|  |
| --- |
| http://www.becas.sep.gob.mx/images/logo.png  Presentación SOs Móviles  MeeGo  **27/08/2016**  Unidad 1  Kevin Antonio Villar Piñón 13550430  Ulises Manuel De la Rosa Contreras 13550362  Sistemas Operativos Para Dispositivos Móviles |
|  |
|  |
|  |

### Nuestro trabajo debe tener tabla de contenido, anexos, todo esto en formato APA.

Desarrollo de la explicación de la plataforma a exponer, cada una de las partes que se desarrollaron dentro del tema, características, lenguajes, tipos de plataforma, manejo de datos, y cada una de las plataformas, así como cada tema descrito en la exposición, los siguientes temas son lo que hay que agregar a la documentación haciendo mención a la plataforma y cada una de sus características.

Cada tema mencionado hay que desarrollarlo, de la siguiente forma:

# Introducción.-

MeeGo surge por la alianza entre las empresas Nokia e Intel, como una fusión de dos sistemas operativos: el Maebo de Nokia y el Moblin de Intel. Moblin es una abreviación para Mobile Linux. Con esto obviamente el sistema está basado ampliamente en Linux, más específicamente en Fedora.

El sistema resultaría muy ambicioso para su tiempo ya que planeaba competirle al IOs de Apple y al reciente Android de Google, aparte de que apostaba por una tecnología multiplataforma de las cuales se destacan: smartphones, tablets, smart TVs, netbooks e inclusive funcionar como sistema para ciertos automóviles.

**Evolución**

En una presentación conjunta Nokia e Intel mostraron su sistema operativo al mundo en el **Mobile World Congress de Barcelona en 2010**, la misma edición que vio el nacimiento de Bada. Desde el inicio varias empresas apoyaron el proyecto tales como: Acer, Asus, Gameloft, Peugeot e incluso AMD, gran rival de Intel en el mercado de los microprocesadores.

La apuesta de Intel era fuerte y aseguró que el sistema operativo recibiría una actualización potente cada seis meses, y que cada una llevaría el nombre de una ciudad estadounidense. **A finales de mayo de 2010 se lanzó la primera versión de MeeGo** para diversos netbooks que ya estaban en el mercado y que corrían sobre Windows XP o Linux. La versión para móviles todavía tardaría unos meses en llegar. Fue en octubre, cuando se lanzó MeeGo 1.1 que daba soporte a algunos teléfonos que funcionaban tanto con Maemo como con Moblin, dando acceso a ambas tiendas de aplicaciones, tanto OVI como Moblin Store, según el dispositivo.

Todo parecía ir bien, muy bien. Sólo faltaba el lanzamiento de un dispositivo con MeeGo de fábrica para probar la viabilidad de la plataforma. Nokia era la que tomó la responsabilidad con un smartphone: el **Nokia N9** que debía salir en 2010, pero comenzó a sufrir retrasos. Y aquí fue donde todo empezó a salir mal para MeeGo, concretamente el 21 de septiembre de 2010.

Olli-Pekka Kallasvuo fue sustituido por Stephen Elop como CEO de Nokia. Elop venía de Microsoft y **en febrero de 2011 se anunció que la compañía finlandesa apostaría por Windows Phone** en la que finalmente fue la gama Lumia. El efecto dominó había comenzado.

El Nokia N9 fue finalmente presentado en junio de 2011 -cuando ya había otros terminales en el mercado que equipaban MeeGo de serie-, presentación en la que no sólo se apuntó que sería el primer terminal de Nokia con el sistema operativo: también sería el último. En septiembre, mes en que se lanzó el terminal,**Intel canceló el desarrollo de MeeGo** para colaborar con Samsung en el desarrollo de Tizen -que a su vez decapitaba a Bada-.

Linux se quedaba sola con MeeGo. Gracias a la comunidad, el sistema operativo **tuvo actualizaciones hasta julio de 2012**, cuando finalmente murió tras una vida corta, pero que lo sitúa como “abuelo” de varios sistemas operativos actuales, como Tizen.

Sin embargo, que esté muerto no quiere decir que no se pueda reencarnar. En 2012 un grupo de ex-empleados de Nokia creó **Jolla** y lanzaron su propio sistema operativo, [Sailfish OS](http://www.elandroidelibre.com/2012/07/jolla-la-continuacion-de-meego-compatible-con-android.html), creado a partir de las cenizas de MeeGo y con el que comparte muchos elementos comunes, como la navegación a partir de gestos sin teclas .

La ultima acualización de Selfish OS hasta la fecha es la 2.0.1.11 lanzada en Mayo del 2016.

## Características

### Generales

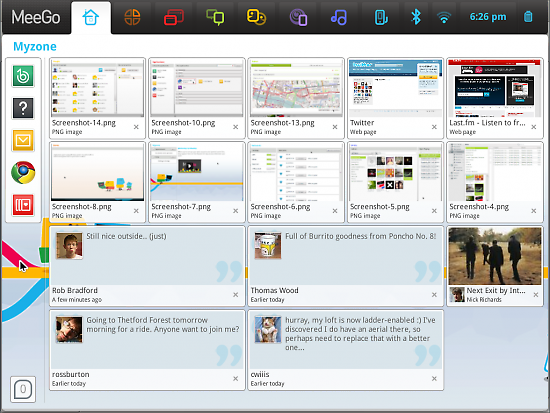
* [VoIP](https://es.wikipedia.org/wiki/VoIP).
* [Mensajería instantánea](https://es.wikipedia.org/wiki/Mensajer%C3%ADa_instant%C3%A1nea).
* [Correo electrónico](https://es.wikipedia.org/wiki/Correo_electr%C3%B3nico) y [calendario](https://es.wikipedia.org/wiki/Calendario).
* Integración con redes sociales como [Facebook](https://es.wikipedia.org/wiki/Facebook) y [twitter](https://es.wikipedia.org/wiki/Twitter).
* Servicios de localización.
* Sincronización de datos en la nube o [cloud computing](https://es.wikipedia.org/wiki/Cloud_computing).
* Aplicaciones de medios.
* Cámara.
* Soporte de gestos.

**Interfaz**

**MeeGo 1.0**

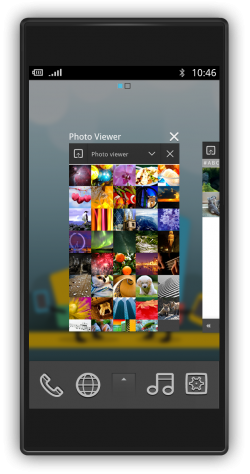


Meego 1.0 en smartphone



Meego 1.0 en Netbook

**MeeGo 1.1**



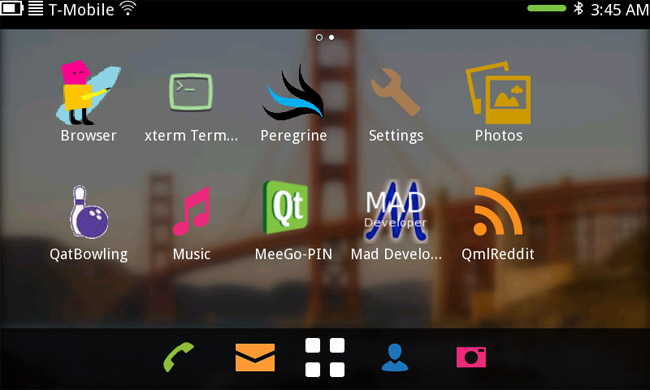
MeeGo 1.1 en Smartphone



MeeGo 1.1 en Automóvil.

Esta versión mejoraba la funcionalidad general del dispositivo como la conectividad, un sensor para la auto rotación , exploradores de web y la reproducción de música y video.

**MeeGo 1.2**



MeeGo 1.2 en smartphone

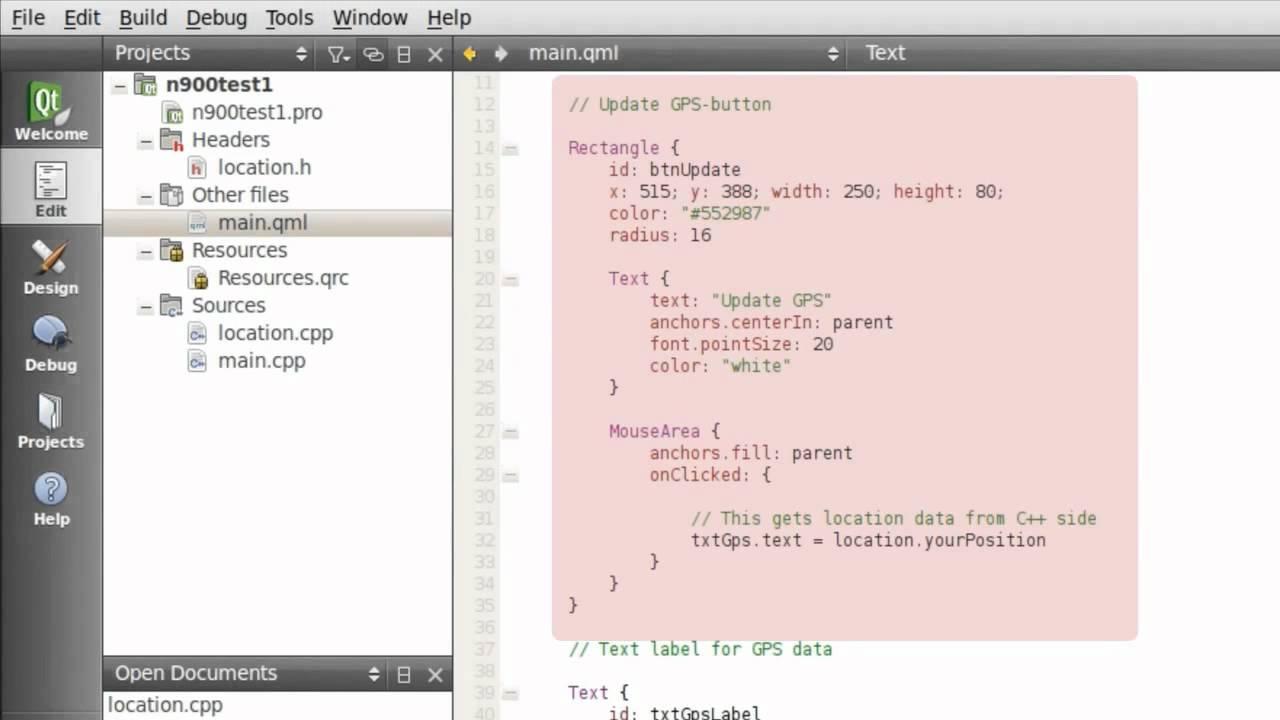
En esta versión hay varios núcleos de referencia para plataformas variadas Intel Atom y ARMv7, telefonía mejorada así como capacidades de conectividad, marco de aplicaciones QML y Qt-Mobility ampliado (que incluye la localización adicional, el sistema, la conectividad y el sensor / capacidades táctiles.

Un completo conjunto de aplicaciones básicas para netbooks, de información y entretenimiento también son añadidas mediante Netbook UX 1.2

# Desarrollo.-

MeeGo utiliza principalmente el lenguaje C++ y Python. También soporta otros lenguajes como PHP, JavaScript y de forma no nativa Java. El framework utilizado por MeeGo fue Qt, el cual soportaba los lenguajes anteriormente mencionados.

No se tiene mucha información sobre los requerimientos finales de MeeGo debido a que salió muy rápido del mercado y no se terminó de desarrollar de forma correcta.



**Ejemplo de entorno C++ en MeeGo**

**Ejemplo de entorno en Python**

Aquí se describirán los lenguajes de programación que se utilizan para programar en la plataforma, los principales entornos de desarrollo con sus características, y los requerimientos de los dispositivos para correr aplicaciones ya sea android, ios o windows phone, también una pequeña guía de cómo configurar el IDE y el emulador para correr las aplicaciones ya sean en el dispositivo o en el emulador, agregar screenshots de código de cada uno de los lenguajes con los que se puede desarrollar aplicaciones para cada una de las plataformas. Complementar con imágenes.

Desarrollar cada una de las partes de la estructura del sistema operativo:

1.4.1. Núcleo.

El núcleo de MeeGo cuenta con un kernel Linux que tiene una política de upstream first que lo mantiene sincronizado con las últimas versiones de Linux, esto permite adoptar fácilmente nuevas versiones del kernel dentro del proyecto, también obliga a los fabricantes de hardware a que se encarguen de todo el proceso de inclusión de sus drivers en el kernel.

1.4.2. Procesos.

En el middleware es la capa en la que corren las aplicaciones y los procesos, en esta capa corren a grandes rasgos:

• Los servicios y APIs que permiten el acceso a todo tipo de conectividad, 3G, WiFi, Bluetooth (Comms Services)

• El web runtime, los servicios de rendering y geo localización (Internet Services)

• Los subsistemas 2D y 3D, los servicios para el rendering de fonts, la librería 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 metadatos, por ejemplo información de media files, la posición del dispositivo (acelerómetro) (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 necesarios 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)

1.4.3. Virtualización.

La herramienta de particionado es sencilla de usar, pero cuando se usa en español hay un problema: hay que crear al menos dos particiones, la raíz (/), que utilizará el sistema de ficheros Btrfs por defecto y la partición de arranque (/boot), de pequeño tamaño (200 MB bastan)

1.4.4. Almacenamiento y sistemas de archivos.

1.4.5. Lenguajes de programación soportados.

La librería utilizada en MeeGo es Qt.(framework multiplataforma orientado a objetos ampliamente usado para desarrollar programas que utilicen interfaz gráfica de usuario). Las aplicaciones Qt pueden ser escritas en muchos lenguajes pero el principal es C++ y en menor medida Python, también están disponibles C#, PHP, Ada, Java.

# Conclusión.-

MeeGo comenzó como una idea prometedora e interesante que bien pudo competir con los grandes sistemas del mercado. En papel todo parecía muy bueno pero poco a poco el proyecto se fue diluyendo, más que nada por la alianza de Nokia y Microsoft lo que ocasionó que su principal distribuidora lo abandonara y quedando en el prácticamente muerto. Aun así, el sistema evolucionó gracias a Jolla con el Selfish OS lo que le valió algunos años más de vida.

Los usuarios de pudieron tener un contacto con MeeGo disfrutaron mucho su interfaz gráfica, que era bonita a la vista, además de su fluidez.