EUSKO JAURLARITZA GOBIERNO VASCO

OGASUN ETA FINANTZA SAILA
Informatika eta Telekomunikazio Zuzendaritza

DEPARTAMENTO DE HACIENDA Y FINANZAS
Dirección de Informática y Telecomunicaciones

Guía técnica para el desarrollo de soluciones móviles

Documento de Estándares

2012

Revisión	Fecha
1	Febrero, 2015
2	Noviembre, 2015

Donostia - San Sebastian, 1 – 01010 VITORIA-GASTEIZ
Tef. 945 01 85 22 – Fax 945 01 87 06

Índice

1 Introducción

En este documento se recogen las pautas y recomendaciones técnicas a tener en cuenta de cara al desarrollo de soluciones móviles en el entorno de la Red Corporativa Administrativa del Gobierno Vasco (RCAGV).

En los siguientes capítulos se describen bajo una perspectiva técnica los tipos de soluciones móviles y las consideraciones a tener en cuenta de cara a elegir un tipo u otro y cómo será la entrega de la aplicación en el sistema de control de versiones (Subversion [SVN]) de la RCAGV.

2 Consideraciones en el desarrollo móvil

En el desarrollo de soluciones de movilidad se deben tener en cuenta distintos aspectos según los requerimientos a cubrir en la aplicación:

Capacidades nativas del dispositivo:

Se refiere al hecho de utilizar las facilidades del propio dispositivo.

- **Hardware del dispositivo**: acelerómetro, giroscopio, cámara, GPS, comunicaciones (*wifi*, *nfc*, *bluetooth*), uso de aceleración 3D para gráficos, capacidad de almacenamiento en la memoria del dispositivo o en memoria externa extendida.

- **Integración con otras aplicaciones**: estado del dispositivo, agenda de contactos, calendario, fotografías asociadas a ubicación, compartir contenidos entre aplicaciones

- **Controles de interfaz**: el sistema operativo del dispositivo determina los controles a usar en la navegación y experiencia de usuario

Costes de desarrollo:

- **Portabilidad**: La portabilidad de una solución móvil a un mayor número de plataformas de dispositivos móviles implican mayores costes de desarrollo y mantenimiento. Para minimizar estos costes, las soluciones móviles se centran en las plataformas de mayor uso y en aquellas que corporativamente tengan que considerarse. El impacto de la portabilidad en el desarrollo se ve afectado por:

    - ✓ **Reutilización de código**: entendido como la posibilidad de desplegar una misma aplicación en múltiples plataformas móviles. Un bajo nivel de reutilización de código supondrá un gran número de modificaciones específicas para cada plataforma aumentando los costes de desarrollo.

    - ✓ **Reutilización de herramientas**: representa la posibilidad de reutilizar un mismo conjunto de herramientas de desarrollo para implementar la solución móvil para las distintas plataformas destino.

    - ✓ **Soporte de plataformas**: número de plataformas móviles a las que se publicará la solución móvil.

En el siguiente cuadro se puede ver cómo impacta el tipo de desarrollo móvil en las distintas características de un proyecto de desarrollo:

	Web	Hibrido	Nativo
Coste	razonable	razonable	caro
Tiempo	corto	corto	largo
Portabilidad	alta	alta	nula
Rendimiento	media	Rendimiento nativo si se necesita	muy rápida
Funciones nativas	No	Todas	Todas
Distribución aplicación store	No	Sí	Sí
Extensible	No	Sí	Sí

3 Tipos de aplicación

En base a los requisitos establecidos para el desarrollo de una aplicación móvil y la arquitectura móvil se podrían catalogar las aplicaciones en los siguientes tipos:

3.1 MWA (Mobile Web Application): Aplicación Móvil Web

Se trata de aplicaciones web desarrolladas para ser visualizadas y utilizadas en cualquier dispositivo que tenga un navegador independientemente de que sea de escritorio o móvil. La aplicación no se instala en el dispositivo sino que es accesible a través de un sitio que puede discriminar la presentación y el formato de la información en función de las dimensiones de la interfaz.

Por ejemplo, si la presentación es en un pc de escritorio podría mostrar una tabla con 10 columnas, pero si el dispositivo es una *tablet* podría mostrar 10 columnas en disposición apaisada o 5 en posición vertical y si el dispositivo fuese un móvil incluso podría presentar la información a nivel de registro.

En este tipo de aplicaciones el desarrollo se abstrae del sistema operativo y de las capacidades del dispositivo basándose en:

- HTML5, CSS3 y Javascript.
- Diseño sensible al dispositivo (*responsive design*).
- *Media queries*, para la generación de la interfaz en función de la resolución.
- *Single Page Applications,* donde la aplicación se muestra en una única página y los contenidos se añaden y eliminan dinámicamente proporcionando una experiencia de usuario más fluida.
- *Frameworks javascript* en capa de presentación (jQueryMobile, Sencha, AngularJS, Backbone, Ember).

La arquitectura de estas aplicaciones se asemeja a las aplicaciones web tradicionales y suele contar con:

- Un conjunto de servicios de negocio que se consumen mediante API's, *webservices*, servicios restFUL y que exponen la información y los datos a gestionar.

- Una capa de presentación donde el contenido es sensible a la resolución de la pantalla de visualización.

Normalmente se recurre a este tipo de desarrollo cuando se quiere adaptar una aplicación web a un dispositivo móvil en corto plazo de tiempo o con unos costes contenidos o cuando se quiere que un módulo de una aplicación pueda consumirse desde dispositivos móviles.

3.2 Native RMA (Native Resident Mobile Application): Aplicación Móvil Residente Nativa

Se trata de las aplicaciones que se instalan en el dispositivo y que se han desarrollado de forma específica para una determinada plataforma conforme al *Software Development Kit* (SDK) correspondiente.

Las plataformas móviles junto con sus correspondientes SDKs y lenguajes de programación más extendidas serían:

- Android: Android Development Tool, lenguaje Java
- Apple iOS: XCode, lenguaje Swift y Objective C
- Microsoft Windows Phone: Windows Phone SDK, lenguaje .Net
- Blackberry: Blackberry RIM, lenguaje Java

Este tipo de aplicaciones se caracterizan por:

- Tener un desarrollo específico para la plataforma destino.
- Disponer de un lenguaje de programación específico para cada plataforma.
- Tener un conjunto de herramientas de desarrollo recomendadas por el fabricante.
- Explotar las capacidades nativas del dispositivo (cámara, gps, acelerómetro,...).

Desde el punto de vista de los costes de desarrollo esta opción sería la más costosa ya que supone un desarrollo completo y recursos específicos por plataforma destino donde se publicará la aplicación.

3.3 Hybrid RMA (Hybrid Resident Mobile Application): Aplicación Móvil Residente Híbrida

Las aplicaciones móviles residentes híbridas son aquellas que se instalarán en el dispositivo móvil y cuyo desarrollo se implementa mediante lenguajes y patrones propios de las aplicaciones web así como el acceso a las capacidades específicas del dispositivo móvil.

Características:

- Desarrollo basado en un lenguaje web (C#, Java, Javascript, C++...) que se compila al lenguaje del sistema operativo correspondiente (Windows Phone, Android, iOS, Blackberry OS...)
- Uso de las capacidades nativas mediante la inclusión de *plugins* específicos de la plataforma.
- Herramientas (phonegap, cordova, xamarin, appcelerator, kony...)

4 Entorno de desarrollo

A continuación se va a describir de modo breve el entono de desarrollo en los diferentes tipos de aplicaciones.

4.1 MWA (Mobile Web Application)

Para este tipo de desarrollo se empleará el entorno habitual empleado para el desarrollo de aplicaciones web.

El desarrollo de este tipo de aplicaciones puede abordarse con **UDA** en cuanto a la parte servidora mientras que para la interfaz web deben considerarse pautas como:

- ***Single Page Application*** (SPA): aplicaciones en las que se interactúa en una única página donde se van mostrando los recursos necesarios en respuesta a la acción del usuario.

- ***Responsive Design***: diseño sensible al dispositivo de manera que tanto la interfaz como la interacción con la aplicación se adaptará al dispositivo desde el que se interactúa con la aplicación (monitor de escritorio con ratón y teclado, *tablet* con teclado, *tablet* o móvil con teclado virtual).

Además para este tipo de desarrollos es posible apoyarse en *frameworks* de capa de presentación como pueden ser jQuery Mobile, Angular JS, backbone, bootstrap, moustache, ionic.

4.2 Native RMA (Native Resident Mobile Application)

En el caso de desarrollar una aplicación nativa el desarrollo se deberá de realizar mediante el SDK correspondiente a cada una de las plataformas para la que se va a publicar la solución móvil.

4.3 Hybrid RMA (Hybrid Resident Mobile Application)

En el caso de desarrollar una aplicación híbrida mediante Cordova/Phonegap, es necesario diferenciar entre los dos flujos de trabajo posibles.

Cross-platform development

Se utilizan las herramientas de Cordova/Phonegap para la generación de los *builds* para las diferentes plataformas.

Para realizar los *builds* y gestionar las diferentes plataformas se deberán de instalar los SDKs correspondientes a las mismas.

El soporte para cada una de las plataformas es el siguiente:

- iOS (Mac)
- Amazon Fire OS (Mac, Linux, Windows)
- Android (Mac, Linux, Windows)
- BlackBerry 10 (Mac, Linux, Windows)
- Windows Phone 7 (Windows)
- Windows Phone 8 (Windows)
- Windows 8 (Windows)
- Firefox OS (Mac, Linux, Windows)

Platform-centered development

En este caso se utilizarán los SDKs correspondientes a las plataformas para las que se desean generar *builds* de la aplicación.

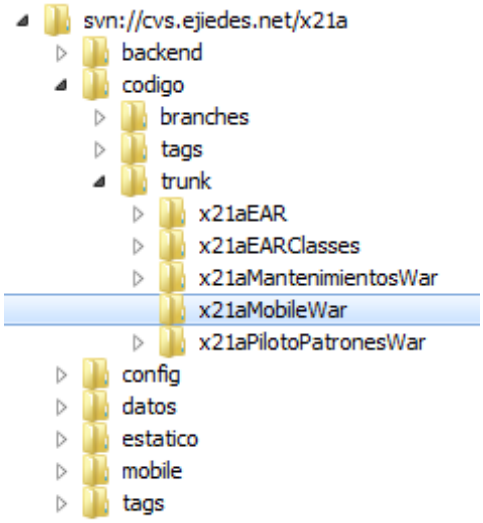
5 Versionado

En este apartado se va a detallar la estructura en la que se albergarán los proyectos en el SVN. La estructura de carpetas que se deberá respetar variará dependiendo del tipo de aplicación a implementar.

5.1 MWA (Mobile Web application)

Para este tipo de desarrollo se utilizará la estructura habitual definida para aplicaciones web.

La aplicación podrá tener distintos módulos web en función de la separación funcional que haya implementado y uno, varios o todos ellos pueden ser susceptibles de presentarse en dispositivo móvil a través del navegador.

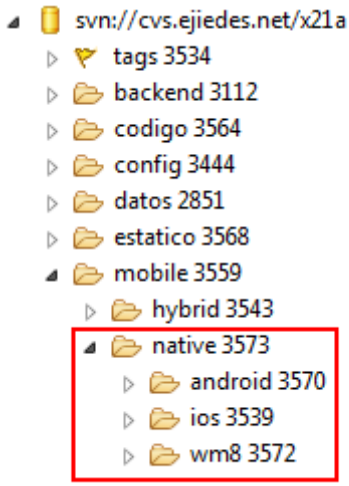


5.3 Native RMA (Native Resident Mobile Application)

Si se ha optado por desarrollar aplicaciones nativas para cada una de las plataformas para las que vamos a desarrollar la aplicación, se deberá de alojar en su carpeta correspondiente el código fuente asociado a cada plataforma.

- android: Fuentes correspondientes al desarrollo con el SDK de Android.
- ios: Fuentes correspondientes al desarrollo con el SDK de iOS.

- wp8: Fuentes correspondientes al desarrollo con el SDK de Windows Phone 8.
- En caso de desarrollar para otras plataformas existirá una versión por cada una.

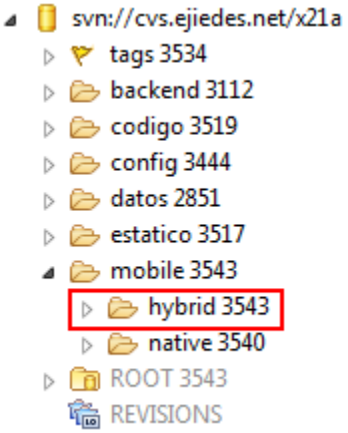


5.2 Hybrid RMA (Hybrid Resident Mobile Application)

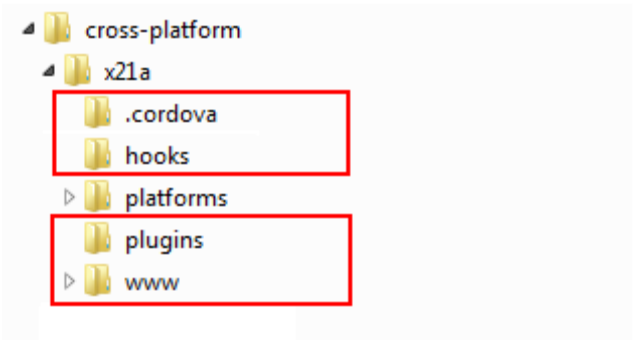
A la hora de desarrollar una aplicación híbrida con Cordova/Phonegap se diferencian dos flujos de trabajo a la hora de desarrollar la aplicación. La elección de uno u otro flujo determinará la estructura del SVN en la que se versionará la aplicación.

Cross-platform development

Las aplicaciones desarrolladas mediante *cross-platform workflow* se versionarán dentro del directorio *mobile/hybrid* del repositorio SVN.

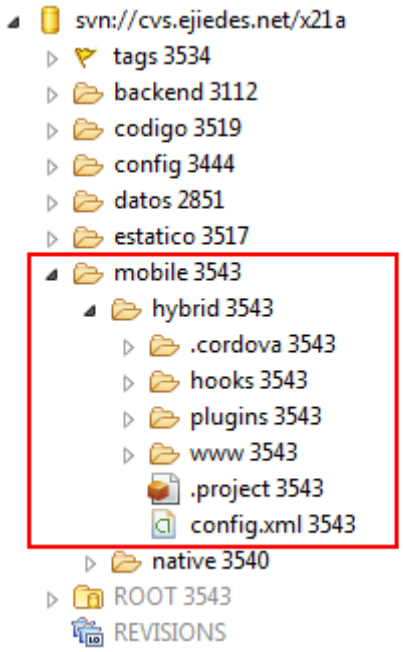


A la hora de subir el desarrollo al SVN, en el caso de trabajar con un flujo de trabajo *cross-platform*, se deberían de subir los siguientes directorios:



El directorio *platforms* no deberá de versionarse ya que se generan mediante las tareas *build* de cordova CLI.

El aspecto final del repositorio será el siguiente:

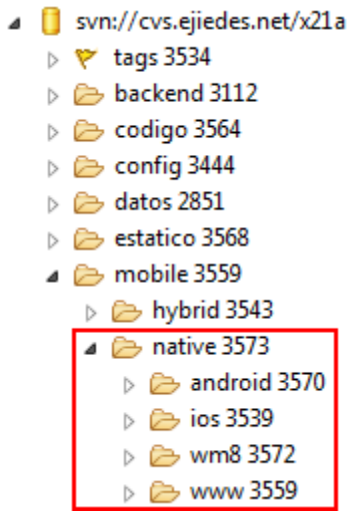


Platform-centered development

En el caso de que se opte por desarrollar mediante Phonegap/Cordova pero haciendo uso del SDK específico de cada plataforma, se deberán de subir los ficheros a los siguientes directorios del SVN:

- www: Directorio común donde se localizan los recursos web comunes a ser utilizados en la *web view* correspondiente de cada plataforma.
- android: Fuentes correspondientes al desarrollo con el SDK de Android.
- ios: Fuentes correspondientes al desarrollo con el SDK de iOS.
- wp8: Fuentes correspondientes al desarrollo con el SDK de Windows Phone 8.

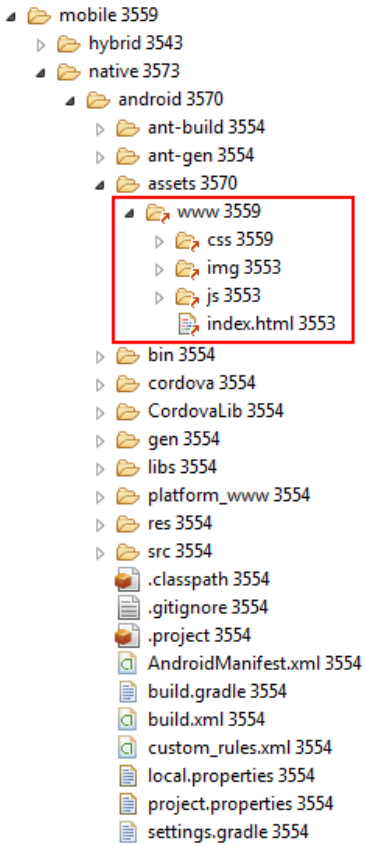
- En caso de desarrollar para otras plataformas existirá una versión por cada una.



Para desarrollar de manera común sobre el contenido del directorio www desde cada uno de los SDKs se hará uso de la funcionalidad *Externals* del SVN. De este modo se definirá una dependencia entre los proyectos específicos de cada plataforma con el directorio común www.

Android

Se definirá mediante *svn:externals* una dependencia del directorio común www dentro del directorio android/assets:



6 Procedimiento de implantación y publicación

El proceso de implantación y publicación para los desarrollos móviles en el entorno de la RCAGV tiene como actores implicados al grupo de desarrollo de la solución móvil, el servicio de «Gestión de Cambios» y el servicio de «Soporte».

El procedimiento a seguir para realizar la publicación de las aplicaciones móviles será similar al que existe actualmente para las aplicaciones convencionales.

En una primera fase del proceso, la aplicación/solución móvil deberá solicitar:

1. **Solicitud de código de aplicación y SVN**

   Se deberá de solicitar una RFC para obtener un código de aplicación indicando en la petición que se trata de una aplicación móvil y en que tienda (*store*) se va a publicar.

   Se deberá solicitar una intervención en Subversión:

   - Si es una iniciativa nueva con código de aplicación nuevo, para crear el repositorio SVN.
   - Si es una iniciativa nueva para un código de aplicación existente, para modificar el repositorio SVN con la estructura *mobile* descrita en el punto 5.

2. **Solicitud de publicación**

   Se deberá solicitar una tarea de despliegue de la aplicación en las diferentes tiendas (*store*) en las que vaya a ser publicada.

   Se deberá dejar el código de la aplicación móvil a publicar en la zona de traspaso:

   /traspaso/aplic/[código_aplicación]/

Publicación

Los datos necesarios para la publicación de una aplicación móvil son diferentes en función de la *Tienda* donde sea alojada. Será necesario suministrar estos datos al servicio de «Gestión de Cambios» para que pueda realizarse la correcta publicación en la *Tienda* correspondiente.

Publicar una aplicación en las diferentes tiendas de aplicaciones puede resultar intimidante para mucha gente, simplemente por la razón de que hay muchos motivos por los cuales una aplicación puede ser rechazada. Para evitar en la medida de lo posible esta situación, es recomendable leerse las guías de distribución de las distintas plataformas.

- Apple App Distribution Guide
- Android Google Play Guide y Android Launch Checklist
- Windows Phone Distribution

7 Buenas prácticas de desarrollo móvil

A continuación se describen una serie de pautas y consideraciones a tener en cuenta a la hora de realizar un desarrollo móvil.

Tamaño de la aplicación

El espacio que ocupa la aplicación es un factor a tener en cuenta:

- A la hora de descargarla desde la tienda (*store*) si se trata de una aplicación a instalar en el dispositivo.
- En la medida en la que requiera espacio para la persistencia de sus propios recursos

Guías de diseño

A la hora de desarrollar una aplicación que vaya a ser instalada en un dispositivo es imprescindible consultar las guías de diseño respectivas de cada plataforma.

- iOS Developer Library
- iOS Human Interface Guidelines
- Android Design
- Windows Design

Tipografías para lograr una apariencia nativa

En el caso de aplicaciones híbridas, utilizar la tipografía característica de la plataforma ayuda a conseguir una apariencia nativa debido a que a los usuarios les resultará más familiar. Afortunadamente es bastante sencillo incorporar estas fuentes dentro de una aplicación, a continuación figuran las guías y fuentes utilizadas.

- iOS Fonts
- Android Fonts
- Windows Phone Fonts

La interfaz de usuario reside en el dispositivo

Con el objetivo de acelerar el renderizado de la aplicación se ha de evitar generar la interfaz de usuario en la capa del servidor.

El objetivo es generar en el cliente la interfaz de usuario mediante HTML, JavaScript y CSS a partir de la información dinámica solicitada al servidor.

La comunicación entre el cliente y el servidor se realizará mediante servicios REST y de acuerdo a la respuesta se modificará la interfaz de usuario en el cliente mediante el uso de JavaScript.

Single page applications

A la hora de diseñar y desarrollar las aplicaciones móviles es imprescindible ofrecer al usuario una experiencia de uso fluida en el que la respuesta por parte de la aplicación a sus interacciones sea lo más inmediata posible.

Una estrategia para conseguir este objetivo es generar, en la medida de lo posible, una única página que contenga las diferentes pantallas de la aplicación. De este modo las transiciones entre ellas serán más rápidas y fluidas debido a que no se necesita solicitar la siguiente pantalla a mostrar.

Limitar el acceso a la red

Uno de los principales factores que ralentiza el uso de las aplicaciones móviles son los tiempos de espera en las peticiones a los servidores de aplicaciones.

Este acceso remoto es necesario en la mayoría de los casos para obtener la información que debe de ser mostrada al usuario. Sin embargo es posible identificar cierto contenido que sea susceptible de ser cacheado en mayor o menor medida.

- **Contenido estático**: El contenido que se identifique como estático y no varíe nunca deberá de ser almacenado en el dispositivo. De este modo evitaremos peticiones innecesarias al servidor.

- **Contenido dinámico**: En ciertos casos, es posible que no se necesite una actualización en tiempo real del contenido dinámico. Por ejemplo el contenido se carga al acceder por primera vez a la pantalla o se actualiza pasado un determinado tiempo. Para ello se puede recurrir a estrategias de almacenamiento local para persistir temporalmente la información.

    - *LocalStorage*.
    - Sistema de ficheros.
    - Base de datos SQL local.

No bloquear la interfaz de usuario

Se debería de evitar el bloquear la interfaz de usuario a la hora de realizar peticiones al servidor de aplicaciones. Del mismo modo no se deberá de demorar el momento de mostrar la interfaz a cuando se reciban los datos. Una alternativa sería el uso de elementos gráficos animados que indiquen que está en marcha un proceso de actualización de la información.

De este modo se logra una ejecución más fluida de la aplicación y se eliminan los bloqueos de la interfaz de usuario que pueden deslucir la experiencia de uso.

Uso de aceleración hardware nativa en las transiciones

Para dotar a las aplicaciones de una mayor fluidez y velocidad a la hora de realizar transiciones y navegaciones entre las diferentes pantallas se deberán de utilizar las nuevas funcionalidades que ofrecen las transiciones CSS que emplean las capacidades de aceleración hardware de los dispositivos.

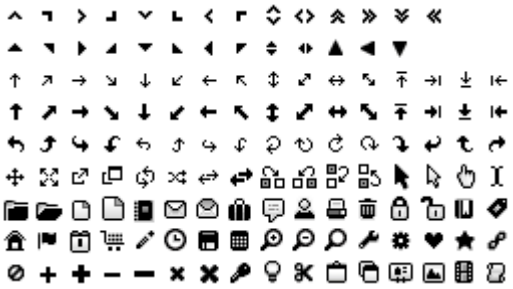
Hoy en día la mayoría de los dispositivos disponen de capacidades gráficas aceleradas por hardware con lo que se consigue un desempeño visual más eficiente y agradable para el usuario.

Un ejemplo es el del uso de transiciones 3d en los CSS para la navegación entre pantallas.

Optimización de las imágenes

El uso de las imágenes en los dispositivos móviles es susceptible de ser optimizado de varios modos:

- **Uso de *CSS Sprite Sheets***: En lugar de incluir las imágenes mediante una gran cantidad de ficheros individuales es recomendable generar *CSS Sprites*. De este modo solo se descarga un único fichero y se determina la imagen que se debe mostrar mediante el uso de CSS.



- **Emplear el tamaño de imagen oportuna**: Se debe de evitar el escalar las imágenes en el propio HTML. Es más conveniente el disponer de varios tamaños de imágenes para diferentes dispositivos que una imagen excesivamente pesada que sea escalada en cada uno de ellos.

Optimización del código JavaScript

A la hora de escribir el código JavaScript es muy importante el optimizar su uso e implementación. Es muy importante prestar atención a las recomendaciones para la optimización de código correspondientes al *framework* o librerías javascript que se empleen.

8 Datos necesarios para publicar una aplicación

Los datos necesarios para la publicación de una aplicación móvil son diferentes en función de la Tienda donde sea alojada. Será necesario suministrar estos datos al servicio de Gestión de Cambios para que pueda realizarse la correcta publicación en la Tienda correspondiente.

A continuación se detallan los datos necesarios para las tiendas de Apple, Google y Microsoft, que son los siguientes:

Tienda	Apple	Google	Microsoft
	App Store	Google play	
Nombre App	Si	Si (hasta 30 caracteres)	Si
Título	-	Si (hasta 30 caracteres)	-
Descripción	Se recomienda incluir párrafos de descripción para cada versión idiomática	Se recomienda incluir párrafos título/descripción para cada versión idiomática (hasta 4.000 caracteres)	Se recomienda incluir párrafos descripción para cada versión idiomática
Capturas de Pantalla	Al menos una para dispositivos de 3,5 pulgadas, 4 pulgadas e iPad.	Entre 2 y 8 capturas. Archivo JPEG o PNG de 24 bits. Longitud mínima 320 pixeles y longitud máxima 3.840 pixel por foto	Resolución 768 x 1280 px
Icono	Si	PNG de 32 bits, 512 x 512	99 x 99 px
Taxonomía	Categoría principal, Categoría secundaria y Palabras clave	Tipo de Aplicación, Categoría y Clasificación de contenido	Categoría principal, Categoría secundaria y Palabras clave
Sitio web	Por defecto, http://www.euskadi.eus		
Correo electrónico	-	Por defecto, ejgv.apps@gmail.com	-