

Global Atlas Mobile Application for IRENA Organization.

D. Gaston Iglesias⁽¹⁾, V. Sanjaime Calvet⁽²⁾, A. Del Rey Pérez⁽³⁾, I. Brodin Trujillano⁽⁴⁾

⁽¹⁾ Área de información geográfica en Prodevelop, S.L. Plaza D. Juan de Villarrasa 14, 46001, Valencia. dgaston@prodevelop.es

⁽²⁾ Área de información geográfica en Prodevelop, S.L. Plaza D. Juan de Villarrasa 14, 46001, Valencia. vsanjaime@prodevelop.es

⁽³⁾ Área de información geográfica en Prodevelop, S.L. Plaza D. Juan de Villarrasa 14, 46001, Valencia. adelrey@prodevelop.es

⁽⁴⁾ Área de información geográfica en Prodevelop, S.L. Plaza D. Juan de Villarrasa 14, 46001, Valencia. ibrodin@prodevelop.es

RESUMEN

El objetivo principal del desarrollo de esta aplicación es la búsqueda y visualización de información relativa a energías renovables en un dispositivo móvil. La información es ofrecida a través de la Agencia Internacional de Energías Renovables (IRENA [1]) localizada en Abu Dhabi.

La aplicación móvil ha sido desarrollada combinando las últimas tecnologías en software libre en visualización de datos sobre dispositivos móviles y entornos web, permitiendo de este modo un desarrollo multiplataforma para poder maximizar el acceso al mayor número de dispositivos disponibles en el mercado. En este documento se analizan las soluciones aplicadas para un desarrollo de este tipo.

Los datos son accesibles a través de los servicios de Global Atlas de IRENA que proporciona información geoespacial con estándares OGC [2]. La aplicación es capaz de acceder a estos protocolos para consultar y mostrar los datos usando un catálogo de servicios (CSW [3]) y obtener capas de mapas temáticos disponibles.

La aplicación proporciona funcionalidades útiles para la visualización de los datos como la generación de gráficas para datos temporales, información de datos puntuales o visualización en un globo 3D de las capas temáticas.

Palabras clave: *Jornadas, SIG, software libre, Girona, Energías renovables, Global Atlas, IRENA.*

ABSTRACT

The main objective of the development of this application is the search and visualization of information concerning renewable energy displayed on mobile devices. The information offered will be accessed through the Global Atlas of the International Renewable Energy Agency (IRENA [1]) located in Abu Dhabi.

The mobile application has been developed combining the latest open source technologies in data visualization on mobile devices and web environments, allowing a multi-platform development to maximize the access to devices available on the market. The solutions applied in that kind of development are analyzed in this document.

The data is available calling the services of Global Atlas of IRENA that provides geospatial information through OGC [2] standards. The mobile application is capable to access these protocols to query and show information through the catalog service (CSW [3]) to get the list of available layers.

The app provides useful functionalities to visualize data like chart generation for temporal data, information by point or visualization of the thematic maps on a 3D globe.

Key words: *Conferences, GIS, free software, Girona, Renewable energy, Global Atlas, IRENA.*

INTRODUCCIÓN

IRENA es la Agencia Internacional de Energías Renovables, que ha propulsado el lanzamiento del proyecto Global Atlas [4] para las energías renovables, que es una iniciativa para tener acceso a conjuntos de datos, conocimientos y soporte financiero necesarios para evaluar el potencial en energía renovable de los 67 países y más de 50 institutos y socios que contribuyen a esta iniciativa.

Este proyecto permite a los usuarios superponer la información listada en un catálogo de más de 1000 conjuntos de datos con cobertura mundial, donde la información está continuamente actualizada. La información está categorizada en:

- Mapas solares
- Viento
- Geotermia
- Bioenergía
- Marina

Los usuarios pueden, a través de sus interfaces GIS consultar y utilizar la información ofrecida por IRENA, pero además ésta ofrece un portal con un visor para mostrar la información, donde pueden modificar los datos y guardarlos en sus perfiles de usuario.

El proyecto de Global Atlas Mobile viene a cubrir la visualización de la información geoespacial disponible a través de dispositivos móviles. Para ello se ha pensado en una aplicación con las funciones básicas de acceso a cartografía disponible en los más importantes sistemas operativos móviles que existen actualmente en el mercado. La siguiente gráfica muestra la distribución actual del mercado de los diferentes sistemas operativos. Esta gráfica se ha tomado como punto de partida para la selección de las plataformas sobre las que desarrollar la aplicación Global Atlas Mobile.

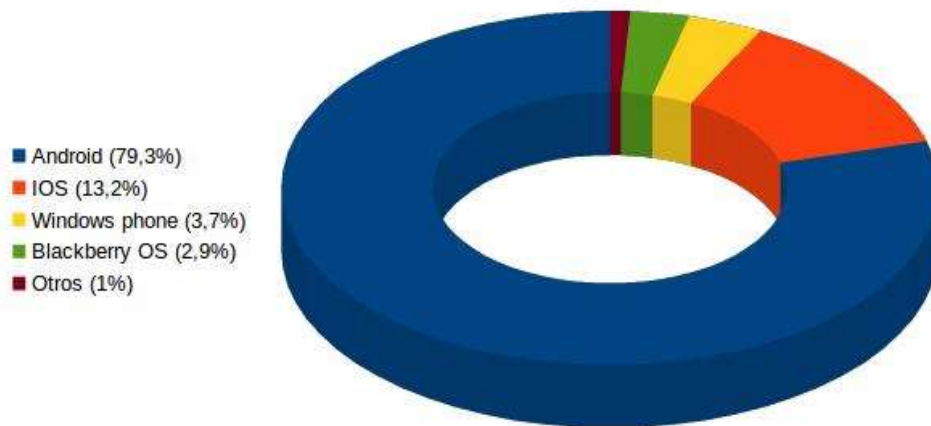


Figura 1: Mercado de telefonía móvil por sistema operativo

Considerando las restricciones de presupuesto y el requisito preestablecido de proyecto de llegar al mayor número de plataformas móviles posible, se decide cubrir como mínimo a los usuarios de Android, iOS, Windows Phone y Blackberry, que supone el 99% del mercado de sistemas operativos para Smartphone.

SERVICIOS DE DATOS

IRENA ofrece la información centralizada a través de un servicio de catálogo basado en Geonetwork [5]. Sobre la interfaz web de Geonetwork pueden hacerse búsquedas, pero en el caso de Global Atlas Mobile se utiliza el API de Geonetwork para acceso a los datos de las categorías. Inicialmente la previsión era el acceso a través del API estándar CSW, pero por restricciones en el modo de configuración de las capas que actualmente tiene el sistema ha sido necesario usar el API específica de la aplicación Geonetwork. En este caso no ha sido posible acceder a los *keywords* necesarios de las capas a través del API CSW.

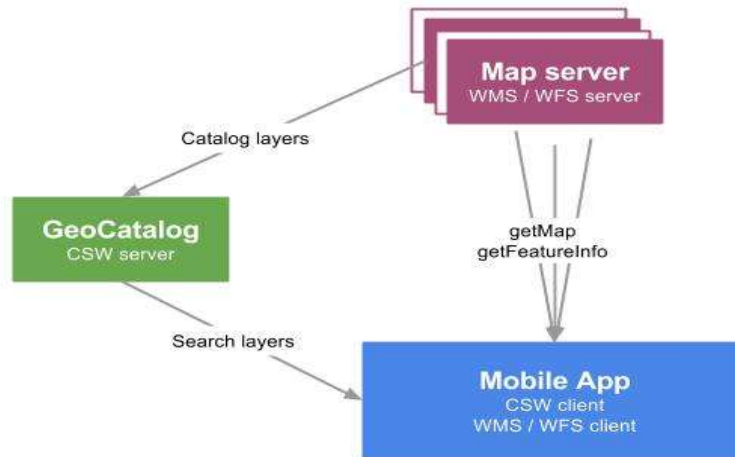


Figura 2: Acceso a datos externos de la aplicación móvil

La información dentro de Geonetwork se divide en diferentes categorías que son reflejadas en la aplicación web para el acceso a la información. Todas las capas del sistema no están disponibles para dispositivos móviles, por lo que se ha creado una nueva categoría para filtrar las capas disponibles en móviles.



Figura 3: Interfaz de Geonetwork en Global Atlas

Todos los servicios de datos referenciados son servidores Web Map Service [6], por lo que son fácilmente visualizables en una aplicación web de cualquier tipo, utilizando software libre.

La aplicación móvil consulta las capas disponibles a través de una única petición al servidor filtrando por su categoría. La petición contendrá información de unos metadatos básicos de cada capa en los que se incluye los siguientes:

- Título.
- Identificador para obtener la capa.
- *Abstract*
- Enlaces a *GetLegendGraphic* y una *preview* de la capacidad.
- *Keywords* y categorías asociados a la capa.
- Área geográfica.

Con la información devuelta por el servidor de catálogo la aplicación es capaz de organizar los metadatos y el acceso a los datos. Con esta consulta al servidor de catálogo la aplicación genera la estructura de categorías y puede mostrar la descripción de cada capa así como una *preview* de la misma.

SOLUCIONES EN LA APLICACIÓN CLIENTE

Arquitectura de cliente

Las restricciones de presupuesto ponen de manifiesto la imposibilidad de realizar cuatro desarrollos nativos para las plataformas elegidas, que encarecerían el coste del proyecto. Este aspecto orienta la aplicación desde el principio hacia el desarrollo web multiplataforma utilizando Cordova [7] para poder y empaquetar y subir a los diferentes mercados (*markets*).

La tecnología utilizada en los desarrollos es la siguiente:

- Cordova para compilación multiplataforma
- CocoonJS para iOS y Android
- HTML5 y javascript
- OpenLayers3[8] para visualización de cartografía
- Grunt [9] para automatizar las tareas de los componentes web

Los inconvenientes mayores de un desarrollo multiplataforma son un menor rendimiento del resultado final y la necesidad de desarrollar una aplicación web híbrida. La mayor ventaja que presenta es la de tener un único desarrollo lo que finalmente implica un menor número de horas de ejecución del proyecto.

El siguiente gráfico muestra la arquitectura del entorno de desarrollo de Cordova donde vemos como iOS y Android utilizan CocoonJS y este funciona sobre sus SDKs respectivas a través del Command Line Interface de Cordova (CLI). Todo ello, se soporta sobre Cordova donde se instalan los *plugins* para usar funcionalidades nativas y donde se instala también la aplicación web. En este caso Grunt es el encargado de desplegar la parte web sobre Cordova.

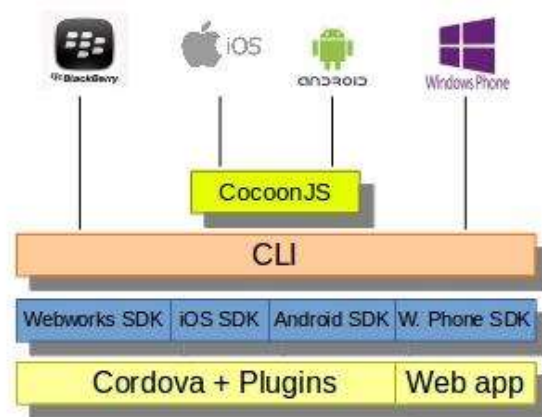


Figura 4: Arquitectura del entorno de desarrollo

Aunque se dispone por tanto de un proyecto común para todas las plataformas, cada una tiene necesidades de personalización y desarrollo específicos. A continuación se detallan los problemas más graves de personalización encontrados:

- El uso de componentes nativos (compartir en redes sociales, ubicación, etc.) necesita de desarrollos específicos dependientes de Cordova que no siempre funcionan bien o su funcionamiento no es igual para todas las plataformas. En el caso más grave, (compartir en redes sociales para Windows Phone) no ha podido resolverse.
- La redimensión y arrastre de los componentes web en la pantalla no se comporta igual para todas las plataformas, por lo que necesita de personalizaciones con código específico.
- El globo 3D tiene comportamientos muy diferentes y requiere una gran personalización por plataforma
- Para asegurar la fluidez en la navegación con WebGL en distintas versiones de Android e iOS ha sido necesario el uso de CocoonJS [10], una librería de *javascript* que permite mejorar aspectos del rendimiento pero que penaliza en el peso final de la aplicación.

Otro problema encontrado es que, de entrada, el desarrollo en cuatro sistemas operativos no garantiza que funcione en todos los dispositivos de ese sistema operativo ni en todas las versiones. El mayor problema es que una determinada versión puede no soportar una característica incluida en la aplicación o simplemente no sea capaz de mover los gráficos con suficiente fluidez por restricciones del propio sistema. Esto provoca que la aplicación tenga diferencias entre plataformas.

Solución sobre datos por punto y gráficas

Algunas capas del proyecto Global Atlas tienen asociada para cada punto información temporal, es decir hay almacenados en cada coordenada datos de distintos años, pudiéndose visualizar estos datos de forma tabular. Además también pueden ser representados en forma de gráfica aportando información de análisis extra. La manera en la que el servicio WMS tiene previsto el almacenamiento de información temporal es a través del parámetro TIME en las peticiones *GetMap* y es lo ideal para este tipo de escenarios. En este caso, el sistema usado tiene el inconveniente de que no toda la información devuelta en las peticiones *GetFeatureInfo* al servidor WMS es susceptible de ser representada mediante gráficas, por lo que pueden representarse datos que no tenga sentido hacerlo. Además hay que confiar en el *parseo* de la información textual que el servidor ha enviado para poder generar una gráfica que tenga sentido. Es decir, en general parece un sistema bastante débil pero que permite disponer de este análisis adaptando la aplicación a la información existente.



Figura 5: Información por punto y gráficas

El ámbito geográfico

Uno de los problemas encontrados es que para un gran número de capas se tiene la necesidad de restringir la cantidad de datos de los que se tiene disposición filtrando por área geográfica, debido a que el conjunto de datos completo tiene ámbito mundial. Para esto, se ha ideado un sistema a modo de *Wizard* previo a la consulta de capas en el que el usuario selecciona su área de trabajo. La aplicación solo enviará los datos disponibles que intersecten en su área de trabajo. Dicho *Wizard* se ejecutará la primera vez que se lanza la aplicación y posteriormente a petición del usuario. Se trata de la vista de un mapa donde puede salvarse un encuadre del territorio al nivel de zoom que se quiera.



Figura 6: Selección de área de interés

La visualización 3D

Uno de los requisitos de la aplicación es la visualización de la información en un globo 3D. Esta posibilidad tiene restricciones de plataforma debido a que no todos los sistemas son capaces de mover un globo 3D de forma fluida. Se han evaluado dos alternativas: WebGL Earth y Cesium. Inicialmente se apostó por la primera debido a que el globo tenía un funcionamiento más fluido. No obstante, después de algunas pruebas se detectaron problemas de memoria que hacía que nunca se liberara, provocando que la aplicación se cerrara de forma abrupta cuando se alternaba la vista 2D y 3D de forma repetida. Finalmente se optó por desarrollar con Cesium que no presentaba este problema pero se observó que el rendimiento era deficiente en los terminales más modestos. Para resolver el problema de rendimiento se utilizó CocoonJS.

No obstante, cuando se genera una aplicación con CocoonJS, ésta ocupa mucho espacio, por lo que la descarga se hace muy pesada y ocupa mucha memoria.

Aunque CocoonJS ayudó a paliar el problema del rendimiento no ha sido suficiente para todos los dispositivos, especialmente los más antiguos.

Capas base

Las capas base para una aplicación de visualización de capas temáticas es fundamental. En este caso se consideró razonable incluir la cartografía de OpenStreet Map y Bing. En contra de las previsiones iniciales no se ha podido usar la cartografía de Google debido a que no es soportado directamente por OpenLayers 3, ya que Google no permite acceder a los *tiles* directamente por problemas de licencia. Aunque pueden encontrarse formas alternativas de incorporarlo, finalmente OSM y Bing proporcionaron suficiente información.



Figura 7: Entorno de compilación y despliegue para múltiples dispositivos

CONCLUSIONES

El desarrollo de aplicaciones web híbridas multiplataforma se presenta como una opción candidata para cuando no se dispone de presupuesto suficiente para realizar un desarrollo nativo por plataforma.

Esta opción presenta ventajas enormes como son:

- Ahorro de costes importantes en el desarrollo
- No tener que disponer de especialistas en aplicaciones nativas de cada una de las plataformas. Es complicado tener personal que controle desarrollos en todos los sistemas y que trabajen juntos en el mismo proyecto, ya que incrementa mucho más los gastos.
- Tiempo de desarrollo más rápido.
- Gran similitud entre el resultado en las distintas plataformas.

No obstante, también hay que tener en cuenta que presenta multitud de inconvenientes como son:

- Restricciones de rendimiento frente a las aplicaciones nativas debido a que al final se está empaquetando una aplicación web. A los propios sistemas operativos no les interesa en ningún caso que los desarrolladores desarrollen soluciones multiplataforma en vez de nativas.
- El uso de CocoonJS para obtener un rendimiento digno a pesar de que penaliza en el peso final de la aplicación es lamentablemente es un escenario bastante común.
- Las personalizaciones por plataforma son inevitables.
- Sistema de construcción complejo.

REFERENCIAS

- ◆ [1] International Renewable Energy Agency website <http://www.irena.org/>
- ◆ [2] Open Spatial Consortium <http://www.opengeospatial.org/>
- ◆ [3] OGC Catalogue services (CSW) <http://www.opengeospatial.org/standards/cat>
- ◆ [4] Global Atlas for renewable energy <http://globalatlas.irena.org/>
- ◆ [5] Geonetwork open source project <http://geonetwork-opensource.org/>
- ◆ [6] OGC Web Map Service protocol (WMS) <http://www.opengeospatial.org/standards/wms>
- ◆ [7] Cordova project for native mobile applications <http://cordova.apache.org/>
- ◆ [8] OpenLayers 3 project <http://openlayers.org/>
- ◆ [9] Grunt project <http://gruntjs.com/>
- ◆ [10] Cocoonjs project <https://www.ludei.com/cocoonjs/>