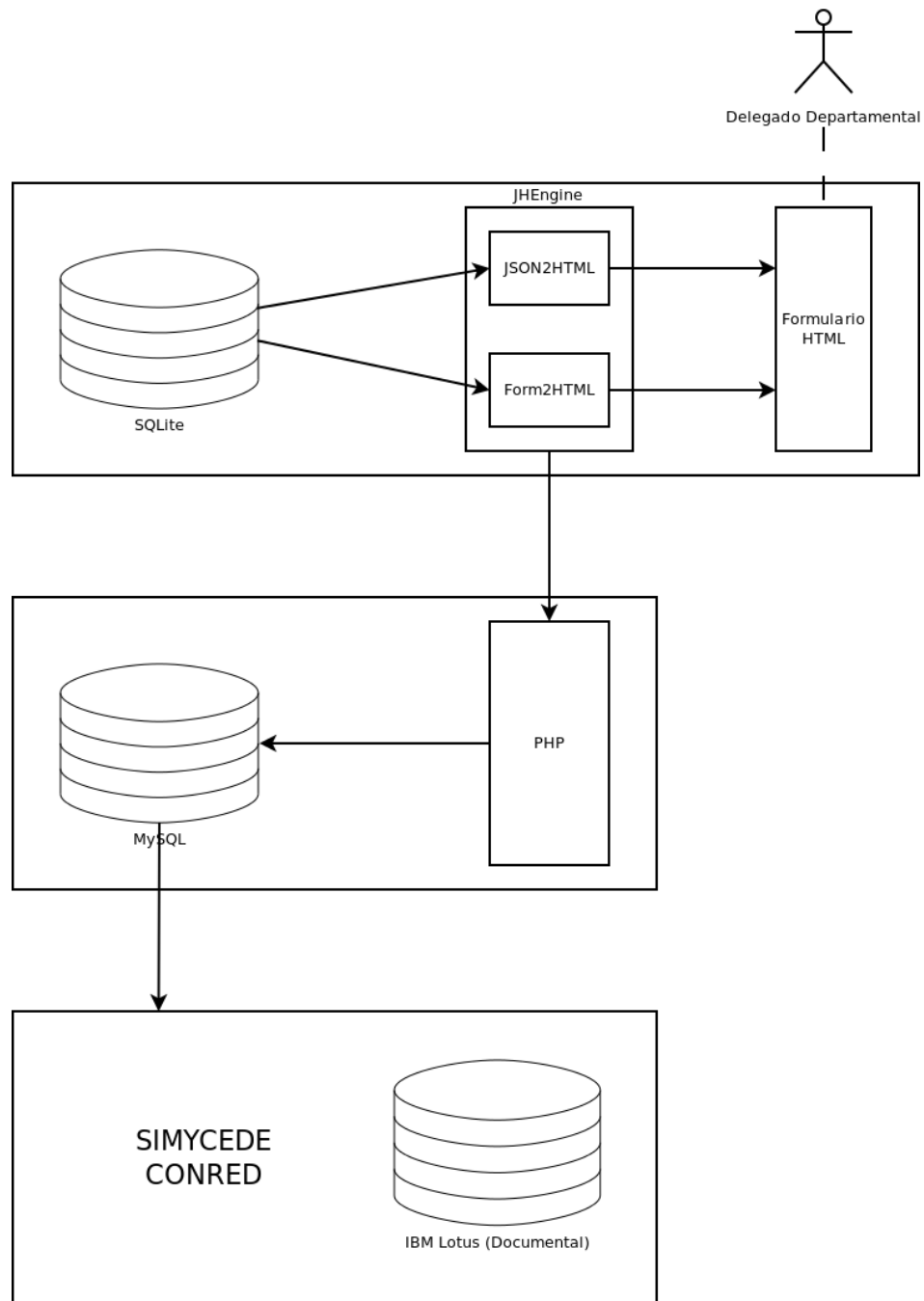


Figura 8. Diagrama general de la arquitectura de la aplicación.

5. Diseños de las bases de datos

a. SQLite

La base de datos de la aplicación servirá para almacenar los borradores de los formularios que los delegados llenan, indicando si hay una versión guardada pendiente de envío. Los usuarios podrán enviar versiones parcialmente llenas de los formularios al servidor central, el cual tendrá la función de almacenar todas las versiones de un mismo formulario, mientras en la aplicación sólo se almacenará la más reciente, haya sido ya enviada (ya que en el futuro se puede corregir o agregar información) o no.

En caso de haber un formulario que se deseó enviar, pero no se pudo, la aplicación correrá un servicio de manera periódica, el cual intentará realizar de nuevo el envío. Los identificadores para dichos formularios pendientes se tendrán almacenados en una tabla que funcionará como la cola que el servicio deberá procesar (pending_forms).

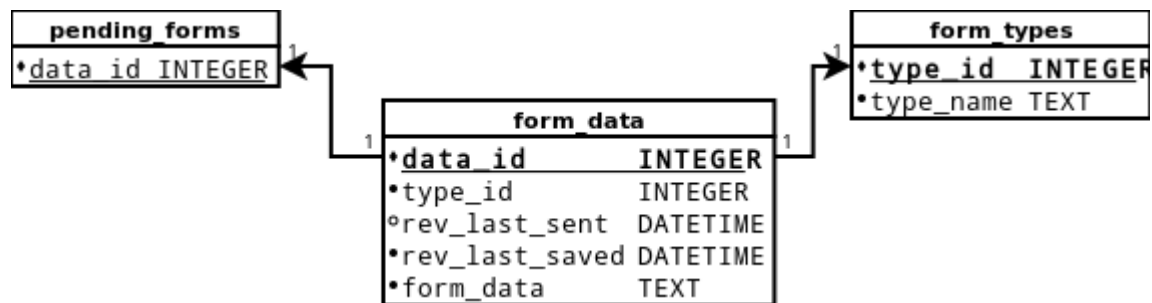


Figura 9. Diseño de la base de datos SQLite local para uso de la aplicación.

b. MySQL

La base de datos central es la encargada de archivar todos los envíos, es decir, de almacenar todas las versiones que los delegados generan de sus diferentes formularios. También es encargada de realizar una autenticación básica en el sistema, para que CONRED pueda determinar quién hizo el envío. Los usuarios se asocian con su número de teléfono, que actúa como su identificador único. Dicho número se obtiene desde la aplicación móvil.

Para poder determinar la procedencia de un envío, aparte del identificador único autoincrementable que la aplicación móvil asigna a cada formulario, se cuenta con campos que contienen el usuario que realiza el envío y el identificador único autoincrementable del envío en el contexto local del teléfono.

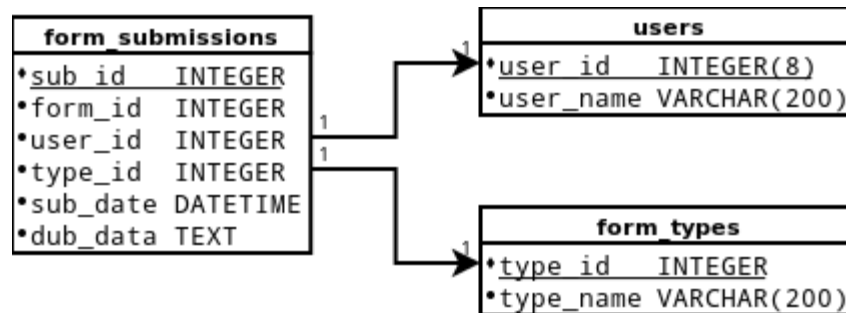


Figura 10. Diseño de la bse de datos MySQL para almacenaje de la información en la sede central.

B. Características de la aplicación

1. Ingreso de datos nuevos

El usuario selecciona el tipo de formulario que desea llenar, y el motor utiliza un conjunto de información predeterminado para generar el despliegue con los datos iniciales ya ingresados. Cada diferente tipo de formulario tiene su propio despliegue y puede tener sus propios valores predeterminados.

2. Guardado de información recolectada

Luego de ingresar información recabada en un formulario, el usuario podrá guardar en la base de datos de la aplicación dicha información. A nivel de aplicación, cuando se desea enviar un formulario al servidor central, la información se guarda en la base de datos primero, para obtener el identificador local del formulario, el cual sirve para identificar los envíos en la base de datos central.

3. Envío de información recolectada

El propósito principal de la aplicación es enviar información a la base de datos en el servidor central. El servidor central es el que archiva las diferentes versiones enviadas de todos los formularios. La última versión de cada formulario permanece guardada en la base de datos de la aplicación, pero los formularios que ya fueron enviados son marcados para que en la pantalla de cargar un formulario previamente trabajado, los formularios que no han sido enviados tengan una prioridad mayor.

Si un usuario presiona el botón de envío, pero por causas externas éste no puede ser completado o confirmado, el formulario será agregado a la cola de envíos pendientes, y se calendarizará un servicio de la aplicación que regularmente volverá a intentar realizar el envío y notificará al usuario si se pudo realizar el envío o si se contactó al servidor central pero el servicio rechazó el envío debido a una condición no cumplida (un usuario no existente en la base de datos o un tipo de formulario que no se encuentra ingresado).

4. Modificación de datos previamente guardados

Un formulario previamente guardado (haya sido enviado o no) puede volver a cargarse para modificar la información ingresada. Esta característica permite el envío de datos que son más determinantes para la toma de decisiones en la Sede Central de CONRED. El usuario puede luego seguir trabajando en campos menos críticos, y enviar la información completa posteriormente.

También puede aprovecharse esta característica para corregir datos erróneos previamente enviados, aunque debido a que CONRED correrá un proceso automático de cosecha de datos, no se recomienda enviar datos erróneos a sabiendas, ya que pueden ser alimentados a la base de datos de SIMYCEDE y provocar decisiones incorrectas.

5. Mantenimiento

El código de la aplicación se entregará completo a CONRED, para que en caso haya un cambio de plataforma o partes de la aplicación se vuelvan obsoletas con tecnologías futuras, el personal informático pueda modificarlo y que la aplicación siga siendo utilizable.

Otra acción que se tomó para que el personal de CONRED pueda darle mantenimiento a la aplicación fácilmente fue el desarrollar la aplicación con el marco de trabajo Phonegap, que ya se mencionó que utiliza tecnologías web para compilar aplicaciones nativas para el sistema operativo del celular. Es muy común que personal de informática tenga ya conocimientos sobre cómo utilizar estas tecnologías, y en caso de no tenerlo y tener que aprender, el estándar es abierto a todo el público y existe documentación muy extensiva, lo cual significa una curva de aprendizaje menos pronunciada.

6. Extensibilidad

CONRED solicitó que la aplicación incluyera 5 formularios determinados por ellos. No obstante, los campos de dichos formularios son cambiados constantemente, y hay muchos más tipos de formularios que pueden ser útiles en la aplicación. Uno de los propósitos principales del diseño del motor es el disminuir la cantidad de trabajo que toma la implementación normal de un nuevo tipo de formulario, y que en caso sea necesario modificar campos de un formulario, esto se pueda realizar fácil y rápidamente.

El motor permite que por medio de JSON se especifique la transformación que permitirá al motor transformar información para ser desplegada en un formulario. El motor también obtiene la información de un formulario que utiliza la nomenclatura correcta en formato JSON. Para un nuevo tipo de formulario, se deberá crear la correspondiente plantilla y agregar una entrada en la base de datos local de la aplicación y en la base de datos central.

V. RESULTADOS Y DISCUSIÓN

A continuación se presenta una muestra tanto de lo que ven los usuarios al utilizar la aplicación como algunos aspectos internos de la misma, para ilustrar a grandes rasgos el producto final.

A. Pantalla Inicial

La pantalla inicial únicamente contiene los botones que permiten seleccionar un tipo de formulario para empezar a ingresar datos o seleccionar un formulario previamente guardado para seguir ingresando datos.

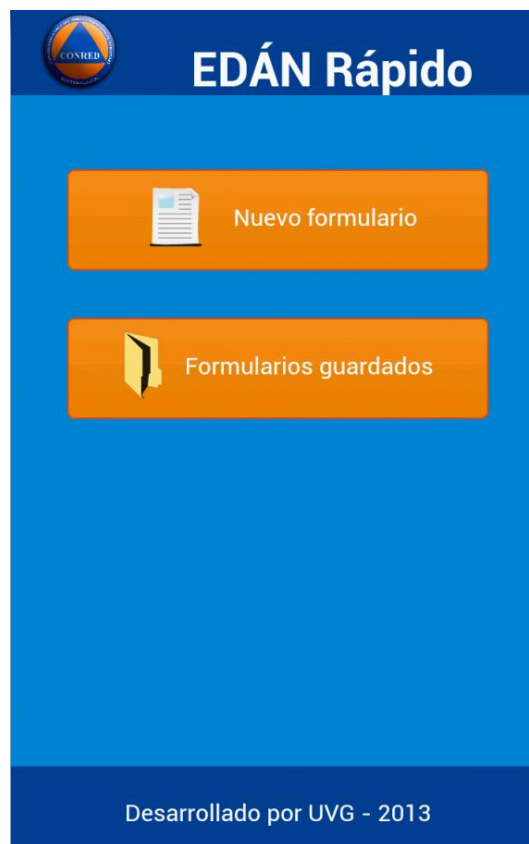


Figura 11. Pantalla inicial de la aplicación.

Durante la primera ejecución se inicializa la base de datos y se obtiene el número de teléfono del dispositivo (obtenido de su tarjeta SIM). En caso el operador telefónico no almacenara el número de teléfono en la tarjeta, se pide el número telefónico en una ventana de diálogo.

B. Nuevo Formulario

Esta pantalla despliega la lista de los tipos de formulario disponibles para llenar con la aplicación.

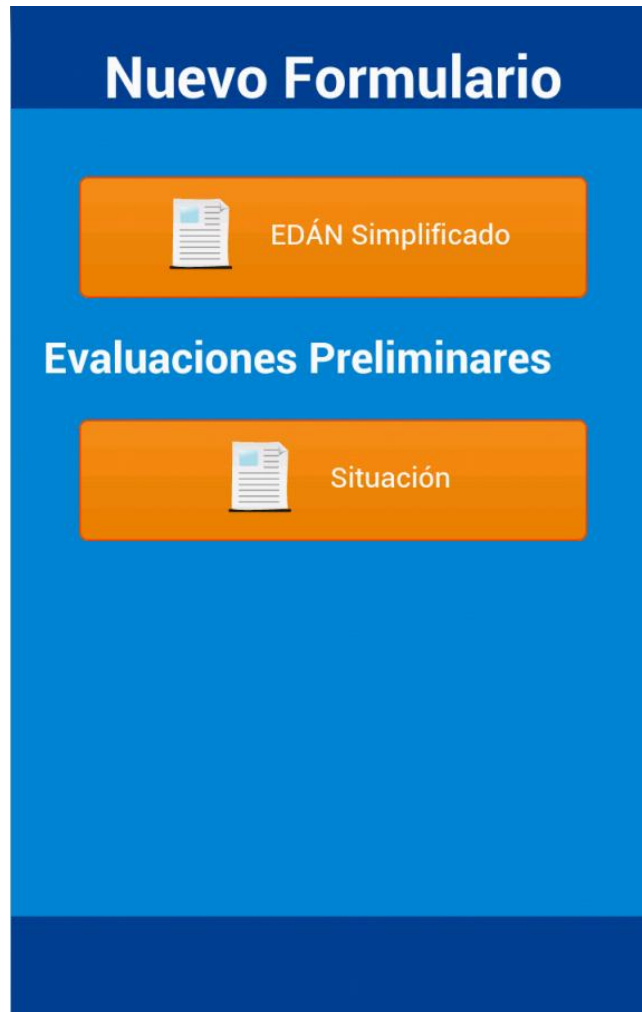


Figura 12. Pantalla para iniciar a llenar un formulario en blanco.

C. Formularios Guardados

Esta pantalla despliega los formularios que ha llenado el usuario anteriormente. Se muestran en dos divisiones: Formularios pendientes de envío y Formularios enviados. Los formularios se encuentran ordenados (dentro de su respectiva “categoría”) por la fecha en que fueron guardados por última vez, ya que conforme se hayan enviado versiones finales, éstas descienden a una prioridad menor que formularios nuevos que se hayan enviado recientemente o que estén guardados en la aplicación, hayan sido enviados o no, para que los delegados trabajen en ellos luego.



Figura 13. Pantalla de carga de un formulario previamente llenado y guardado en la base de datos del teléfono.

D. Ingreso de datos

Esta página contiene la funcionalidad principal de la aplicación ya que es donde los delegados ingresan los datos recolectados. Después de llenar el formulario, el usuario decide guardar o enviar la información que llenó. El programa sigue la siguiente lógica para el almacenaje y/o envío.

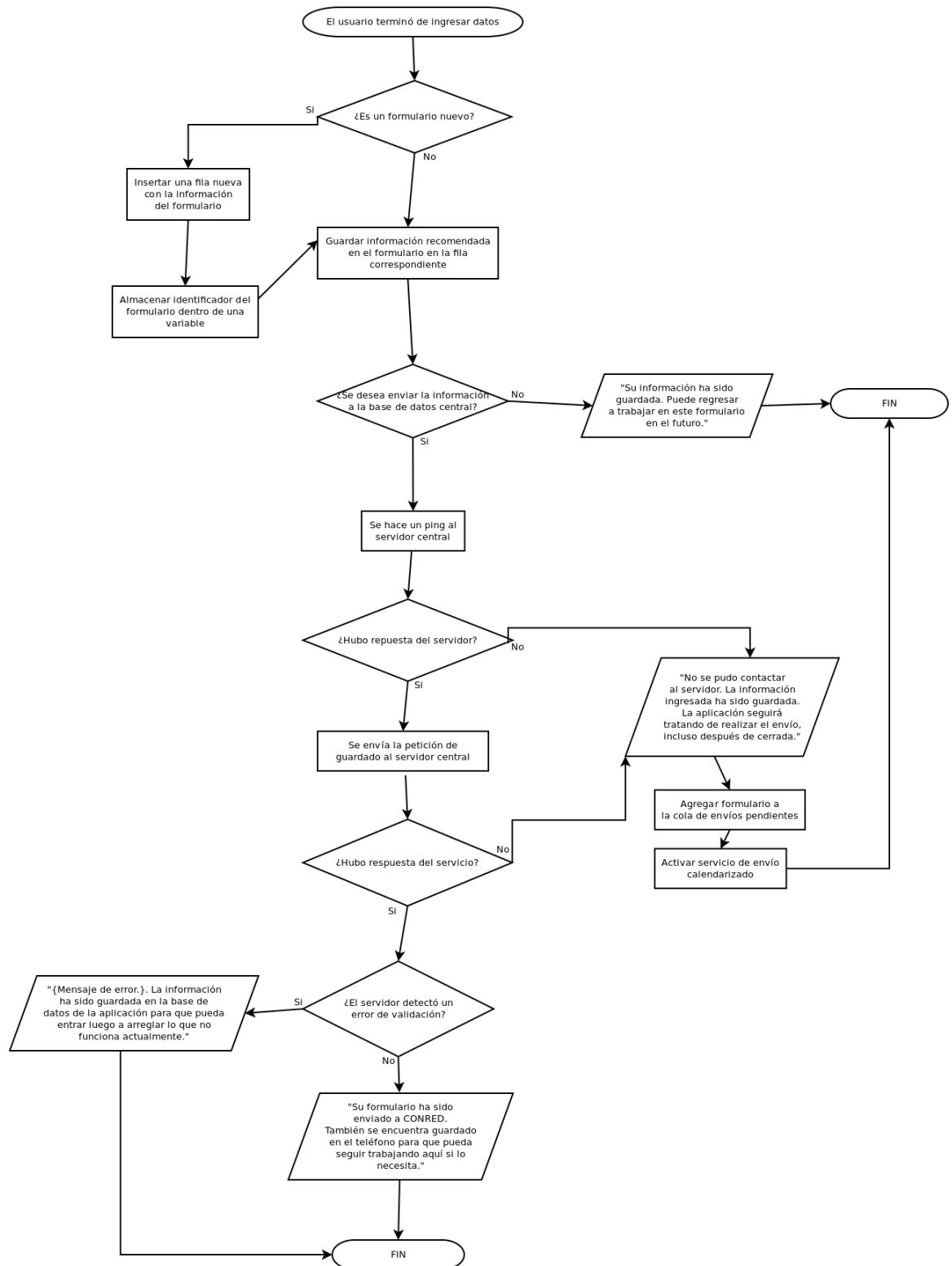


Figura 14. Diagrama de flujo que describe el proceso de guardado/envío de un formulario lleno o parcialmente lleno.

EDÁN Rápido

Guardar Borrador Guardar y Enviar

Información del incidente

Evento generador y sus efectos

Ubicación

Categoría

Nombre

Municipio

Municipio

Departamento

Coordenadas

Latitud

Copyright © 2013 - A.C., D.O. y A.S.

Figura 15. Pantalla de ingreso de datos de la aplicación.

E. jhEngine

j(ason)h(tml)Engine es el motor de transformaciones biyectivas en JSON que se implementó en el proyecto. Funciona principalmente como un wrapper para la librería JSON2HTML y la implementación de la función que obtiene en formato JSON los datos ingresados en un formulario HTML, llamada formToJSON.

La función formToJSON es asociada a los elementos de formulario (<form/>) de un documento HTML y lo que hace es iterar sobre todos los elementos de ingreso de datos y en base al atributo name, que se encuentra en la forma de una llamada al contenido de un contenedor JSON (“json.object.value”), almacena la información en dicho formato.

También incluye un conjunto de funciones para agregar funcionalidad de geoposicionamiento y cámara, entre otros, a los formularios por medio de contenedores con clases CSS con los que se selecciona el elemento con jQuery y se les asocia la función manejadora.

jhEngine funciona como un objeto que almacena las plantillas para los diferentes tipos de formularios y automatiza y generaliza el proceso de despliegue de datos a partir de información ya almacenada o a partir de valores predeterminados.

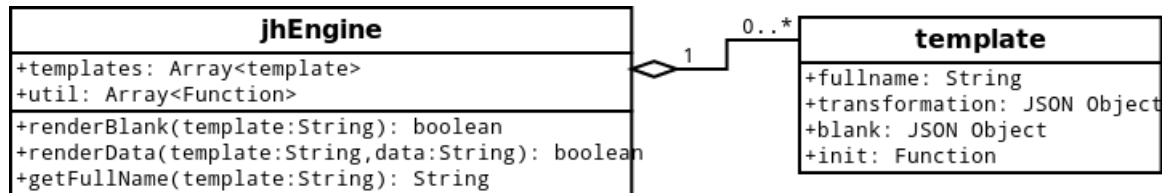


Figura 16. Diagrama de clases para el motor jhEngine.

Para agregar un formulario nuevo en jhEngine, se debe agregar un objeto “template” al arreglo que se encuentra en el objeto principal de jhEngine con los siguientes atributos:

- El nombre que se le despliega al usuario cuando está usando el formulario.
- La plantilla de transformaciones que se usará para desplegar el formulario en HTML a partir de datos en formato JSON.
- De ser necesario, un conjunto de datos que se desea cargar inicialmente al formulario. Para que funcione basta con ingresar únicamente el elemento raíz de la estructura.
- Una función de inicialización que permite ligar las acciones que corresponden a diferentes botones u otras acciones.

Se puede ver que el formulario se agregó de manera sencilla, y que ya dada la transformación, el sistema se encarga de generar los formularios y de llenarlos con los datos que se le indiquen, ya sea datos en blanco o cargados de una versión anterior. También hace posible representar los datos ingresados en el formulario por medio de JSON de manera genérica para todos los formularios que se implementen.

```

jhEngine.templates.ejemplo = {
  fullname: "Formulario de ejemplo",
  transformation: [
    {"tag":"div","id":"sectionChooser","children":[
      {"tag":"select","children":[
        {"tag":"option","value":"","html":"Seleccione una
p\xElgina para ingresar datos"}
      ]}
    ]},
    {"tag":"form","children":[
      {"tag":"div","class":"section","id":"form","children":[
        {"tag":"div","class":"row","children":[
          {"tag":"label","html":"Nombre"},
          {"tag":"input","type":"text","name":"ejemplo.nombre","value":"${e
jemplo.nombre}","placeholder":"Nombre","html":""}
        ]},
        {"tag":"div","class":"row","children":[
          {"tag":"div","class":"coordenadas","children":[
            {"tag":"label","html":"Coordenadas"},
            {"tag":"label","class":"level2","html":"Latitud"},
            {"tag":"input","type":"number","name":"ejemplo.coordenadas.latitu
d","value":"${ejemplo.coordenadas.latitud}","placeholder":"Latitu
d","html":""},
            {"tag":"label","class":"level2","html":"Longitud"},
            {"tag":"input","type":"number","name":"ejemplo.coordenadas.longit
ud","value":"${ejemplo.coordenadas.longitud}","placeholder":"Long
itud","html":""},
            {"tag":"div","class":"btn","html":"Obtener
coordenadas"}
          ]}
        ]}
      ]}
    ],
    blank: {
      "ejemplo": {}
    },
    init: function() {
      util.baseInit();
    }
  }
}

```

Figura 17. Objeto <template> correspondiente a un formulario sencillo en jhEngine.

Formulario de ejemplo

Guardar Borrador Guardar y Enviar

Seleccione una página para ingresar datos ▼

Nombre

Nombre

Coordenadas

Latitud

Latitud ▲ ▼

Longitud

Longitud ▲ ▼

Obtener coordenadas

Desarrollado por UVG - 2013

Figura 18. Despliegue obtenido a partir de la plantilla de transformación del ejemplo básico de la Figura 17.

```
{ "ejemplo":
  {
    "nombre": "Alejandra Canahui",
    "coordenadas":
      {
        "latitud": 14.6055777,
        "longitud": -90.4887356
      }
  }
}
```

19. Datos en formato JSON obtenidos a partir del formulario básico de ejemplo por medio de jhEngine.

F. Servicio para envíos pendientes

En el caso de que un envío al servidor central no se pueda realizar, se agrega su identificador a la cola de envíos pendientes y si dicha cola se encontraba vacía anteriormente, se activa el servicio que se encarga de intentar enviarlos periódicamente hasta que el servidor central responda y confirme el envío.

Este servicio presenta la ventaja de permitir a los usuarios cerrar la aplicación y realizar otras tareas sin tener que revisar periódicamente si ya cuentan con una conexión a internet adecuada y luego ingresar de nuevo a la aplicación para realizar manualmente el reenvío. Esto resulta aún más útil cuando el intento de reenvío no es exitoso porque se debería repetir el proceso previamente descrito hasta el momento en que se logre enviar los datos.

El marco de trabajo Phonegap no soporta la calendarización de tareas, por lo que se programó una adición para el marco de trabajo que hace la llamada al sistema operativo para que calendarice lo que en Android se conoce como un Intent. Un Intent es una tarea del sistema operativo.

Para poder ejecutar el servicio aún con la aplicación cerrada también se tuvo que implementar un código para envío de formularios con una lógica similar a la implementada en Phonegap, utilizando programación nativa de Android.

La secuencia de las llamadas entre clases que hacen posible el servicio se describe en la próxima figura.

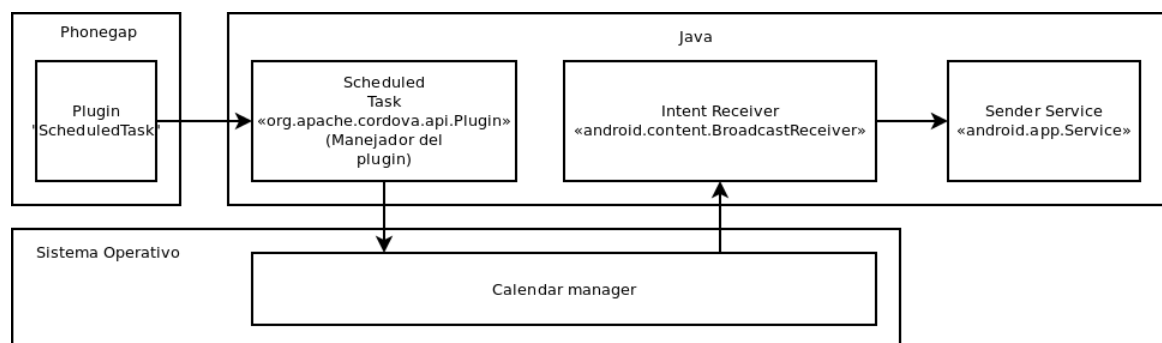


Figura 20. Diagrama de llamadas entre los componentes involucrados en el inicio del servicio de envío.

G. Procesos del servidor de la base de datos central

La base de datos central realiza dos procesos:

- Confirmar a la aplicación móvil que se tiene una conexión establecida para que proceda con el envío de datos.
- Recibir datos enviados por la aplicación. Esto implica:
 1. Verificar que el usuario que envía y el tipo de formulario que se está enviando sean válidos.
 2. Almacenar los datos recibidos a la base de datos central.
 3. Enviar a la aplicación que hizo la petición ya sea un mensaje de error o la confirmación de la operación.

El primero de estos procesos es sobremanera simple, ya que únicamente retorna un mensaje de confirmación con valor verdadero (empaquetado en formato JSON) para que la aplicación interprete el valor como un indicador del estado de la conexión con el servidor y proceda a hacer el envío de los datos recolectados.

A continuación se encuentra el código del archivo **ping.php** que lleva a cabo este proceso.

```
echo json_encode(array('success'=>'true'));
```

Figura 21. Código del archivo ping.php en el servidor central

El segundo proceso se ejecuta como una transacción de SQL para que en caso de un error la base de datos mantenga su integridad. Este proceso también retorna una respuesta empaquetada en JSON, que indica si se tuvo éxito o si se detectó algún error.

A continuación se encuentra el código del archivo **submit.php** que lleva a cabo este proceso.

```

$dbh = new
PDO("mysql:host=".$config['dbhost'].";dbname=".$config['dbname'],
$config['dbuser'], $config['dbpassword']);
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
try{
    $dbh->beginTransaction();

    // Verificaciones:
    // Existe el usuario?
    $user = $dbh->prepare("SELECT * FROM users WHERE user_id =
:user_id");
    $user->bindParam(":user_id", $_REQUEST['user_id']);
    $user->execute();
    if(!$user->fetch()){
        $response = array(
            "error" => "Usuario inválido."
        );
        sendResponse($response);
    }
    // Existe el tipo?
    $type = $dbh->prepare("SELECT * FROM form_types WHERE
type_id = :type_id");
    $type->bindParam(":type_id", $_REQUEST['type_id']);
    $type->execute();
    if(!$type->fetch()){
        $response = array(
            "error" => "Tipo de formulario inválido."
        );
        sendResponse($response);
    }
    $form_id = $_REQUEST['form_id'];
    if($form_id < 0){
        $form_id = $dbh->prepare("SELECT MAX(form_id) AS
form_id FROM form_submissions");
        $form_id->execute();
        $form_id = $form_id->fetch();
        $form_id = intval($form_id['form_id'])+1;
    }
    $insert = $dbh->prepare("INSERT INTO
form_submissions(form_id, user_id, type_id, sub_date, sub_data)
VALUES(:form_id, :user_id, :type_id, NOW(), :sub_data)");
    $insert->bindParam(":form_id", $form_id);
    $insert->bindParam(":user_id", $_REQUEST['user_id']);
    $insert->bindParam(":type_id", $_REQUEST['type_id']);
    $insert->bindParam(":sub_data", $_REQUEST['sub_data']);
    $insert->execute();

    $dbh->commit();
    $response['success'] = true;
    $response['form_id'] = $form_id;
}

```

```
    } catch(Exception $e) {  
        $dbh->rollback();  
        $response = array(  
            "error" => $e->getMessage(),  
        );  
    }  
  
    sendResponse($response);
```

Figura 22. Código del archivo submit.php en el servidor central

Por último, cabe mencionar que la configuración para acceso a la base de datos se ingresa en el archivo **config.php**, que contiene un arreglo con las siguientes variables:

- **dbhost:** La dirección IP (o localhost si ésta aplica) en donde se encuentra corriendo el servidor de bases de datos en la que se almacenarán los envíos.
- **dbname:** El nombre de la base de datos en la que se almacenarán los envíos.
- **dbuser:** El nombre del usuario con permisos para ingresar los datos a almacenar en la base de datos indicada.
- **dbpassword:** La contraseña que utiliza el usuario para autenticarse al servidor de bases de datos.

VI. CONCLUSIONES

- La implementación de un motor de aplicación que generalice el proceso de desplegar información previamente guardada en un formulario y de obtener los datos ingresados en un formulario disminuye la cantidad de pasos a realizar y de código a escribir para poder agregar uno o varios tipos de formularios a la aplicación.
- Desarrollar una aplicación utilizando técnicas y lenguajes ampliamente conocidos y herramientas de código abierto disminuye la cantidad de tiempo que se debe dedicar a aprender lo necesario para realizar mantenimiento sobre dicha aplicación y para extender su funcionalidad.
- En una aplicación para teléfonos celulares inteligentes que realiza operaciones críticas utilizando una conexión a internet en entornos donde las telecomunicaciones no están en buen estado, es recomendado utilizar el estándar JSON, ya que éste describe estructuras de objetos utilizando menos caracteres que XML y por ende el tamaño de la información a enviar es menor.
- Bajo las condiciones de telecomunicaciones previamente descritas y tomando en cuenta la presión bajo la cual trabajan los delegados departamentales cuando van a recolectar datos, el servicio de envío de formularios les ahorra la tarea de periódicamente volver a intentar enviar los datos que no se han transferido al servidor central. Esto les permite concentrarse de lleno en otras asignaciones que puedan tener.

VII. RECOMENDACIONES

- Desarrollar una plataforma que provea una interfaz amigable para la creación de plantillas de transformación para nuevos tipos de formularios.
- Diseñar e implementar un lenguaje formal para transformaciones de datos en formato JSON.
- Portar el servicio de envío de formularios pendientes a otros sistemas operativos como iOS para poder exportar la aplicación a dichas plataformas.
- Implementar la capacidad de reconocimiento de voz para posibilitar un llenado de formularios más ágil y rápido.
- Realizar minería de datos sobre la información obtenida por el sistema para identificar patrones de relación entre parámetros de las evaluaciones y decisiones tomadas para agilizar la respuesta proporcionada por CONRED en desastres naturales.
- Implementar un chequeo de integridad entre la aplicación móvil y el servidor central de CONRED para saber con más certeza si los datos se enviaron y recibieron correctamente.

VIII. Bibliografía

Adobe Systems, Inc. (2013). *Phonegap / About*. Recuperado el 29 de Agosto de 2013, de Phonegap: <http://phonegap.com/about/>

Berners-Lee, T. (Junio de 1993). *Hypertext Markup Language (HTML)*. Recuperado el 29 de Agosto de 2013, de World Wide Web Consortium (W3C): <http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>

ECMA. (2002). *Introducing JSON*. Recuperado el 29 de Agosto de 2013, de JSON: <http://json.org/>

Flanagan, D. (2011). *JavaScript: The Definitive Guide* (6ta. ed.). Sebastopol, California: O'Reilly Media.

Intel Corporation. (2012). *The Development of Mobile Applications using HTML5 and PhoneGap on Intel Architecture-Based Platforms*. Recuperado el 29 de Agosto de 2013, de Intel Developer Zone: <http://software.intel.com/en-us/articles/the-development-of-mobile-applications-using-html5-and-phonegap-on-intel-architecture-based>

Moappi Enterprise. (2013). *Transform JSON to HTML*. Recuperado el 29 de Agosto de 2013, de JSON2HTML: <http://json2html.com/>

Molina Rivera, Y. J., Sandoval Cardona, J., & Toledo Franco, S. A. (2012). *Sistema operativo Android: Características y funcionalidad para dispositivos móviles*. Pereira, Colombia: Universidad Tecnológica de Pereira.

The jQuery Foundation. (2013). *jQuery*. Recuperado el 29 de Agosto de 2013, de jQuery: <http://www.jquery.com>

The PHP Group. (11 de Octubre de 2013). *What is PHP?* Recuperado el 29 de Agosto de 2013, de PHP: Hypertext Preprocessor: <http://www.php.net/manual/en/intro-whatis.php>

World Wide Web Consortium (W3C). (28 de Mayo de 2013). *Differences from HTML4*. Recuperado el 29 de Agosto de 2013, de World Wide Web Consortium (W3C): <http://www.w3.org/TR/html5-diff/>

World Wide Web Consortium (W3C). (2012). *HTML & CSS*. Recuperado el 29 de Agosto de 2013, de World Wide Web Consortium (W3C): <http://www.w3.org/standards/webdesign/htmlcss#whathtml>

World Wide Web Consortium (W3C). (07 de Junio de 2011). *Introduction to CSS 2.1*. Recuperado el 29 de Agosto de 2013, de World Wide Web Consortium (W3C): <http://www.w3.org/TR/CSS21/cover.html#minitoc>

Yoon, H.-J. (Abril de 2012). A Study on the Performance of Android Platform. *International Journal on Computer Science & Engineering* , 532-537.

IX. APÉNDICE

A. Formulario de Evaluación de Daños y Análisis de Necesidades (EDÁN) para Android

DATOS GENERALES		(A) FECHA EVALUACIÓN								(B) FECHA INCIDENTE								EVALUACION DE DAÑOS Y ANALISIS DE NECESIDADES																							
		DIA		MES		AÑO		HORA		DIA		MES		AÑO		HORA																									
1														INFORMACIÓN DEL EVALUADOR																											
		(C) NOMBRE												(D) INSTITUCIÓN																											
														(E) TELÉFONO												(F) CARGO															
INFORMACIÓN DEL INCIDENTE		(A) EVENTO GENERADOR Y SUS EFECTOS																																							
2		(B) LABOR, PUEBLO, PARCELAMIENTO, CANTÓN, CASERÍO Y/O ALDEA																																							
		(C) MUNICIPIO																																							
		(D) DEPARTAMENTO																																							
		(E) COORDENADAS								(F) ACCESO				SI				NO				(G) CLIMA				SI				NO											
		E.a LATITUD								F.a AÉREO				SI				NO				F.d RUTA ALTERNATIVA				G.a DESPEJADO				SI				NO							
		E.b LONGITUD								F.b TERRESTRE				SI				NO				G.b NUBLADO				SI				NO											
		F.c ACUÁTICO								SI				NO				G.c VIENTO FUERTE				SI				NO															
		G.d LLUVIA FUERTE								SI				NO				G.e ACTIVIDAD ELÉCTRICA				SI				NO															
		G.f VIENTO MODERADO								SI				NO																											
REGISTRO DE DAÑOS Y AFECTACIÓN A PERSONAS. HABILITACIÓN DE ALBERGUES	3	(A) PERSONAS								(B) FAMILIAS								(C) ALBERGUE 001								ALBERGUE 002															
		A.a RIESGO								A.g DESAPARECIDAS								B.a RIESGO								C.a NOMBRE DEL ALBERGUE:								C.a NOMBRE DEL ALBERGUE:							
		A.b AFECTADAS								A.h HERIDAS								B.b AFECTADAS																							
		A.c DAMNIFICADAS								A.i FALLECIDAS								B.c DAMNIFICADAS								C.b ENCARGADO								C.b ENCARGADO							
		A.d ATENDIDAS								A.j HOSPITALIZADOS								B.d ATENDIDAS								C.c TELEFONO								C.c TELEFONO							
		A.e EVACUADAS																B.e EVACUADAS								C.d INSTITUCION								C.d INSTITUCION							
		A.f ALBERGADAS																B.f ALBERGADAS								C.e CAPACIDAD								C.e CAPACIDAD							
REQUERIMIENTOS, INSTITUCIONES PRESENTES Y ACCIONES REALIZADAS	4	(A) ALIMENTOS								(B) OTROS								(C) EVALUACIÓN								(D) ACCIONES REALIZADAS															
		A.a AGUA PURA								B.a ENERGÍA ELÉCTRICA								C.a EVALUACIÓN ESTRUCTURAL																							
		A.b ESPONJAS								B.b RED DE AGUA POTABLE								C.b EVALUACIÓN DE RIESGO																							
		A.c FRAZADAS TIPO PONCHO								B.c AMBULANCIAS								C.c EVALUACIÓN MATPEL																							
		A.d RACIÓN FAMILIAR								B.d HOSPITAL MÓVIL								(E) INSTITUCIONES PRESENTES																							
		A.e RACIÓN FRÍAS								B.e EQUIPO BREC																															
										B.f ERI SE-CONRED																															
										B.g SEGURIDAD PÚBLICA																															
										B.h UNIDAD CONTRA INCENDIOS																															

Figura 23. Formulario de Evaluación de Daños y Análisis de Necesidades (EDÁN) para Android

B. Formulario de Evaluación Preliminar de Situación



Fecha de evaluación	/ /20
Código	

EVALUACIÓN PRELIMINAR DE SITUACIÓN

1.0 INFORMACIÓN GENERAL																									
1.01 Comunidad y/o dirección																									
1.02 Municipio						1.03 Departamento																			
1.04 Latitud						1.04 Longitud																			
1.05 Lugar donde se tomo la coordenada																									
1.06 ¿Cómo llegar a la comunidad?																									
Asfalto		Kms.		Terracería		Kms.		Vereda		Kms.		Aéreo		Kms.		Acuático		Kms.		Acceso		Fácil		Difícil	
1.07 Evaluación solicitada por												Institución				Teléfono									
1.08 Cantidad de Infraestructura en riesgo												1.09 Cantidad de viviendas en la comunidad													
1.10 Instituciones presentes en la comunidad																									
1.11 Nombre de la persona que acompaña en la comunidad						Cargo						Teléfono													
1.11 Nombre de la persona que acompaña institucionalmente						Cargo						Teléfono													
2.0 ANTECEDENTES																									
Fecha		Evento										Acciones													
3.0 CARACTERÍSTICAS DEL SITIO																									
3.01 Tipo de amenaza		Hidrometeorológico		Geológico		Volcánico		Antropogénico o socioparticipativo		Mixto															
Especifique																									
3.02 Pendiente del terreno		0.00 - 2 Grados		2.01 - 15 Grados		15.01 - 30 Grados		30.01 - 45 Grados		Mayor a 45 grados															
3.03 Tipo de suelo		Arcilloso		Arenoso		Rocoso		Mixto (especifique)																	
3.04 Uso del suelo																									
3.05 Infraestructura existente		Iglesias		Escuela		Energía eléctrica		Servicios de salud (especifique)																	
		Carreteras		Agua potable		Drenaje sanitario		Otros (especifique)																	
3.06 Distancia a:		Rio		Mts. Nacimiento		Mts. A pie de ladera		Mts. Otros																	
		Quebrada		Mts. Barranco		Mts. A corona de ladera		Mts. (Especifique)																	
4.0 DESCRIPCIÓN GENERAL DEL EVENTO																									
<p>♦</p> <p>♦</p> <p>♦</p> <p>♦</p> <p>♦</p>																									



5.0 DESCRIPCIÓN ESPECÍFICA

• _____

• _____

• _____

• _____

6.0 RECOMENDACIONES

• _____

• _____

• _____

• _____

7.0 AVAL DE PRESENCIA EN LA COMUNIDAD

f. _____	f. _____
firma del evaluador	firma de la persona que acompaña
<input type="text"/>	<input type="text"/>
Nombre del evaluador	Nombre e identificación
<input type="text"/>	<input type="text"/>
Cargo e institución	Cargo
Otras firmas de visto bueno	



8.0 CROQUIS DESCRIPTOR DEL EVENTO					
<p><i>Monedas:</i></p> <div style="display: flex; justify-content: space-around; font-size: small;"> <div> <p>① casa</p> <p>② escuela</p> <p>③ iglesia</p> <p>④ Hospital o centro de salud</p> </div> <div> <p>⑤ Carretera</p> <p>⑥ Río</p> <p>⑦ Camino</p> </div> <div> <p>⑧ Domicilio</p> <p>⑨ Grutas</p> <p>⑩ Monumento</p> </div> </div>					
9.0 REGISTRO FOTOGRÁFICO					
9.01 Foto panorámica	Si		No		¿Qué describe?
9.02 Foto general	Si		No		¿Qué describe?
9.03 Foto de detalles	Si		No	Cantidad	¿Qué describen?

C. Formulario de Evaluación Preliminar de Habitabilidad



Fecha de evaluación	/ / 20
Código	

EVALUACIÓN PRELIMINAR DE HABITABILIDAD

1.0 INFORMACIÓN GENERAL																
1.01 Comunidad y/o dirección																
1.02 Municipio					1.03 Departamento											
1.04 Latitud					1.04 Longitud											
1.05 Lugar donde se tomó la coordenada																
1.06 ¿Cómo llegar a la comunidad?																
Astfalto	Km	Terracería	Km	Vereda	Km	Aéreo	Km	Acuático	Km	Acceso	Fácil	Difícil				
1.07 Evaluación solicitada por					Institución			Teléfono								
1.08 Cantidad de viviendas a ubicar en el terreno																
1.09 Instituciones presentes en la comunidad																
1.10 Nombre de la persona que acompaña en la comunidad					Cargo		Teléfono									
1.10 Nombre de la persona que acompaña institucionalmente					Cargo		Teléfono									
2.0 ANTECEDENTES																
Fecha	Evento						Acciones									
3.0 CARACTERÍSTICAS DEL SITIO																
3.01 Área del terreno			m²		3.02 Área a utilizar			m²		3.03 Área para servicios		m²				
3.04 Pendiente del terreno		0.00 - 2 grados		2.01 - 15 grados		15.01 - 30 grados		30.01 - 45 grados		Mayor a 45 grados						
3.05 Tipo de suelo		Arcilloso		Arenoso		Rocoso		Misto (especifique)								
3.06 Uso actual del terreno																
3.07 Infraestructura si existente		Iglesias		Escuela		Energía eléctrica		Servicios de salud (especifique)								
		Carreteras		Agua potable		Drenaje sanitario		Otros (especifique)								
3.08 Distancia a:		Río		m		Nacimiento		m		A pie de ladera		m				
		Quebrada		m		Barranco		m		A corona de ladera		m (Especifique)				
4.0 CRITERIOS PARA ESTIMAR LA HABITABILIDAD																
4.01 Porcentaje del terreno			4.03 Existen deslizamientos dentro del terreno					4.05 ¿Existen áreas susceptibles a inundación dentro del terreno?								
Menor a 5 grados			%		SI		No		Cantidad observada		SI		No		% del terreno afectado	
Entre 5 y 15 grados			%		SI		No		4.04 Si existen deslizamientos		SI		No		% del terreno afectado	
Mayor a 15 grados			%		SI		No		Describe las dimensiones promedio en metros *		SI		No		% del terreno afectado	
4.02 Especificar según condición			4.05 Presencia de indicadores de deslizamiento					SI		No		MP Afectados		4.06 Si existen áreas con record de inundación ¿Qué nivel alcanzó?		
			Agrietamientos en el terreno										Menor a 0.5 metros		1.60 m.	
			Nacimiento de agua dentro del terreno										entre 0.5 y 1 metros		1.00 m.	
			Arboles o cercos torcidos										entre 1 y 1.5 metros		0.60 m.	
			Grietas en viviendas o infraestructura dentro del terreno										Mayor a 1.5 metros		0.	
			Ruptura de tubería de drenaje o agua potable													



5.0 CONDICIONANTES DE NO HABITABILIDAD		Si	No
5.01 El terreno fue parcial o totalmente afectado por flujos de lodo y deslizamientos, tanto escarpes como depósitos.			
5.02 El terreno es afectado por ríos que cambiaron su cauce, aun y cuando se halla restablecido el flujo al cauce original (cauces migratorios).			
5.03 El terreno se encuentra sobre abanicos aluviales.			
5.04 El terreno está a menos de 5 veces la altura de taludes verticales que superan los 2 metros de altura.			
5.05 El terreno esta sobre o bajo la ladera que presenta grietas o gradas en el terreno, indicativo de deslizamientos activos.			
5.06 El terreno a urbanizar se encuentra dentro o bajo la influencia de deslizamientos antiguos o recientes.			
5.07 El terreno presentó históricamente inundaciones con profundidades iguales o mayores a 1.50 metros.			
5.08 El terreno se encuentra dentro de la planicie de inundación (área equivalente a 5 veces el ancho del cauce).			
5.09 El terreno total o parcialmente se encuentra sujeto a áreas de anegamiento, encharcamiento o pantanos.			

6.0 ESTIMACIÓN PRELIMINAR DE HABITABILIDAD

El terreno es habitable en su totalidad si se presentan las siguientes condiciones:
El terreno tiene una pendiente menor o igual a 15 grados, no presenta indicadores de ser afectado por deslizamientos, flujos de lodo o inundaciones.

El terreno es parcialmente habitable o habitable con medidas de mitigación dadas las condiciones:
*En parte del terreno existen pendientes superiores a los 15 grados
 En parte del terreno hay indicadores de deslizamientos activos o inactivos
 En parte del terreno hay registro de deslizamientos históricos
 Existen zarzones, quebradas, nacimientos, ríos o cuerpos de agua que afecten parte del terreno*

El terreno es considerado no habitable en los siguientes casos:
*Si la totalidad del terreno tiene pendientes que superen los 15 grados,
 Si es afirmativo alguno de los incisos de la sección 5.
 Si el nivel de inundación es igual o mayor a 1.5 metros*

CONCLUSIONES: _____

7.0 AVAL DE PRESENCIA EN LA COMUNIDAD

<p>f. _____</p> <p>firma del evaluador</p> <p>_____</p> <p>Nombre del evaluador</p> <p>_____</p> <p>Cargo e institución</p>	<p>f. _____</p> <p>firma de la persona que acompaña</p> <p>_____</p> <p>Nombre e identificación</p> <p>_____</p> <p>Cargo</p>
<p>Otras firmas de visto bueno</p>	



8.0 CROQUIS DEL TERRENO					
<p><i>Monedas:</i></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> <p> Casa</p> <p> Escuela</p> <p> Iglesia</p> <p> Hospital o centro de salud</p> </div> <div style="width: 30%;"> <p> Carretera</p> <p> Río</p> <p> Cercado</p> <p> Frontera</p> </div> <div style="width: 30%;"> <p> Drenaje</p> <p> Pozo</p> <p> Pozo</p> <p> Pozo</p> </div> </div>					
9.0 REGISTRO FOTOGRÁFICO					
9.01 Foto panorámica	Si	No	¿Qué describe?		
9.02 Foto general	Si	No	¿Qué describe?		
9.03 Foto de detalles	Si	No	Cantidad		¿Qué describen?

D. Formulario de Evaluación Rápida de Daño en Edificaciones

DM2011_V2

EVALUACIÓN RÁPIDA DE DAÑO EN EDIFICACIONES

Evaluación No.: _____ de _____

Cabezas de familia:

Hombre:

Mujer:

Cédula o DPI:

Tel.:

Cédula o DPI:

Tel.:

UBICACIÓN DEL INMUEBLE EVALUADO

Departamento: Municipio: Dirección:

Latitud: Longitud: No de Niveles: ☐ Alquilada ☐ Propia ☐ Otros

EVENTO Nombre del evento: Fecha:

Deslizamiento ☐ Inundación ☐ Sismo ☐ Volcánico ☐ Viento fuerte ☐ otros:

DATOS DEL INMUEBLE

Uso del inmueble: ☐ Vivienda ☐ Escuela ☐ Centro de salud ☐ Iglesia ☐ Comercio ☐ Edificio Municipal ☐ Salón comunal

otro:

Material de techo: ☐ Lámina ☐ Losa (terrazza) ☐ Madera ☐ Teja

otro:

Material de muros o paredes: ☐ Block ☐ Ladrillo ☐ Adobe ☐ Concreto

☐ Lámina ☐ Madera ☐ otro:

Material de piso: ☐ Tierra ☐ Cemento líquido ☐ Cemento ☐ Cerámico

CONDICIONES DEL INMUEBLE (Marcar con una "x" la condición más crítica)

Daños en muros (Daño leve) ☐ Daños en techos (Daño moderado) ☐ Colapso parcial (Daño moderado) ☐ Elementos inclinados (Daño moderado) ☐ Edificio inclinado (Daño severo) ☐ Colapso total (Daño severo) ☐

CONDICIONES DEL ÁREA (Marcar con una "x" o dibuje según corresponda)

Ubicación de la infraestructura:

Condición especial:

Corona ☐ Ladera ☐ Pie ☐ Planicie de inundación ☐ Río o quebrada ☐

Condiciones de la estructura en relación al terreno:

Asentamiento diferencial ☐ Corrimiento ☐ Agrietamiento del suelo ☐ Deslizamiento ☐ Derrumbe (caída de mat.) ☐

En caso de inundación: Nivel del agua H en metros:

Fuerza de la corriente: ☐ No perceptible ☐ Medio ☐ Suave ☐ Fuerte

CLASIFICACIÓN RECOMENDADA (Ver criterios hoja 2)

Marcar con una "x" la clasificación de riesgo y la de habitabilidad (Independiente)

Sin Riesgo	En Riesgo	Inhabitable	Habitable parcialmente	Habitable
------------	-----------	-------------	------------------------	-----------

Hoja 1/2

ASPECTOS SOCIALES Y DE SERVICIOS

Familias que habitan la vivienda:

Cantidad total de personas que habitan la vivienda:

Cantidad por grupo etario: Adultos ☐ Niños ☐ Tercera edad ☐

Cantidad por sexo: Hombres ☐ Mujeres ☐

Personas con capacidades especiales:

Medios de vida:

Cantidad de ambientes en la vivienda:

El inmueble actual cuenta con servicios:

☐ Agua entubada ☐ Energía Eléctrica ☐ Drenaje Sanitario ☐ Letrina

Albergados: ☐ Sí ☐ No

¿Dónde?

Posee lote para traslado: ☐ Sí ☐ No

En criterio del evaluador se requiere trasladar la vivienda de sitio: ☐ Sí ☐ No

Lider comunitario: Tel.:

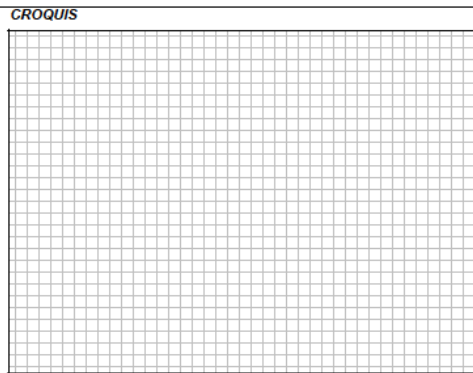
OBSERVACIONES

AVAL DE LA VISITA

f. f.

Nombre del Evaluador: Institución: Fecha:

Hoja 2/2



CRITERIOS


Habitable: Si no se encuentra en riesgo y si la estructura no tiene daño o el daño es leve.

Habitable parcialmente: Si existe una parte de la vivienda en condición habitable.


Inhabitable: Si el daño en la estructura se encuentra de moderado a severo o si el sitio se encuentra en condición de riesgo es decir, si se presenta asentamiento diferencial severo, corrimiento de la vivienda, Agrietamiento en el suelo, deslizamiento, derrumbe, niveles de agua mayor a 1.50 metros, fuerza de la corriente de mediana a fuerte, al borde de una ladera, sobre la ladera, al pie de ladera sobre la ribera del río.

Riesgo: Si se presenta asentamiento diferencial severo, corrimiento de la vivienda, Agrietamiento en el suelo, deslizamiento, derrumbe, niveles de agua mayor a 1.50 metros, fuerza de la corriente de mediana a fuerte, al borde de una ladera, sobre la ladera, al pie de ladera sobre la ribera del río o presenta condiciones especiales desfavorables.

E. Evaluación Rápida de Centros Educativos



EVALUACIÓN RÁPIDA DE CENTROS EDUCATIVOS



1 INFORMACIÓN DE REFERENCIA

Nombre CE: Código UDI: Tel.:

Directoría: Cédula o DPI: Tel.:

Propiedad: ☐ MINEDUC ☐ Municipal ☐ Comunitario ☐ Privado otros Módulo No.: de

2 UBICACIÓN DEL INMUEBLE EVALUADO

Departamento: Municipio: Comunidad, aldea y caserío:

Latitud: Longitud: Dirección:

3 EVENTO

Nombre del evento: Fecha:

Deslizamiento ☐ Inundación ☐ Sismo ☐ Volcánico ☐ Viento fuerte ☐

otros:

4 MATERIALES DE CONSTRUCCIÓN

Material de techo: ☐ Lámina ☐ Losa (terrazza) ☐ Madera ☐ Teja

otro:

Material de muros o paredes: ☐ Block ☐ Ladrillo ☐ Adobe ☐ Concreto

☐ Lámina ☐ Madera otro:

Material de piso: ☐ Tierra ☐ Cemento líquido ☐ Cemento ☐ Cerámico

5 CONDICIONES DEL CENTRO EDUCATIVO (Marcar con una "X")

Daños en muros (Daño leve) ☐ Daños en techos (Daño moderado) ☐ Colapso parcial (Daño moderado) ☐ Elementos inclinados (daño moderado) ☐ Edificio inclinado (daño severo) ☐ Colapso total (Daño severo) ☐

Herramienta basada en Normas de AGIES

6 CONDICIONES DEL ÁREA (Marcar con una "X")

Ubicación de la centro educativo:

Condición especial:

Corona ☐ Ladera ☐ Pie ☐ Planicie de inundación ☐ Río o quebrada ☐

Condiciones de la estructura en relación al terreno:

Asentamiento diferencial ☐ Corrimiento ☐ Agrietamiento del suelo ☐ Deslizamiento ☐ Derrumbe (caída de mat.) ☐

En caso de inundación: Nivel del agua H en metros

Fuerza de la corriente: ☐ No perceptible ☐ Medio ☐ Suave ☐ Fuerte

7 CLASIFICACIÓN RECOMENDADA (Ver criterios hoja 2)

Marque con una "X" la clasificación de riesgo y la de habitabilidad

Sin Riesgo ☐ En Riesgo ☐ Inhabitable ☐ Habitable parcialmente ☐ Habitable ☐

8 DAÑO EN MOBILIARIO, EQUIPO DE COMPUTO Y MATERIAL

	Cantidad D. Moderado	Cantidad D. Severo	Cantidad Sin Daño
Escritorios			
Cátedras			
Pizarrones			
Equipo de Computo			
Material Educativo			

9 ASPECTOS SOCIALES Y DE SERVICIOS

¿Quién construyó el módulo? ☐ FIS/FONAPAZ ☐ Obras publicas ☐ UCEE

☐ Municipalidad ☐ Comunidad otro:

☐ Rural ☐ Urbana Año de construcción: No. de pisos:

Cantidad de ambientes en la escuela: Aulas S.S. Cocina

Dirección Cancha otro:

El inmueble actual cuenta con servicios: ☐ Agua entubada ☐ Letrina ☐ Pozo

☐ Energía Eléctrica ☐ Drenaje Sanitario ☐ Fosa séptica

Nivel que funciona: ☐ Pre-Primaria ☐ Primaria ☐ Básicos ☐ Diversificado

Jornada: ☐ Matutina ☐ Vespertina ☐ Nocturna ☐ Fin de semana

Cantidad educadores máxima por jornada:

Cantidad alumnos máxima por jornada:

Cantidad alumnos por sexo: Hombres Mujeres

Utilizado actualmente como albergue de emergencia: ☐ Si ☐ No

Líder comunitario: Tel.:

10 AVAL DE LA VISITA (Responsable del Centro Educativo)

f. f.

11 DAÑOS EN MUROS PERIMETRALES

Desplome ☐ Grietas ☐

12 DAÑOS EN PISOS

Hundimiento o grietas en piso ☐

13 CROQUIS (dibujar e identificar módulos y su entorno)

CRITERIOS

Habitable: Si no se encuentra en riesgo y si la estructura no tiene daño o el daño es leve.

Habitable parcialmente: Si existe una parte de la escuela en condición habitable.

Inhabitable: Si el daño en la estructura se encuentra de moderado a severo o si el sitio se encuentra en condición de riesgo es decir, si se presenta asentamiento diferencial severo, corrimiento de la escuela, Agrietamiento en el suelo, deslizamiento, derrumbe, niveles de agua mayor a 1.50 metros, fuerza de la corriente de mediana a fuerte, al borde de una ladera, sobre la ladera, al pie de ladera sobre la ribera del río.

Riesgo: Si se presenta asentamiento diferencial severo, corrimiento de la escuela, agrietamiento en el suelo, deslizamiento, derrumbe, niveles de agua mayor a 1.50 metros, fuerza de la corriente de mediana a fuerte, al borde de una ladera, sobre la ladera, al pie de ladera sobre la ribera del río o presenta condiciones especiales desfavorables.

Nombre del Evaluador: Institución: Fecha:

Hoja 2/2

X. GLOSARIO

- **API:** Interfaz de Programación de Aplicaciones por sus siglas en inglés, un API es un conjunto de funciones ofrecidos por una librería de software para ser utilizado por otro software como una capa de abstracción.
- **Aplicación Híbrida:** una aplicación híbrida en el contexto de computación móvil es la que combina elementos de aplicaciones nativas con aplicaciones web.
- **Biyectividad:** cualidad de una función en la cual todos los elementos del conjunto de salida tienen una imagen distinta en el conjunto de llegada y a cada elemento del conjunto de llegada le corresponde un elemento del conjunto de salida.
- **Cola:** estructura de datos en la cual los objetos se obtienen en el mismo orden en el que se ingresaron a la misma.
- **Marco de trabajo:** estructura tecnológica con artefactos o módulos de software que pueden servir de base para la organización y desarrollo de software.
- **Lenguaje de Marcado:** forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.
- **Librería de Software:** conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.
- **Objeto:** un objeto es una unidad dentro de un programa de computadora que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución.
- **Servicio del Sistema Operativo:** programa o aplicación que se ejecuta directamente en el Sistema Operativo, con la diferencia de que corren en segundo plano y no como una aplicación con interfaz gráfica.
- **Tarjeta SIM:** es una tarjeta inteligente desmontable usada en teléfonos móviles y módems que almacenan de forma segura la información necesaria para conectar un dispositivo a la red telefónica.
- **Transacción de Base de Datos:** conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.
- **XHTML:** lenguaje estándar de definición de documentos de hipertexto, definido en base al estándar XML.

- **XML:** Extensible Markup Language. Metalenguaje que permite la definición de lenguajes basados en un sistema de etiquetado.
- **XSLT:** estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.