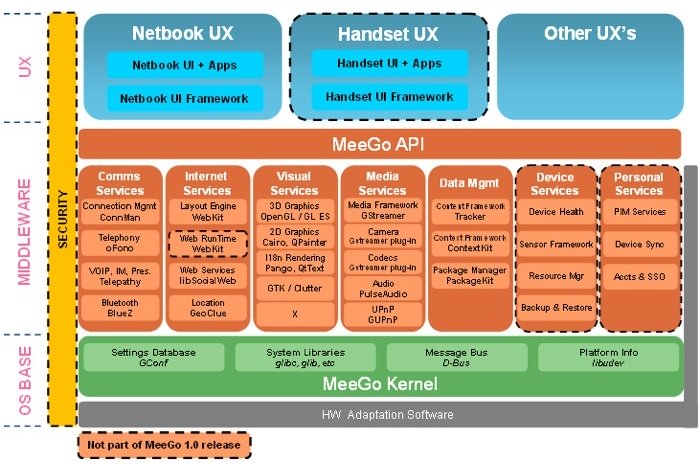
MEEGO

### Arquitectura

Meego es una distro Linux union de Maemo de Nokia y Moblin de Intel, cuya arquitectura a grandes rasgos se divide en 3 capas, [OS Base](http://realnorth.net/blog/meego-architecture-overview/" \l "os-base), [Middleware](http://realnorth.net/blog/meego-architecture-overview/" \l "middleware) y [UX](http://realnorth.net/blog/meego-architecture-overview/" \l "ux).



#### OS Base

Esta capa esta constituida por la HAL (La capa de abstracción de hardware ) y un kernel Linux, a [diferencia de Android](http://www.zdnet.com/blog/open-source/dibona-google-will-hire-two-android-coders-to-work-with-kernelorg/6274), Meego sigue una política de upstream first que lo mantiene sincronizado con las ultimas versiones de el kernel Linux, esto permite adoptar fácilmente nuevas versiones del kernel dentro del proyecto, sino que también obliga a los fabricantes de hardware a que se encarguen de todo el proceso de inclusión de sus drivers en el kernel.

#### Middleware

En esta capa corren a grandes rasgos:

* Los servicios y APIs que permiten el acceso a todo tipo de conectividad, 3G, WiFi, Bluethot (Comms Services)
* El web runtime, los servicios de rendering y geolocalización (Internet Services)
* Los subsistemas 2D y 3D, los servicios para el rendering de fonts, la libreria Clutter y el server X (Visual Services)
* Todos los servicios y codecs necesarios para la reproducción y captura de audio y video (Media Services)
* El subsistema que se encarga de proveer los servicios para la manipulación de metadata, por ejemplo información de media files, la posición del dispositivo (acelerometro) (Data Management).
* Todos los subsistemas y servicios encargados de administrar el estado del dispositivo, como así también las sincronización, backup y restauración del sistema. (Device Services)
* Todos los servicios necesario para el manejo y administración de los datos personales del usuario, (contactos, tareas, calendario) y el manejo de las cuentas de acceso a servicios web. (Personal Services)

##### Meego API

La Meego API esta basada en QT 4.6.x (al día de hoy, pronto se estara dando soporte a la versión 4.7), además se están desarrollando complementos como MeeGo Touch Framework, MeeGo Web Runtime (permite crear aplicaciones nativas usando HTML, CSS y Javascript) y Qt Mobility (un set de librerías con funciones y componentes especialmente desarrollando para se usados en ambientes mobile)

#### UX

Meego fue pensado desde un principio para ser usado en una gran variedad de dispositivos (netbooks, smartphones, tablets, televisores y automóviles), en esta capa corre una User Interface especializada para cada uno de ellos.

<http://realnorth.net/blog/meego-architecture-overview/>

Lenguajes soportados

La librería utilizada en Meego es Qt.([framework](https://es.wikipedia.org/wiki/Framework) [multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma) [orientado a objetos](https://es.wikipedia.org/wiki/Programación_orientada_a_objetos) ampliamente usado para desarrollar [programas](https://es.wikipedia.org/wiki/Programa_informático) ([software](https://es.wikipedia.org/wiki/Software)) que utilicen [interfaz gráfica de usuario](https://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario), así como también diferentes tipos de herramientas para la [línea de comandos](https://es.wikipedia.org/wiki/Interfaz_de_línea_de_comandos) y consolas para [servidores](https://es.wikipedia.org/wiki/Servidor) que no necesitan una interfaz gráfica de usuario. ) Aplicaciones Qt pueden ser escritas en muchos lenguajes, pero el principal es C++. Python tambien esta disponible, C#,PHP,Ada,Java.