



Framework eMobc HTML5

Manual del Usuario

Versión 0.1

AVISO LEGAL

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a."**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b."**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(g) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c."**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

d."**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, Noncommercial, ShareAlike.

e."**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

f."**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

g."**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work

relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

h. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

i. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

j. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. **Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved, including but not limited to the rights described in Section 4(e).

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as

required by Section 4(d), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(d), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under: (i) the terms of this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-NonCommercial-ShareAlike 3.0 US) ("Applicable License"). You must include a copy of, or the URI, for Applicable License with every copy of each Adaptation You Distribute or Publicly Perform. You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License. You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. You may not exercise any of the rights granted to You in Section 3 above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation. The exchange of the Work for other copyrighted works by means of digital file-sharing or otherwise shall not be considered to be intended for or directed toward commercial advantage or private monetary compensation, provided there is no payment of any monetary compensation in connection with the exchange of copyrighted works.

d. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and, (iv) consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(d) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

e. For the avoidance of doubt:

i. Non-waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. Waivable Compulsory License Schemes. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License if Your exercise of such rights is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c) and otherwise waives the right to collect royalties through any statutory or compulsory licensing scheme; and,

iii. Voluntary License Schemes. The Licensor reserves the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers

voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License that is for a purpose or use which is otherwise than noncommercial as permitted under Section 4(c).

f.Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING AND TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THIS EXCLUSION MAY NOT APPLY TO YOU.

6.Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a.This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b.Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

- a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
- b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
- c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
- d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
- e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
- f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

Índice de contenidos

Índice de contenido

AVISO LEGAL.....	2
License.....	2
Índice de contenidos.....	7
1 Descripción general.....	9
1.1 App.xml.....	9
1.2 Levels y NextLevels.....	9
1.3 Funcionamiento de la aplicación.....	10
1.4 Level.....	11
1.4.1 Estructura de las aplicaciones.....	11
1.4.1.1 Next Level.....	11
1.4.1.2 Diferencia entre level y ventana.....	11
1.4.2 Asociación de datos de los levels.....	11
1.5 ApplicationData y AppLevelData.....	14
2 Descripción del Parser.....	15
3 Funcionamiento del EntryPoint.....	16
3.1 Descripción.....	16
3.2 Algunas características.....	16
3.3 Su funcionamiento.....	16
4 Funcionamiento de las Rotaciones.....	17
5 Funcionamiento de los Menús.....	18
5.1 Los menús en la aplicación.....	18
5.2 Como definir los tipos de Menú.....	20
5.3 Adición de menús a la aplicación.....	22
6 Temas y Estilos.....	23
6.1 Procedimiento general.....	23
7 Descripción de Todos los Tipo de Actividades.....	24
7.1 Pantalla de Splash.....	25
7.2 Pantalla de Cover.....	25
7.3 Pantalla de Galería de Imágenes.....	26
7.4 Pantalla de PDF.....	27
7.5 Pantalla QR.....	27
7.6 Pantalla de Explorador Web.....	28
7.7 Pantalla de Lista.....	28
7.8 Pantalla de Vídeo.....	29
7.9 Pantalla de Mapa.....	29
7.10 Pantalla de Búsqueda.....	30
7.11 Pantalla Formulario.....	31
7.12 Pantalla Imagen + Texto.....	31
7.13 Pantalla de Imagen con Zoom.....	32
7.14 Pantalla de Imagen + Lista.....	32
7.15 Pantalla de Multimedia.....	33
7.16 Pantalla de Calendario.....	34
7.17 Pantalla de Quiz.....	35
8 Guía de programación.....	36

8.1 Cómo crear un nuevo tipo de ventana para el proyecto HTML5.....	36
8.1.1 Paso 1: Tipo de actividad.....	36
8.1.2 Paso 2: Crear y definir un xml.....	36
8.1.3 Paso 3: Crear datos de la aplicación.....	36
8.1.4 Paso 4: Añadir parser.....	36
8.1.5 Paso 5: Añadir Activity.....	36

Índice de tablas

Tabla 1: Tipos de Pantallas Soportadas por eMobic.....	25
--	----

Índice de Figuras

Figura 1: Esquema General de eMobic.....	9
Figura 2: Secuencia de Ejecución de una aplicación eMobic.....	10
Figura 3: Flujo de generación de una aplicación eMobic.....	14
Figura 4: Rotaciones de la Aplicación.....	17
Figura 5: Menús Top y Bottom.....	18
Figura 6: Side Menu.....	19
Figura 7: Context Menu.....	19
Figura 8: Tamaño de los Menu.....	20
Figura 9: Ejemplo de la Pantalla de Cover.....	25
Figura 10: Ejemplo de la Pantalla de Galería de Imágenes.....	26
Figura 11: Ejemplo de la Pantalla de PDF.....	27
Figura 12: Ejemplo de la Pantalla de Explorador Web.....	28
Figura 13: Ejemplo de la Pantalla de Lista.....	29
Figura 14: Ejemplo de Pantalla de Vídeo.....	29
Figura 15: Ejemplo de la Pantalla de Mapa.....	30
Figura 16: Ejemplo de la Pantalla de Búsqueda.....	31
Figura 17: Ejemplo de la Pantalla de Formulario.....	31
Figura 18: Ejemplo de la Pantalla de Image con Texto.....	32
Figura 19: Ejemplo de la Pantalla de Lista con Image.....	33
Figura 20: Ejemplo de la Pantalla de Multimedia.....	33
Figura 21: Ejemplo de la Pantalla de Calendario.....	34
Figura 22: Ejemplo de la Pantalla de Quiz.....	35

1 Descripción general

eMobic es un framework que permite generar aplicaciones móviles definidas desde un panel de control o manualmente para distintas plataformas: Android, iOS y HTML5.

El framework está basado en archivos XML. Una vez se ha definido la aplicación que se quiere crear, éste genera una serie de archivos XML que describen todas las propiedades, pantallas, y funciones que tendrá la aplicación.

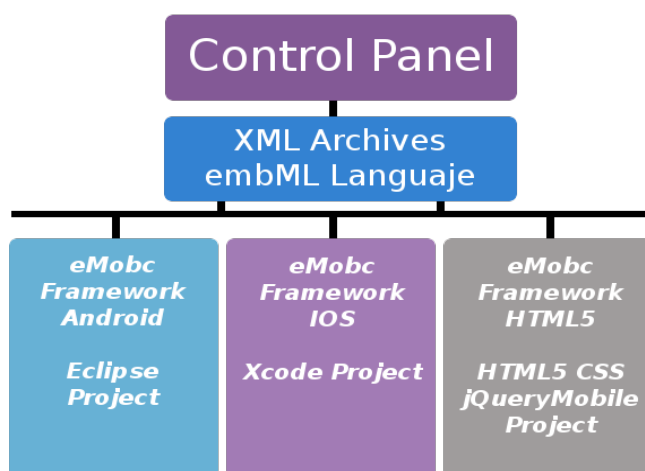


Figura 1: Esquema General de eMobic.

Cada plataforma dispone de un proyecto base donde se almacenarán estos archivos xml. Durante la ejecución de la aplicación, son parseados a medida que son necesarios para ir generando la aplicación a partir de la información recopilada.

El código necesario para ejecutar la aplicación ya está disponible en los proyectos base. Tan solo es necesario añadir los archivos xml y compilar el proyecto para obtener una aplicación funcional.

1.1 App.xml

El archivo xml principal es el app.xml (Ver application data). Toda la información general que se aplica a la aplicación está en este archivo como el tipo de publicidad, las rotaciones soportadas, los menús de navegación (ver menús) o el Entry Point (ver Entry Point).

El EntryPoint básicamente es la pantalla que se mostrará en primer lugar (por defecto será la portada).

1.2 Levels y NextLevels

Los XML definen tanto la aplicación en general (app.xml) como cada una de sus pantallas. Estas pantallas están estructuradas en el framework como levels. Un level es una ventana de la aplicación. Es de un tipo concreto y tiene unos datos asociados.

Todos los levels que contiene una aplicación estarán definidos en su app.xml.

Las acciones dentro de las pantallas de la aplicación, como pulsar un botón o una imagen, pueden tener asociados un Next Level. Un Next Level es la próxima ventana que se mostrará cuando el usuario realice la acción a la que está asociado. (ver Levels)

```
<levels>
  <!-- web -->
  <level>
    <levelId>web</levelId>
    <levelTitle>Ventana Web</levelTitle>
    <levelFile>web.xml</levelFile>
    <levelType>WEB_ACTIVITY</levelType>
  </level>
```

1.3 Funcionamiento de la aplicación

El funcionamiento de una aplicación construida con el framework se basa principalmente en parseado de los archivos XML que definen la aplicación. El primer archivo que se parsea es app.xml. Se generarán todos los datos de la aplicación.

Una vez la portada o el entry point sea visible, cada acción del usuario puede llevar a nuevas ventanas (Next Level) o a salir de la aplicación.

Cuando se realiza una acción que tiene asociado un **NextLevel**, la aplicación busca el archivo de datos xml asociado a ese **NextLevel** y lo parsea para recuperar la información y crear la pantalla que se va a mostrar. (Ver Levels).

Conviene diferenciar, tanto en Android como en iOS, la creación de la aplicación desde el panel de control del funcionamiento. El panel de control es el que genera los archivos xml y el proyecto en Android e iOS es el que va construyendo, a partir de esos xml, la aplicación en tiempo de ejecución.

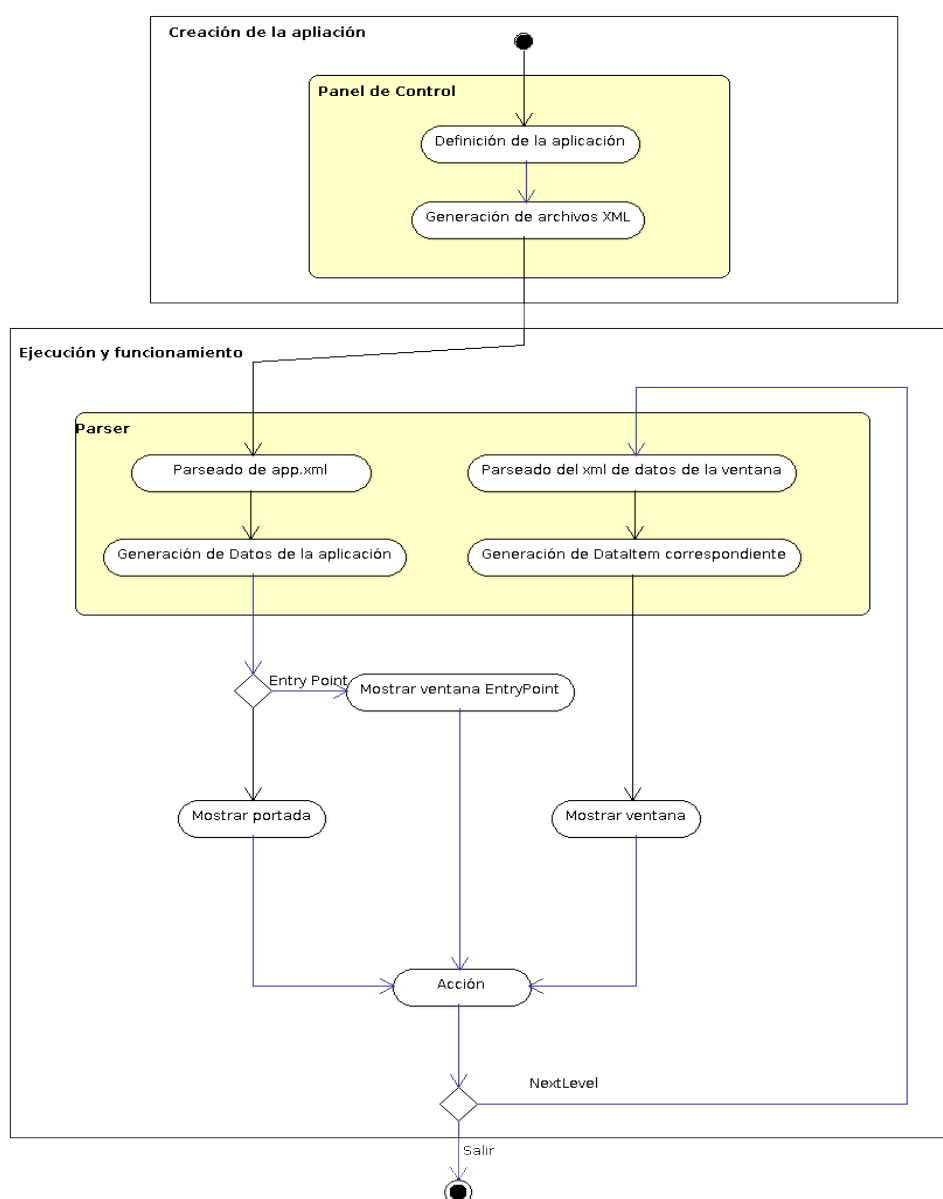


Figura 2: Secuencia de Ejecución de una aplicación eMobic.

1.4 Level

1.4.1 Estructura de las aplicaciones

Las aplicaciones que genera el framework se estructuran en base a **Levels** (niveles). Un level es un tipo de ventana que tiene la aplicación. Por esa razón se incluyen todos los levels de una aplicación en el app.xml.

1.4.1.1 NextLevel

Cuando en una ventana tiene un botón, y al pulsarlo tiene asociado un enlace a una nueva ventana, decimos que tiene un **NextLevel** asociado a la acción de pulsar el botón.

Un Next Level es el level que se creará cuando se realiza una acción. Para definir un Next Level es necesario tanto el level que queremos crear, como los datos asociados a él que la aplicación tendrá que cargar.

Portada.xml

```
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web1</nextLevelDataId>
  </nextLevel>
</button>
```

1.4.1.2 Diferencia entre level y ventana

Un level no es una de todas las pantallas que tendrá la aplicación. Si una aplicación tiene tres tipos de ventana diferentes (por ejemplo: calendario, lista con imágenes y lector pdf) tendrá tres levels definidos en su app.xml. Sin embargo, otra aplicación puede tener en total 3 ventanas, todas ellas de tipo contenedor web y tan solo tendrá un level declarado en su app.xml.

Para diferenciar cada una de estas 3 ventanas entre sí, porque previsiblemente queremos que contengan diferente información, se utilizan los datos asociados al level, definidos mediante el tag “nextLevelDataId” de la llamada al Next Level (ver figura 1).

```
<levels>
  <!-- web -->
  <level>
    <levelId>web</levelId>
    <levelTitle>Ventana Web</levelTitle>
    <levelFile>web.xml</levelFile>
    <levelType>WEB_ACTIVITY</levelType>
  </level>
```

1.4.2 Asociación de datos de los levels

Para identificar cada level y sus datos se utilizan identificadores de tipo String. Un level tiene un identificador único y un archivo xml de datos asociado. Cada vez que se quiere crear un Next Level se utiliza el identificador del level (que está definido en el app.xml) y el identificador de los datos.

Cada vez que se llama a un Next Level desde la aplicación, se consulta la información del app.xml por levelId y se leen los datos identificados por el tag “nextLevelDataId” dentro del archivo xml especificado en el tag “levelFile”.

Un archivo de datos xml puede tener varios datos dentro. Cada uno tiene un identificador único al que se puede hacer referencia en cada llamada a un Next Level.

A continuación se muestra un ejemplo para clarificar el funcionamiento de los levels y los datos asociados. Continuando con el escenario anterior, en el que se tienen tres ventanas de tipo contenedor web, vamos a ver cómo se referencian entre sí los tags y los archivos xml.

Primero se presentarán cómo son nuestros archivos xml y qué significa cada tag dentro de ellos.

Este es un fragmento de portada.xml. Es la definición de una ventana de portada que tiene, entre otras cosas, un botón de título Navegador con un Next Level asociado a una ventana de contenedor web.

La etiqueta <nextLevelLevelId> tiene el identificador del level al que se irá si se pulsa el botón

La etiqueta <nextLevelDataId> tiene el identificador de los datos que el level debe cargar cuando se cree.

```
portada.xml
...
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web1</nextLevelDataId>
  </nextLevel>
</button>
...
```

Este es un fragmento del archivo app.xml donde se declaran los levels de nuestra aplicación.

Como solo tendrá ventanas de tipo web, solo aparece un level. La etiqueta <levelId> contiene el identificador del level.

La etiqueta <levelFile> contiene el archivo xml que almacena los datos del level.

La etiqueta <levelType> contiene el tipo de ventana que corresponde al level (en este caso web).

```
app.xml
...
<levels>
  <!-- web -->
  <level>
    <levelId>web</levelId>
    <levelTitle>Ventana Web</levelTitle>
    <levelFile>web.xml</levelFile>
    <levelType>WEB_ACTIVITY</levelType>
  </level>
</levels>
...
```

Este es un fragmento del archivo de datos del level web. Por simplicidad se ha reducido el número de ventanas web de tres a dos.

La etiqueta <dataId> contiene el identificador de los datos dentro del archivo xml de datos.

La etiqueta <local> establece si hay que buscar una dirección en la web o en un archivo local.

La etiqueta <webUrl> contiene la dirección web o el archivo html local que cargará el contenedor web.

(Para más información de los tags de los archivos xml consulta Archivos xml).

Una vez explicados todos archivos xml que van a intervenir en el ejemplo, se mostrarán las interacciones entre ellos.

```
web.xml
...
<levelData>
  <data>
    <dataId>web1</dataId>
    <headerImageFile>images/web.png</headerImageFile>
    <headerText>Ventana Web</headerText>
    <local>true</local>
    <webUrl>archivo.html</webUrl>
  </data>
  <data>
    <dataId>web2</dataId>
    <headerImageFile>images/web.png</headerImageFile>
    <headerText>Ventana Web</headerText>
    <local>false</local>
    <webUrl>www.google.es</webUrl>
  </data>
</levelData>
...
```

Actualmente estamos en el escenario en el que el usuario pulsa un botón de la portada que tiene asociado un Next Level a una ventana web que cargará un archivo html local (archivo.html). En ese momento se busca el level en el app.xml por la etiqueta <nextLevelLevelId>.

portada.xml

```
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web1</nextLevelDataId>
  </nextLevel>
</button>
```

app.xml

```
<levels>
  <!-- web -->
  <level>
    <levelId>web</levelId>
    <levelTitle>Ventana Web</levelTitle>
    <levelFile>web.xml</levelFile>
    <levelType>WEB_ACTIVITY</levelType>
  </level>
</levels>
```

El level al que se ha hecho referencia dentro del app.xml, tiene asociado un archivo xml de datos en el tag <levelFile>. Se busca ese archivo y dentro de él aquellos datos a los que se hace referencia con el tag <nextLevelDataId> del archivo portada.xml.

portada.xml

```
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web1</nextLevelDataId>
  </nextLevel>
</button>
```

app.xml

```
<levels>
  <!-- web -->
  <level>
    <levelId>web</levelId>
    <levelTitle>Ventana Web</levelTitle>
    <levelFile>web.xml</levelFile>
    <levelType>WEB_ACTIVITY</levelType>
  </level>
</levels>
```

web.xml

```
<levelData>
  <data>
    <dataId>web1</dataId>
    <headerImageFile>images/web.png</headerImageFile>
    <headerText>Ventana Web</headerText>
    <local>true</local>
    <webUrl>archivo.html</webUrl>
  </data>
  <data>
    <dataId>web2</dataId>
    <headerImageFile>images/web.png</headerImageFile>
    <headerText>Ventana Web</headerText>
    <local>false</local>
    <webUrl>www.google.es</webUrl>
  </data>
</levelData>
```

La segunda ventana cargará la dirección web `http://www.google.es` dentro del contenedor web. Hay que incluir en el archivo que define la portada (`portada.xml`) otro botón para que cargue los datos de la nueva ventana.

De esta manera, cada vez que se pulse el segundo botón, se accederá al mismo archivo de datos xml (`web.xml`) pero al nuevo identificador de datos que ahora contendrá la información para que se cargue google.

```
portada.xml
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web1</nextLevelDataId>
  </nextLevel>
</button>
<button>
  <buttonTitle>Navegador</buttonTitle>
  <buttonFileName>images/buttonw.png</buttonFileName>
  <nextLevel>
    <nextLevelLevelId>web</nextLevelLevelId>
    <nextLevelDataId>web2</nextLevelDataId>
  </nextLevel>
</button>
```

1.5 ApplicationData y AppLevelData.

Hasta ahora hemos visto que para consultar levels y sus archivos de datos xml, se accede al `app.xml`. Esto no es del todo cierto. Cuando se necesita acceder a los datos contenidos en un archivo xml, primero se parsea el archivo y se transforma en un objeto que al que luego se puede acceder mediante código.

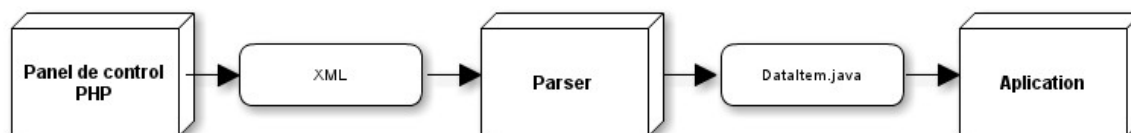


Figura 3: Flujo de generación de una aplicación eMobic.

Como puede verse en la figura 2, los XML los genera el panel de control pero la aplicación en sí no accede a ellos. Necesita un objeto que almacene esa información y al que se pueda acceder desde código. Aunque realmente el parser es parte de la aplicación, trabaja como una entidad independiente y cuando es necesario genera estos objetos que la aplicación utilizará para construir las ventanas. (Ver parser).

Aquí es donde entran en juego los **ApplicationData** y los **AppLevelData**.

ApplicationData : es el objeto que almacenará la información del `app.xml`. Previsiblemente habrá solo uno, aunque es posible generarlo manera dinámica en algunas ventanas como en los formularios.

AppLevelData : es el objeto que almacena la información de cada uno de los archivos xml. A su vez y como cada archivo de datos xml puede tener un conjunto de datos demasiado complejos para almacenarse en un tipo simple, es posible que sea necesario utilizar estructuras de datos complejas. Para estos nuevos tipos de datos se utilizan los **DataItem**.

2 Descripción del Parser

Cada uno de los parser está ubicado en la carpeta parser, y lleva el nombre de la actividad. Por ejemplo, para la actividad FORM_ACTIVITY existe un archivo \$EMOBC_HTML5_HOME/parser/form_activity.php el cual será el encargado de parsear el contenido de los archivos xml de la actividad del tipo FORM_ACTIVITY.

A continuación se muestra el código de la función traer_info que debe ser implementada en estos archivos para el caso de la actividad FORM_ACTIVITY.

```
function traer_info($xml, $data){  
    if(!strpos(" ".$xml, "http")){  
        $sx = simplexml_load_file($xml);  
    }else{  
        $content = file_get_contents($xml);  
        $sx = simplexml_load_string($content);  
    }  
    foreach($sx->data as $item){  
        if ($data == $item->dataId) {  
            foreach($item->form as $item2){  
                $actionUrl = $item2->actionUrl;  
                foreach($item2->field as $item3){  
                    $fieldType[] = $item3->fieldType;  
                    $fieldLabel[] = $item3->fieldLabel;  
                    $fieldName[] = $item3->fieldName;  
                    $fieldParam[] = $item3->fieldParam;  
                }  
            }  
            $nextLevelLevelId = $item->nextLevel->nextLevelLevelId;  
            $nextLevelDataId = $item->nextLevel->nextLevelDataId;  
            $total[0] = $nextLevelLevelId;  
            $total[1] = $nextLevelDataId;  
            $total[2] = $actionUrl;  
            $total[3] = $fieldType;  
            $total[4] = $fieldLabel;  
            $total[5] = $fieldName;  
            $total[6] = $fieldParam;  
        }  
    }  
    return $total;  
}
```

3 Funcionamiento del EntryPoint

3.1 Descripción

El EntryPoint es la pantalla que se carga por primera vez al iniciar la aplicación, después de la pantalla SplashActivity.

Es posible pensar, que exista una pantalla de inicio fija (que es al fin y al cabo una portada), y que sea la misma para todas las aplicaciones.

Pero en ocasiones, el usuario necesita contar con una portada predefinida, o que inicie la aplicación en una pantalla en concreto.

El EntryPoint es el principal encargado de éste cometido.

La función principal es especificar el levelId y el dataId de la pantalla que se quiere iniciar, que como ya se explico antes, corresponde a una pantalla específica.

La forma de establecer dicha información se define en el fichero app.xml, concretamente en el campo entryPoint:

```
<entryPoint>
  <pointLevelId></pointLevelId>
  <pointDataId></pointDataId>
</entryPoint>
```

Simplemente hay que definir el levelId, y su correspondiente dataId, y la aplicación iniciará con la pantalla especificada.

Si no se especifica ningún dato en éste campo, es decir, los campos no tienen ningún contenido establecido, la aplicación carga una pantalla básica por defecto, que no es más que una botonera de inicio.

3.2 Algunas características

La pantalla de portada (o como se ha explicado, el EntryPoint), está fuertemente relacionada con los menús, y es por ello, por lo que la pantalla de inicio cuenta con unas restricciones y características únicas, en cuanto a la creación de éstos.

Todos los menús del EntryPoint tienen unas propiedades específicas:

- Las funciones de los menús, no se ven afectadas, por lo que funcionan de igual manera tanto en todas las pantallas, como en la pantalla de EntryPoint.
- No se cuenta con boton **“home”** en ningún tipo de menú para la pantalla de EntryPoint, el sistema lo ignora si el usuario lo hubiese especificado en el xml.
- No se cuenta con boton **“back”** en ningún tipo de menú para la pantalla de EntryPoint, el sistema lo ignora si el usuario lo hubiese especificado en el xml.

Por otro lado se tiene en cuenta si se encuentra en la pantalla de EntryPoint cuando se pulsa el botón “back” del sistema, ya que su función es distinta para el resto de pantallas. Si se encontrase en la pantalla de EntryPoint, el sistema **cerrará la aplicación** si el botón atrás del sistema es pulsado.

3.3 Su funcionamiento

El funcionamiento de la pantalla de EntryPoint, se basa únicamente en el paso de parámetros.

Concretamente en una variable booleana que pasa como extra desde los métodos showNextLevel().

4 Funcionamiento de las Rotaciones

Las rotaciones existen para aportar mas dinamismo o comodidad a la hora de utilizar una aplicación. Hay tipos de pantallas que requieren una mejor vista de sus componentes, y es por ello por lo que se ofrece un sistema de rotaciones personalizado.

Hay que tener en cuenta que rotar una pantalla es simplemente una reestructuración de los componentes, por tanto, ni su número, ni su funcionamiento varía al rotar la pantalla.



Figura 4: Rotaciones de la Aplicación.

Las rotaciones se aplican para todas las pantallas de la aplicación. Por tanto, su propiedad se verá definida en el fichero app.xml.

```
<rotation></rotation>
```

Los valores que puede tomar dicho campo son:

- `portrait`: La pantalla se torna en modo vertical
- `landscape`: La pantalla se torna en modo horizontal
- `both`: Se toman valores del acelerómetro del dispositivo móvil. Por tanto, la pantalla se torna en modo vertical u horizontal dependiendo de la orientación.

Si se insertase un valor desconocido, o vacío, la aplicación definirá la rotación por defecto “both”.

Nota: La versión HTML5 de eMobic NO soporta rotaciones.

5 Funcionamiento de los Menús

5.1 Los menús en la aplicación

La aplicación cuenta con un sistema de menús, para ofrecer acciones directas en cualquier pantalla.

Existen 4 tipos distintos de menús:

- **Top Menu:** Es un tipo de menú *fijo*, que se posiciona en la parte superior de la pantalla. (Ver Figura 5)
- **Bottom Menu:** Es un tipo de menú *fijo*, que se posiciona en la parte inferior de la pantalla. (Ver Figura 5)
- **Side Menu:** Es un tipo de menú *emergente*, que aparece por un lateral de la pantalla, que ofrece una lista de opciones. (Ver Figura 6)
- **Context Menu:** Es un tipo de menú *emergente*, que aparece al pulsar el boton MENU del sistema. (Ver Figura 7)

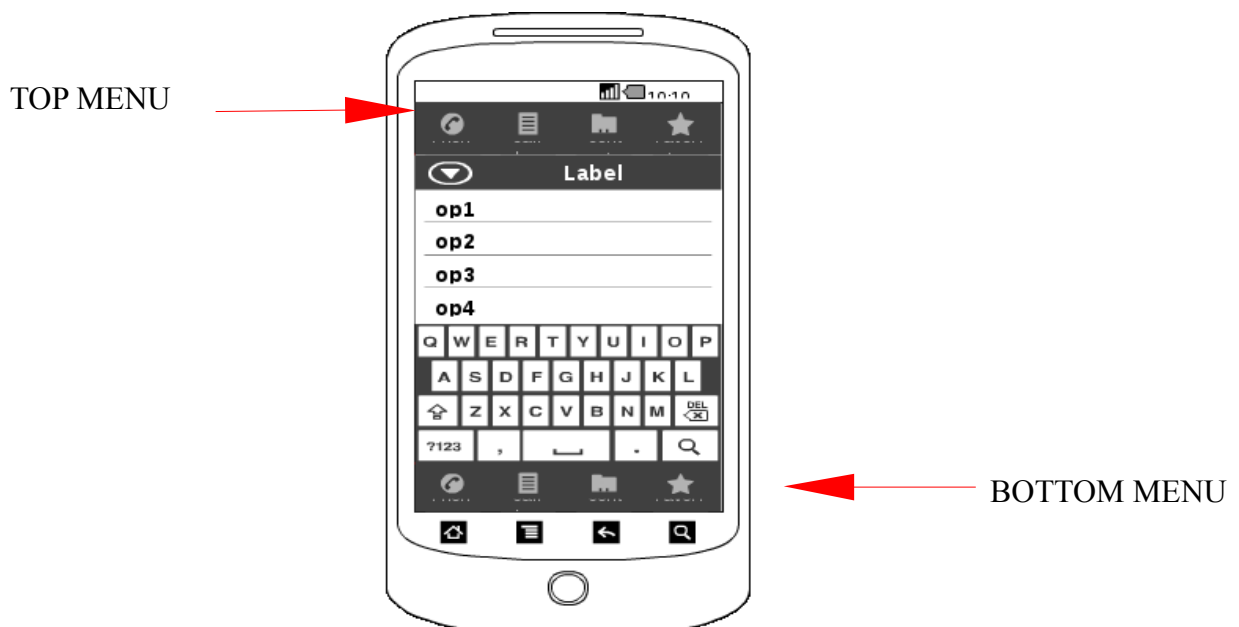


Figura 5: Menús Top y Bottom.

SIDE MENU

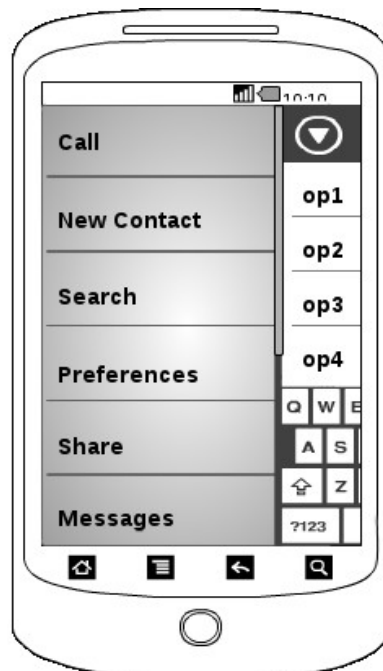


Figura 6: Side Menu.

CONTEXT MENU

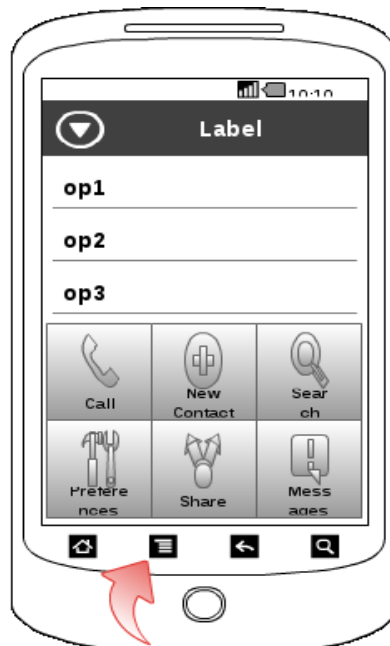


Figura 7: Context Menu.

Los tipos de menu Top Menu y Bottom Menu, son los únicos con propiedad fija, esto quiere decir, que si están definidos, aparecerán fijos en pantalla, reestructurando el resto de componentes de la pantalla. Esos componentes están en un contenedor llamado contentLayout, que es precisamente el que reestructura la pantalla en caso de existir alguno de esos tipos de Menu.

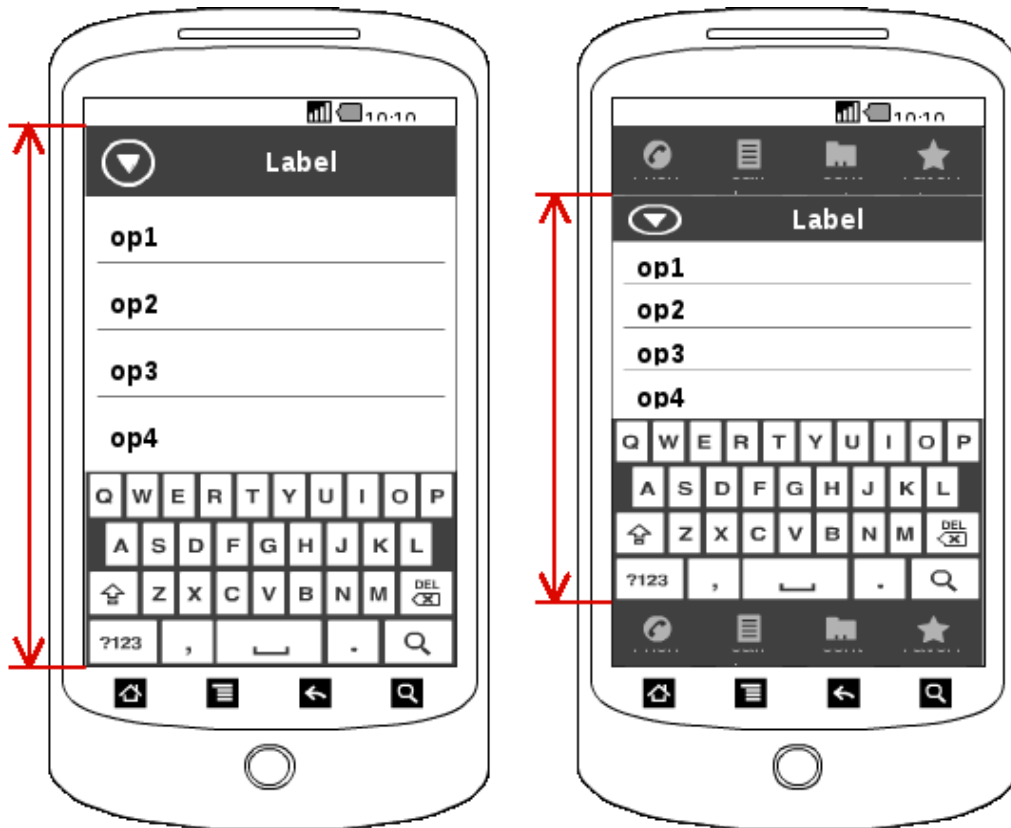


Figura 8: Tamaño de los Menu.

En cambio, los tipos de menú Side Menu y Context Menu, al ser emergentes, no influyen en la reestructuración de los componentes de la pantalla. Además, si alguno de estos menús esta desplegado, si el botón “back” del sistema es pulsado, su contenido volverá a ocultarse.

5.2 Como definir los tipos de Menú

Cada tipo de menú tiene su propio archivo xml, por tanto, al existir 4 tipos distintos de menú, existirán también 4 ficheros xml destinados.

Todos tienen la misma estructura común:

```
<menuActions>
  <action>
    <actionTitle></actionTitle>
    <actionImageName></actionImageName>
    <systemAction></systemAction>
    <leftMargin></leftMargin>
    <widthButton></widthButton>
    <heightButton></heightButton>
    <nextLevel>
      <nextLevelLevelId></nextLevelLevelId>
      <nextLevelDataId></nextLevelDataId>
    </nextLevel>
  </action>
</menuActions>
```

Existe una etiqueta global llamada `menuActions` que engloba toda y cada una de las distintas acciones, con sus correspondientes características.

Una acción (`action`) no es más que la definición para un botón del menú, es decir, es un campo único para cada botón, donde se almacenan los datos tanto de presentación, como de funcionalidad:

- **actionTitle**: nombre de la acción del menú.
- **actionImageName**: nombre de la imagen que presentará ella acción del menú.
- **systemAction**: opción que define si la acción es o no una acción del sistema. Las acciones de sistema son:
 - “back”: acción del sistema para dirigirse a la pantalla anterior.
 - “home”: acción del sistema para dirigirse al EntryPoint.
 - “tts”: acción del sistema para presentar un “Text to Search”
 - “search”: acción del sistema para presentar un “Search”
 - “share”: acción del sistema para presentar un “Share”
 - “map”: acción del sistema para presentar un “Map”
- **leftMargin**: define los márgenes entre las opciones del menú.
- **widthButton**: define el ancho del botón de la acción.
- **heightButton**: define el alto del botón de la acción.
- **nextLevel**: opción que define la pantalla a la que se dirigirá la aplicación al pulsar la opción del menú.

Teniendo en cuenta esta serie de parámetros, hay que tener en cuenta un factor importante.

Cada acción del menú puede o ser una acción de sistema, o abren una nueva activity, nunca las dos cosas, debido a que las propias acciones del sistema abren también nuevas actividades. Visto en un ejemplo:

Acción de Sistema → Back

```
<menuActions>
  <action>
    <actionTitle>Accion1</actionTitle>
    <actionImageName>img1.png</actionImageName>
    <systemAction>back</systemAction>
    <leftMargin>20</leftMargin>
    <widthButton>40</widthButton>
    <heightButton>40</heightButton>
    <nextLevel>
      <nextLevelLevelId></nextLevelLevelId>
      <nextLevelDataId></nextLevelDataId>
    </nextLevel>
  </action>
</menuActions>
```

Acción normal → Abrir nueva ventana

```
<menuActions>
  <action>
    <actionTitle>Accion2</actionTitle>
    <actionImageName>img2.png</actionImageName>
    <systemAction></systemAction>
    <leftMargin>20</leftMargin>
    <widthButton>40</widthButton>
    <heightButton>40</heightButton>
    <nextLevel>
      <nextLevelLevelId>level_1</nextLevelLevelId>
      <nextLevelDataId>data_1_1</nextLevelDataId>
    </nextLevel>
  </action>
</menuActions>
```

Se observa por tanto que

- Si el campo **systemAction** está vacío, el campo **nextLevel** toma los valores de una pantalla específica.
- Si el campo **systemAction** tiene valor (válido), el campo **nextLevel** está vacío.

En el caso de que se introduzca algún dato desconocido, por ejemplo, un **systemAction** no definido, la acción simplemente no tendrá ninguna función.

5.3 Adición de menús a la aplicación

Una vez se han definido los distintos tipos de menú, hay que asignar a la aplicación los menús que se quieren activar.

Ese proceso se centra en el fichero **app.xml**:

```
<menu>
  <topMenu></topMenu>
  <bottomMenu></bottomMenu>
  <contextMenu></contextMenu>
  <sideMenu></sideMenu>
</menu>
```

Simplemente **menu** es un campo en el que se pueden definir si están o no activos los menús explicados anteriormente. En cada una de las 4 etiquetas se asigna el nombre del fichero xml que define un menú. Si no se quiere aplicar algún menú, hay que dejar la etiqueta vacía.

Visto en un ejemplo:

```
<menu>
  <topMenu>menu1.xml</topMenu>
  <bottomMenu></bottomMenu>
  <contextMenu>menu2.xml</contextMenu>
  <sideMenu>menu2.xml</sideMenu>
</menu>
```

Esta definición asigna las características explicadas anteriormente:

- Las pantallas tendrán un menú superior con las características especificadas en el fichero **menu1.xml**
- Las pantallas no contarán con ningún menú inferior.
- Las pantallas tendrán un menú contextual con las características especificadas en el fichero **menu2.xml**
- Las pantallas tendrán un menú lateral con las características especificadas en el fichero **menu2.xml**

6 Temas y Estilos

Los temas y estilos son la base visual en las pantallas. Aportan colores, formas y fondos, que crean una estructura vistosa y agradable al usuario.

La aplicación cuenta con un sistema personalizado de temas y estilos, donde es posible aplicar una temática a un tipo de pantalla, o incluso a una pantalla en concreto.

6.1 Procedimiento general

Inicialmente, al parsear el archivo **app.xml**, se toman dos campos claves: `stylesFileName`, y `formatsFileName`.

Cómo su nombre indica, guardarán el nombre de los ficheros XML que contienen los estilos y los formatos.

Una vez hecho eso, se parsean dichos ficheros XML.

- En el XML de formato, se establecen distintos tipos de formato que existen, por ejemplo, `title1`, `title2`, `text1`, etc...

Cada uno de esos tipos de formato contiene información básica a cerca de estilos de textos, como: `textColor`, `textSize`, `textStyle`, `typeFace`, ...

Tras finalizar ésto, se almacena en un `HashMap` en un objeto de la clase `ApplicationData`.

- En el XML de estilos, se asigna a cada tipo de pantalla un `background`, y además unos campos básicos, referentes a los distintos `Views` de una pantalla, en los que se indica el formato que va a tener. Estos campos son: **header** (cabecera), **footer** (descripción inferior), **basic_text** (textos simples), **selection_list** (títulos de una lista), **o cualquier otro view NO GENERICO, creado por un usuario**.

Por último, como antes, se almacena en otro `HashMap` en un objeto de la clase `ApplicationData`.

- Para aumentar la flexibilidad, el fichero `app.xml` cuenta con una etiqueta llamada `levelFormat`, que indica el estilo específico y único para un `level` de la aplicación, y que actúa de igual forma que lo explicado en el punto anterior.

```
<levelFormat>
  <backgroundFileName>background.png</backgroundFileName>
  <components>header=default;footer=default;</components>
</levelFormat>
```

7 Descripción de Todos los Tipo de Actividades

La Tabla 1 muestra todos los tipos de pantallas soportados por el Framework eMobic y su descripción de funcionalidad.

Tipo de Pantalla	Descripción
Pantalla de Splash	Es la primera pantalla que se muestra de nuestra aplicación. Se muestra para indicar que la aplicación se está cargando. Una vez que desaparece no se puede volver a ella
Pantalla de Cover	Es la pantalla portada. En ella podemos mostrar a todas las opciones y acciones que nuestra aplicación puede soportar.
Pantalla de Galería de Imágenes	Esta pantalla muestra una galería de imágenes al usuario
Pantalla de PDF	Esta pantalla muestra al usuario un lector de pdf
Pantalla QR	Esta pantalla permite al usuario escanear mediante la cámara un código QR
Pantalla de Explorador Web	Esta pantalla muestra al usuario un contenedor Web donde poder visualizar contenido on-line como páginas Web o HTML locales.
Pantalla de Lista	Esta pantalla le muestra al usuario una lista donde. Permite seleccionar entre las opciones que muestra. Contamos con varios formatos de la misma pantalla: <ul style="list-style-type: none">• Lista sólo con texto• Lista con imagen junto al texto
Pantalla de Video	Esta pantalla permite al usuario visualizar un video. El video puede estar en el propio dispositivo o puede ser on-line
Pantalla de Mapa	Esta pantalla muestra al usuario un mapa dependiendo de su localización. El mapa puede contener marcas de sitios preferidos que el usuario puede añadir
Pantalla de Búsqueda	Esta pantalla permite al usuario buscar una porción de texto, georeferencia o imagen dentro del framework
Pantalla Formulario	Esta pantalla permite al usuario crear una nueva pantalla simplemente rellenando los campos necesarios
Pantalla Imagen + Texto	Esta pantalla muestra al usuario una Imagen acompañado por un texto
Pantalla de Imagen con Zoom	Esta pantalla permite mostrar al usuario una imagen aumentada
Pantalla de Imagen + Lista	Esta pantalla nos muestra una lista seleccionable por el usuario, acompañada por una imagen ajena a las celdas.
Pantalla de Multimedia	Esta pantalla permite mostrar al usuario un menú para acceder a las acciones multimedia como pueden ser: <ul style="list-style-type: none">• Acceso a Fotos• Acción de Compartir• Acceso a Videos• Acceso a grabaciones de Voz• etc ...
Pantalla de Calendario	Esta pantalla muestra al usuario un calendario ajeno al calendario del sistema. Este calendario muestra los eventos que la aplicación internamente quiera mostrar.
Pantalla de Quiz	Permite generar tests con varias respuestas y muestra los resultados. También permite generar aventuras interactivas con varios finales que dependen de las decisiones tomadas a lo largo de las preguntas.

Tabla 1: Tipos de Pantallas Soportadas por eMobic.

A continuación se describe cada una de esta pantallas con mayor nivel de detalle

7.1 Pantalla de Splash

7.2 Pantalla de Cover

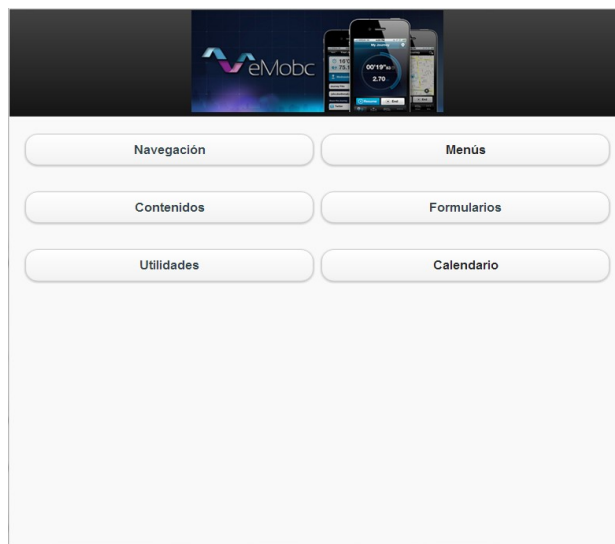


Figura 9: Ejemplo de la Pantalla de Cover

La ventana de portada está diseñada para ser la primera ventana de la aplicación. Habitualmente muestra una lista de botones con acciones básicas en la aplicación o enlaces a otras ventanas secundarias. La ventana de portada básica se compone de una imagen de cabecera y unos botones con Next Level a otras ventanas de la aplicación.

Básicamente es el punto de partida desde donde el usuario puede acceder a todas las funciones de la aplicación.

Es importante saber que toda aplicación tiene que disponer de la Actividad de portada. Es posible modificar la ventana inicial de la aplicación mediante la función entry point (ver entry point), pero la actividad de portada es la que se encarga de gestionarlo (ver más adelante SplashActivity).

7.3 Pantalla de Galería de Imágenes

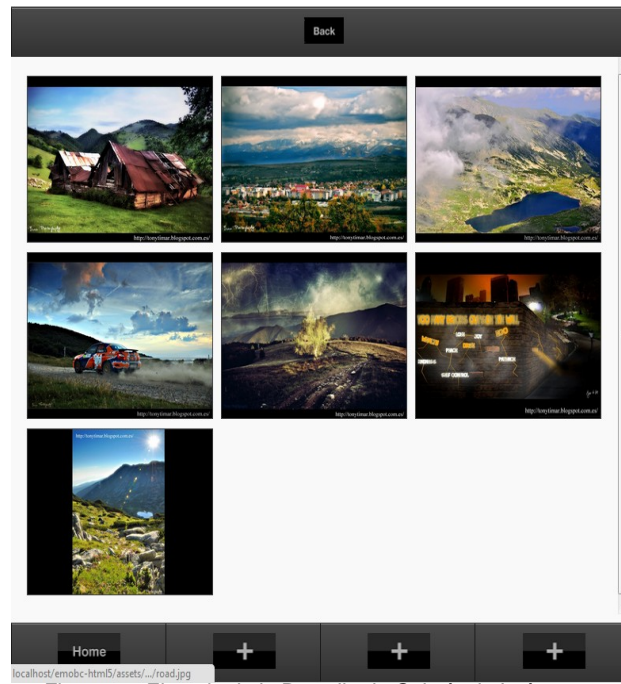


Figura 10: Ejemplo de la Pantalla de Galería de Imágenes.

La galería de imágenes se utiliza para mostrar un conjunto de imágenes seleccionadas en una sola ventana. El usuario puede desplazarse entre las imágenes para verlas.

7.4 Pantalla de PDF

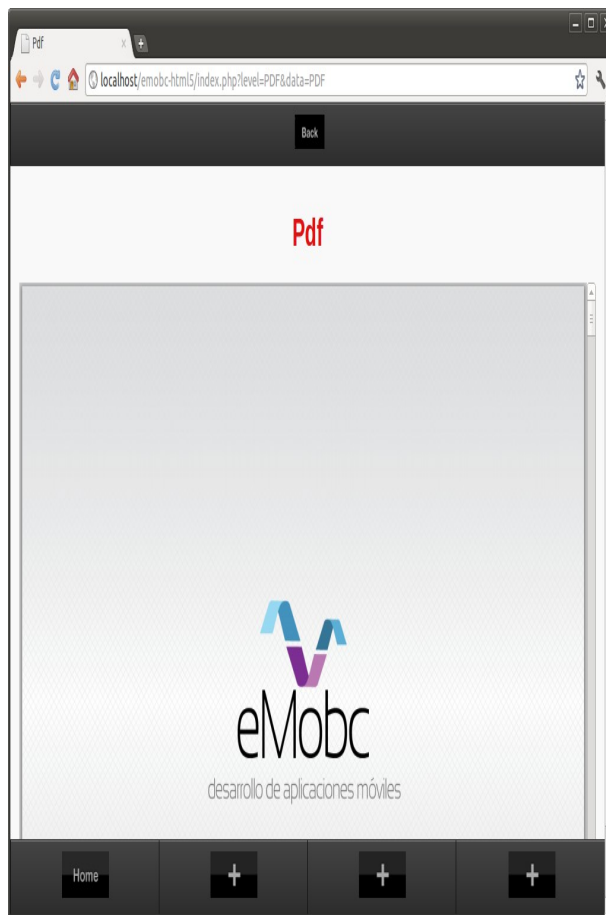


Figura 11: Ejemplo de la Pantalla de PDF.

La ventana de pdf está pensada para mostrar documentos externos (principalmente de archivos pdf) en la aplicación. Para documentos externos, la ventana utiliza Google Docs para mostrar el contenido. Además de pdf, es posible utilizar otros formatos de archivos compatibles con Google Docs.

La herramienta de Google Docs permite descargar el archivo para tenerlo localmente y abrirlo con la aplicación por defecto de lectura de archivos pdf que tenga el usuario instalada en su móvil.

7.5 Pantalla QR

La pantalla QR es aquella que permite al usuario escanear códigos QR. Esta opción sólo está disponible en las versiones nativas del Framework (Android e iOS), por lo tanto en HTML5 parecerá un mensaje avisando que este tipo de pantallas no está disponible.

7.6 Pantalla de Explorador Web



Figura 12: Ejemplo de la Pantalla de Explorador Web.

La ventana permite no solo cargar una dirección web externa sino que puede cargar también archivo html guardados localmente en el proyecto. Además, si quiere realizar una aplicación completamente remota con contenidos web puede utilizar esta ventana.

La carga de archivos html locales está pensada para mostrar fácilmente texto con formato. Usted puede cargar cualquier html pero no se recomienda generar aplicaciones que basen su IU en archivos html ya que puede haber problemas en cómo se muestran los contenidos en diferentes tipos de pantallas.

7.7 Pantalla de Lista

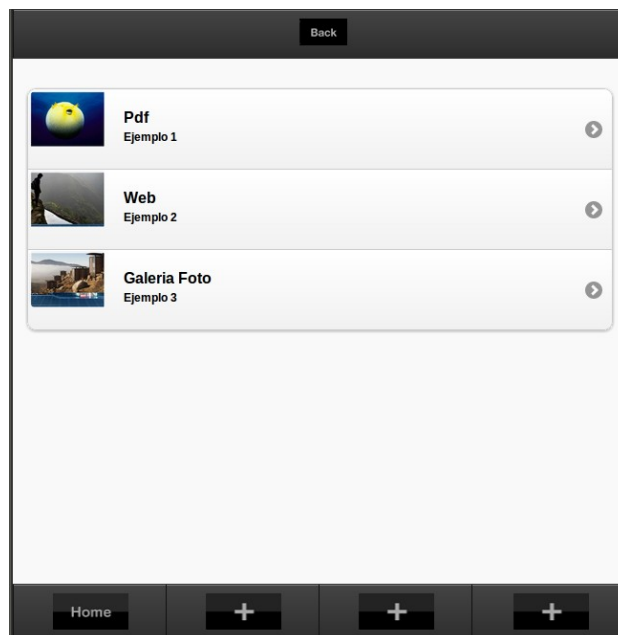


Figura 13: Ejemplo de la Pantalla de Lista.

La ventana de lista muestra una lista con imágenes en cada elemento. Cada elemento de la lista tiene asociado un Next Level y puede llevar a una nueva ventana. Las imágenes de los elementos de la lista pueden ser tanto locales como remotos.

Si las imágenes son remotas, su carga se realiza de manera asíncrona para agilizar el movimiento y desplazamiento de la lista.

7.8 Pantalla de Vídeo

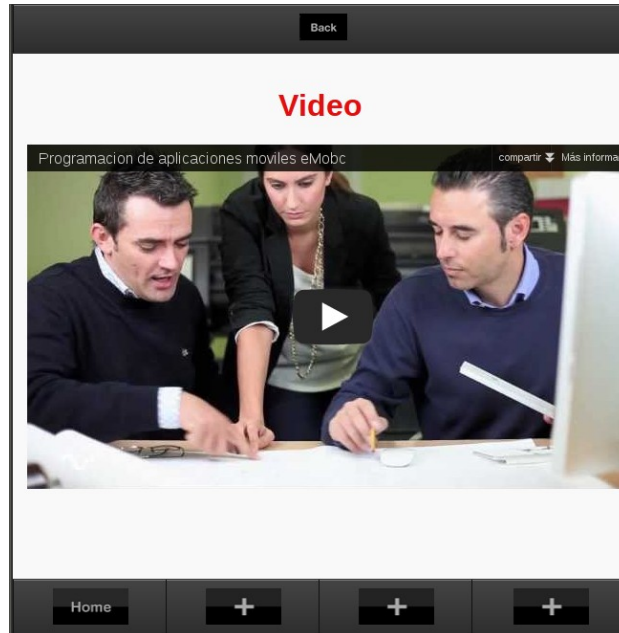


Figura 14: Ejemplo de Pantalla de Vídeo.

La ventana de vídeo permite reproducir vídeo desde una url. El vídeo ocupa la pantalla completa (sin incluir los menús de navegación y banners de publicidad).

7.9 Pantalla de Mapa



Figura 15: Ejemplo de la Pantalla de Mapa.

La pantalla de Mapa muestra un mapa de unas coordenadas específicas, por las que el usuario se puede mover para ver los alrededores a la posición dada.

Además, esta pantalla permite añadir marcas de sitios preferidos o lugares de interés que se cargarán a la vez que se carga el mapa.

7.10 Pantalla de Búsqueda

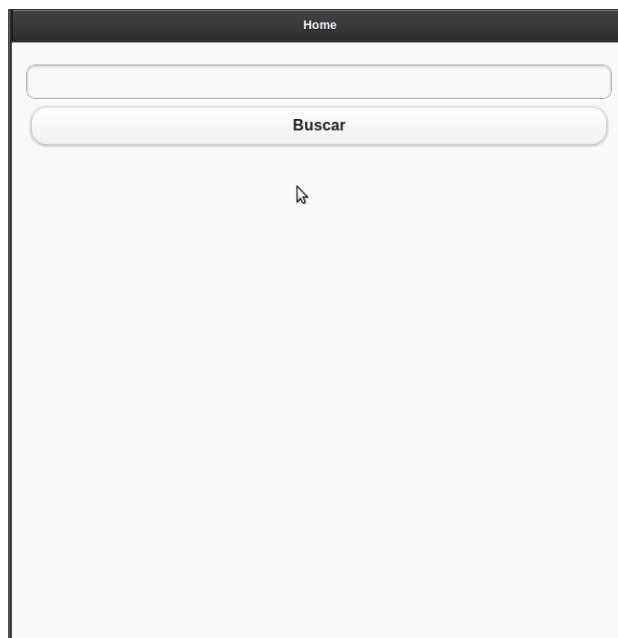


Figura 16: Ejemplo de la Pantalla de Búsqueda.

La pantalla de Búsqueda permite buscar dentro del contenido de la aplicación, presentando el resultado de la búsqueda como una lista de las pantallas a visitar y que contienen el resultado de la búsqueda.

7.11 Pantalla Formulario

A screenshot of a mobile application interface for a form screen. At the top, there is a dark header bar with a "Back" button in white. Below the header, the form consists of several labeled input fields: "Nombre del cliente:", "Nombre del proyecto:", "Descripcion del proyecto:", "Plataforma:", "Telefono:", and "Email:". The "Plataforma:" field is a dropdown menu currently showing "IOS". At the bottom of the screen, there is a dark bar with a "Home" button and three "+" buttons.

Figura 17: Ejemplo de la Pantalla de Formulario.

La ventana de formulario se utiliza para mostrar formularios con campos para que el usuario los pueda rellenar. Los campos que se mostrarán son previamente definidos por el xml y se crearán en tiempo de ejecución.

Incluye la funcionalidad de crear aplicaciones dinámicamente al pulsar el botón de aceptar. Puede utilizarse para

mostrar diferentes pantallas según los datos introducidos por el usuario.

7.12 Pantalla Imagen + Texto



Figura 18: Ejemplo de la Pantalla de Image con Texto.

La ventana Imagen con descripción es una ventana que muestra una imagen con un cuadro de texto en la parte superior. Incluye además un enlace debajo del texto con un Next Level y puede llevar a otra pantalla.

El cuadro de texto con la descripción es de formato único y admite scroll si el contenido es mayor que el espacio reservado. Si quiere utilizar texto con diferentes formatos puede considerar la pantalla de contenedor web.

7.13 Pantalla de Imagen con Zoom

Esta pantalla permite que el usuario pueda ver una imagen con gran Zoom, con una gran resolución. Debido a que la imagen es mayor que una imagen que se pueda mostrar en la galería, contamos con un contenedor 'especial' que permita mostrar la imagen en tamaño real y que además le permite al usuario 'navegar' a través de la imagen para poder ver todas sus partes a un tamaño real.

7.14 Pantalla de Imagen + Lista

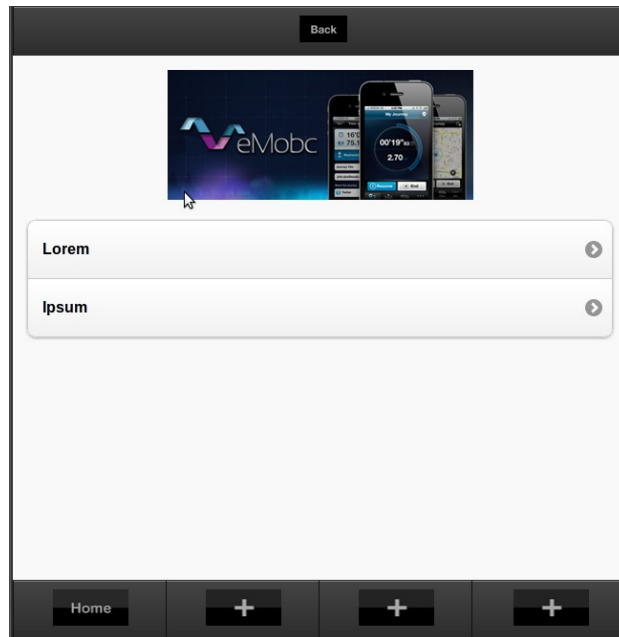


Figura 19: Ejemplo de la Pantalla de Lista con Image.

Esta pantalla muestra al usuario una imagen acompañada por una lista. Es parecida a la pantalla de lista. Las características de esta pantalla en concreto es que se va a mostrar una imagen en la parte inferior de la pantalla y que las celdas de la lista sólo van a poder contener texto y no pueden ir acompañadas por otra imagen.

7.15 Pantalla de Multimedia

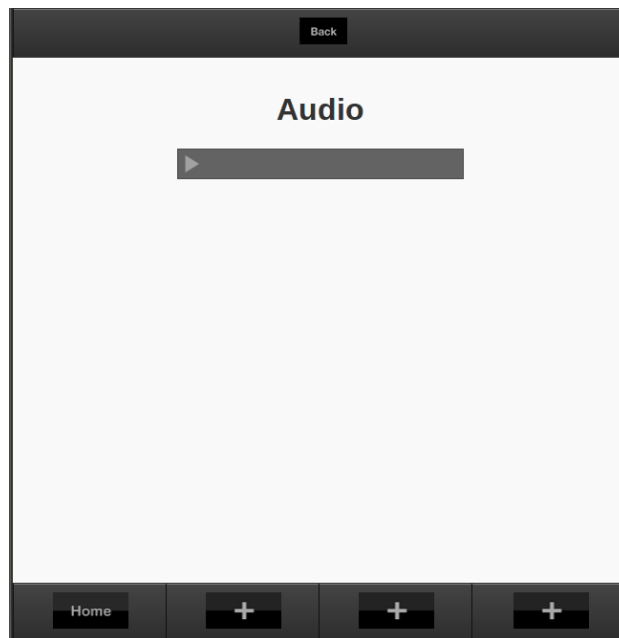


Figura 20: Ejemplo de la Pantalla de Multimedia.

La Pantalla de Multimedia, mostrada en la Figura 20, permite reproducir audio, tanto local como externo. La reproducción de un archivo externo se produce en straming. Es posible reproducir el audio mientras se descarga en segundo plano.

La ventana también muestra información y una imagen de la pista de audio. Si se va a reproducir una canción, es posible añadir las letras para que aparezcan en la ventana mientras se reproducen. Incluye un panel de controles básico para la reproducción de audio:

- Botón Play/Pause
- Botón Anterior
- Botón Siguiente

7.16 Pantalla de Calendario

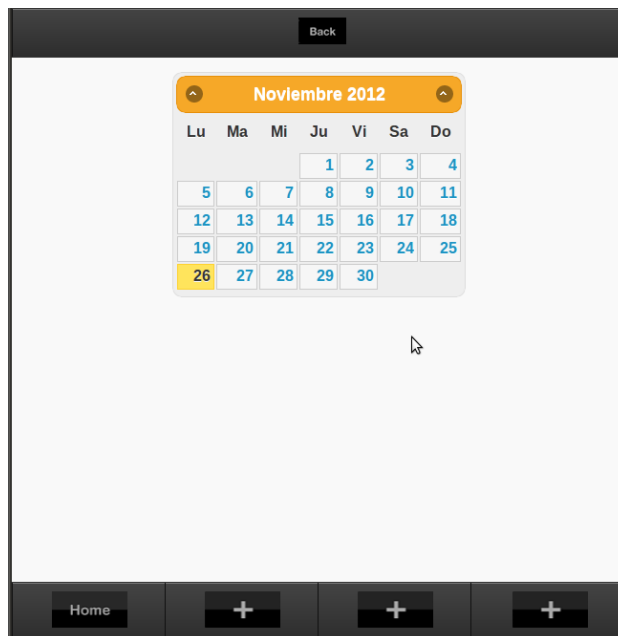


Figura 21: Ejemplo de la Pantalla de Calendario.

La ventana de calendario permite mostrar eventos en un calendario de aspecto mensual. Cada evento tiene asociada una descripción y puede llevar a una nueva ventana.

En el calendario se indican los eventos con colores en los días de la semana:

- **Azul:** Indica que existen eventos en el día
- **Círculo rojo:** Indica el día actual.

Cuando se pulse un día del calendario se mostrarán los eventos (si los hay) en la tabla de eventos. Es posible cambiar el mes para visualizar los posibles eventos en otros meses próximos y pasados.

7.17 Pantalla de Quiz



Figura 22: Ejemplo de la Pantalla de Quiz.

La ventana de quiz permite mostrar al usuario un cuestionario con una pregunta, una imagen (opcional) y unas respuestas. Una vez finalizado el quiz se mostrarán los resultados en una nueva ventana según los pesos de las preguntas y las respuestas que se hayan contestado.

La función Quiz dispone varias pantallas:

- Pantalla Inicial: Muestra una descripción del quiz. Es la pantalla de preparación para inicial el quiz.
- Pantalla de preguntas: Muestra las preguntas y respuestas hasta que no existan más.
- Pantalla de resultados: Muestra los resultados después de haber completado el quiz.

Además de los un cuestionario tradicional, la función Quiz puede ser utilizada para crear aventuras. Una aventura es un quiz en el cual cada respuesta tiene definida una nueva pregunta. De esta manera se pueden crear aventuras interactivas en las que el resultado final depende de las decisiones tomadas a lo largo del quiz.

8 Guía de programación

8.1 Cómo crear un nuevo tipo de ventana para el proyecto HTML5

El siguiente documento es una guía de programación para crear un nuevo tipo de ventana. Muestra en detalle los archivos y métodos que hay que modificar para conseguir que el framework trabaje con el nuevo tipo de ventana apoyándose en un ejemplo concreto (ventana de calendario).

Puede consultar también la “guía rápida nuevo tipo de ventana”.

8.1.1 Paso 1: Tipo de actividad

La nueva ventana tiene su propio tipo de Activity. Los tipos de activity se definen en la clase ActivityType.

```
abstract class ActivityType {  
    const COVER_ACTIVITY = 'COVER_ACTIVITY';  
    const BUTTONS_ACTIVITY = 'BUTTONS_ACTIVITY';  
    const IMAGE_TEXT_DESCRIPTION_ACTIVITY = 'IMAGE_TEXT_DESCRIPTION_ACTIVITY';  
    const IMAGE_LIST_ACTIVITY = 'IMAGE_LIST_ACTIVITY';  
    const LIST_ACTIVITY = 'LIST_ACTIVITY';  
    const VIDEO_ACTIVITY = 'VIDEO_ACTIVITY';  
    const IMAGE_ZOOM_ACTIVITY = 'IMAGE_ZOOM_ACTIVITY';  
    const IMAGE_GALLERY_ACTIVITY = 'IMAGE_GALLERY_ACTIVITY';  
    const WEB_ACTIVITY = 'WEB_ACTIVITY';  
    const QR_ACTIVITY = 'QR_ACTIVITY';  
    const FORM_ACTIVITY = 'FORM_ACTIVITY';  
    const MAP_ACTIVITY = 'MAP_ACTIVITY';  
    const PDF_ACTIVITY = 'PDF_ACTIVITY';  
    const SOUND_ACTIVITY = 'SOUND_ACTIVITY';  
    const CALENDAR_ACTIVITY = 'CALENDAR_ACTIVITY';  
    const QUIZ_ACTIVITY = 'QUIZ_ACTIVITY';  
    const AUDIO_ACTIVITY = 'AUDIO_ACTIVITY';  
    const CANVAS_ACTIVITY = 'CANVAS_ACTIVITY';  
    const CRUD_ACTIVITY = 'CRUD_ACTIVITY';  
    const PROFILE_ACTIVITY = 'PROFILE_ACTIVITY';  
    const SEARCH_ACTIVITY = 'SEARCH_ACTIVITY';  
}
```

Si queremos agregar un nuevo tipo de actividad, debemos en primer lugar agregarla aquí.

8.1.2 Paso 2: Crear y definir un xml

Este archivo XML va a contener los datos para poder crear la ventana. Sin embargo, no tiene porqué ser igual a otros archivos XML. Por ejemplo, una ventana de galería no necesita los mismos datos que una ventana de vídeo. Nuestra nueva ventana puede necesitar una estructura y datos concretos que la diferencian de las demás. Es necesario definir una nueva estructura para esta ventana.

8.1.3 Paso 3: Crear datos de la aplicación

Necesitaremos crear estructuras de datos para albergar los datos que hayamos definido en la etapa anterior.

8.1.4 Paso 4: Añadir parser

Necesitaremos crear un parser, siguiendo la convención de nombre definida anteriormente, (ej: \$EMOBC_HTML5_HOME/parser/nombre_actividad.php) para proporcionar el parser de la aplicación. Este archivo PHP deberá contar con la función:

function traer_info(\$xml,\$data)

La cual será la encargada de parsear el archivo y generar las estructuras de datos definidas con anterioridad