

## **PolarISS WebOS**

Nick Vidal ([nick@iss.im](mailto:nick@iss.im))

Darwin Betancourt (traducción en español)

Hola. Mi nombre es Nick Vidal, soy de Brasil.

Yo soy Darwin Betancourt, de Ecuador, y ayude con esta traducción

Nosotros vamos a dar una visión general del PolarISS WebOS.

Este es una distribución construida en Drupal 7.

Lo interesante de esta distribución es el objetivo de dar un nuevo significado a la palabra “distribución”, perdon por la redundancia.

La mayoría de distribuciones Drupal están destinadas a funcionar de forma aislada de otros sitios Drupal.

Estas permiten a los usuarios en la misma red interactuar unos con otros.

Pero las mismas por si solas no interactúan entre sí.

Por lo tanto, si dos usuarios están en diferentes redes, no pueden interactuar.

El PolarISS WebOS, por otro lado, fue construido para trabajar en conjunto con otras instancias, de manera puramente descentralizada y distribuida.

El objetivo del PolarISS WebOS es ayudar a las personas a interactuar con los demás y compartir información, independiente de la red en la que se encuentran.

Por ejemplo, podríamos tener un usuario que está usando un WebOS de Brasil interactuando con otro usuario que está utilizando un WebOS de Perú.

No importa si el usuario esta en la misma red, porque tienen un protocolo común que les ayuda a comunicarse entre sí.

Esto es muy similar a cómo funciona el correo electrónico.

Pueden enviar un correo electrónico a cualquier persona, incluso si esta persona está en una red diferente.

Para lograr esto, PolarISS WebOS sigue una Arquitectura Orientada a Servicios.

Esto permite que una instancia de PolarISS WebOS pueda comunicarse con otra instancia.

De hecho, esta arquitectura orientada a servicios nos permite dividir el sistema en dos partes separadas que se comunican entre sí: la lógica y la interfaz.

En el servidor, tenemos un WebOS, proporcionando todos los datos y servicios y en el cliente tenemos un Webtop, proporcionando una gran interfaz de usuario.

Es interesante porque a partir de un único Webtop, el usuario puede acceder a múltiples WebOS.

De hecho, lo contrario también es cierto: un usuario puede acceder a un único WebOS de múltiples Webtop.

La interfaz de Drupal está muy ligada al código subyacente, pero gracias a esta arquitectura, la interfaz es totalmente separada de la lógica.

Lo que significa, que podemos tener una interfaz diferente a la que Drupal proporciona con jQuery.

En estos momentos estamos construyendo dos Webtops en Sencha, una librería poderosa de JavaScript.

El Alfa CrucISS Webtop se construye en ExtJS Sencha 4.

La Beta CrucISS Webtop se construye en Sencha Touch.

El Beta fue CrucISS es especialmente diseñado para ser accedido desde teléfonos móviles y las tabletas con pantalla táctil..como el Ipad

El PolarISS WebOS se construye en Drupal y se compone de varios módulos.

Como un ejemplo práctico de esta arquitectura en acción, imagina un usuario de una compañía que tiene un colega en la universidad. Supongamos que la universidad o uno de sus institutos decidió instalar un WebOS y Webtop para colaboración, y este colega es un usuario del sistema universitario. Ambos pueden intercambiar información e interactuar entre sí sin ningún problema, incluso aunque sean de diferentes redes.

Este hecho subyacente es totalmente transparente para ellos. La siguiente figura ilustra este ejemplo. La comunicación entre los dos puede ser a través de un solo WebOS, o uno en cada lado. Un servicio como el chat(mesajería instantánea), que requiere una latencia baja, sólo utiliza un WebOS. Esto mantiene la comunicación ágil y eficaz. El uso de dos WebOS es apropiado para servicios que mantienen registros de cada lado.

Por ejemplo, cuando un usuario envía una publicación, ésta es enviada a todos los suscriptores, incluso si están fuera de línea. El WebOS de los suscriptores es responsable de recibir y guardar estas publicaciones para su posterior recuperación.

Toda esta comunicación es coordinada por un estándar llamado IOS (Instant Operating System).

El IOS como lo había dicho es un estándar abierto que promueve la extensibilidad y la interoperabilidad que permite a las personas de diferentes redes poder compartir información entre sí.

El IOS ofrece el siguiente conjunto de servicios:

Servicios: Exponer servicios web. Proporcionar acceso seguro a través de

diferentes redes;

Usuarios: Agregar usuarios. Asignación de Roles. Dé permisos. Gestión de contactos personales;

Contenido: Crear contenido con campos asociados. Validación de formularios;

Canales: Crear canales y vincularlos.

Archivos: Administrar imagen, audio y video. Exportar archivos;

Idioma: Proporcionar localización. Traducir cadenas y contenido.

Buscar: Proporcionar los resultados de búsqueda tanto estructurados como no estructurados.

Sesiones: Guarda sesiones. Ayudar en las sesiones instantáneas entre los usuarios.

## DEMO

Ahora vamos a mostrar un demo llamado "Tea Sencha Network". El "Tea Sencha Network" es un análisis de redes sociales construido en la parte superior de Drupal y Sencha Touch. Su nombre fue inspirado en la película "The Social Network".

He creado un crawler Web que agrega un total de 125 post(publicaciones) a través de 20 categorías de un blog de Sencha. He identificado algunos patrones en estos datos semi-estructurados y lo que se ha migrado en Drupal 7. Quería crear algo que sea provechoso de "Fields in Core" y se me ocurrió la idea de dividir cada entrada en varias secciones. Esto sería más fácil para los usuarios de teléfonos móviles, ya que pueden ser capaces de buscar un índice para cada entrada y saltar directamente a una sección de interés.

También decidí dividir el grupo Sencha en dos equipos. Cada uno tendría su propio sitio de Drupal.

Ellos son:

El de color Verde:

(<http://green.teasencha.net>), integrado por 11 miembros, y

El de color Amarillo:

(<http://yellow.teasencha.net>), también integrada por 11 miembros.

El objetivo es demostrar cómo es posible acceder a información de los diferentes sitios desde una única interfaz. Esto es muy importante para la construcción de redes sociales descentralizadas.

Para lograr esto, he seguido una Arquitectura Orientada a servicios. Utilicé sitios Drupal para los servidores y Sencha Touch del lado del cliente.

Mientras que los sitios Drupal sirven los datos y los servicios, la interfaz Sencha Touch consumen los datos y servicios.

Yo decidí crear un gráfico utilizando Raphael JS mostrando cómo cada miembro de Sencha tiene que ver con la otra. Usé Categorías que los dos usuarios comparten en común como criterio. En el gráfico se resalta estas conexiones.

En el futuro, esto es algo que puede ser más explorado. Las personas pueden usar una force-directed-graph dirigido para navegar redes sociales, las categorías y la información. Como el grupo era relativamente pequeño, he creado un gráfico que gira alrededor de sí mismo. Un usuario se encuentra en el medio del gráfico y es posible cambiarlo con otro al arrastrarlo.

Cualquier acción que el usuario realiza en el gráfico, es inmediatamente sincronizada en el menú. De esta manera, el usuario puede saltar de una visualización a otra. La interfaz fue especialmente adaptada para dispositivos móviles. Por eso solo una visualización está disponible a la vez para ahorrar espacio en la pantalla. Sin embargo, sería interesante proporcionar una visualización simultánea de menú y el gráfico.

El menú proporciona acceso a un conjunto de servicios de diferentes sitios web. He utilizado JSONP ya que ambos sitios son confiables, pero sería una buena idea usar CORS para un sitio en producción. El módulo IOS que he desarrollado tiene soporte CORS.

Los servicios están relacionados entre sí y utilizan un mecanismo que se llama ServiceLink. Con ServiceLink, es posible que un usuario pueda navegar por una Lista de contactos de un amigo, e ir avanzando por las diferentes fases que muestra la aplicación hasta regresar nuevamente a su punto inicial como es la lista de contactos, todo esto sin darse cuenta que se estableció contacto con varios sitios web.