

Refinando la experiencia de mi SO para tablets

Usando Surface Go 2 con 4G/LTE, Debian GNU/Linux e i3wm

Willem L. Middelkoop
Sep. 28, 2020



A principios de este año intenté crear mi propio sistema operativo para tableta instalando Debian GNU/Linux en una tableta Microsoft Surface Go. Aprendí mucho sobre lo que me gusta de las tabletas. Pero seguía prefiriendo mi iPad Pro, principalmente por su experiencia de usuario pulida. Esta vez me propuse refinar mi software de tableta para reemplazar mi iPad.

Linux en tabletas (Surface)

Si estás pensando en instalar Linux en una tableta: no estás solo. Mi publicación anterior sobre [instalar Debian GNU/Linux en Surface GO](#) recibió mucha atención. Aparente-

mente, muchos de ustedes están buscando una alternativa para los sistemas operativos cerrados que generalmente se instalan en las tabletas (como iPadOS, Android o Windows 10S).

Hardware de Microsoft Surface y Linux

Como la historia está llena de dulce ironía, es Microsoft quien realmente fabrica un hardware bastante atractivo que funciona bien con Linux. Su línea Surface incluye diferentes computadoras portátiles y tabletas. La página de reddit "[SurfaceLinux](#)" está llena de interesantes historias de usuarios que vale la pena consultar. La gente lo está haciendo y tú también podrías hacerlo.



El hardware de Microsoft Surface es popular porque es realmente bueno

Primer intento: Surface Go (versus iPad Pro)

Después de [instalar y modificar Debian GNU/Linux en Surface Go](#), apodado "Willem OS", intenté convertirlo en mi computadora principal. Si eres un visitante frecuente de mi blog, sabrás que [he estado usando iPads y tabletas durante años como computadora principal](#). Finalmente, consideré que mi primera versión de "WillemOS" no era lo suficientemente buena: el iPad Pro estaba a años luz de distancia en lo que respecta a una experiencia de usuario pulida.

De este primer intento [aprendí](#) mucho, lo que me llevó a una mejor comprensión de lo que busco en una experiencia de tableta. **Busco un software simple pero versátil para ejecutar en mi tableta.**

Pero "simple" es en realidad muy difícil de lograr. Para hacer las cosas *realmente* simples, necesitas integrar perfectamente el hardware y el software. Apple lo hace bien, el iPad Pro proporciona una gran experiencia de usuario porque:

- **continuar donde lo dejaste:** guarda todo cuando lo guardas en una [bolsa](#), estando (instantáneamente) listo para continuar cuando lo sacas. No se requiere guardar o (re)iniciar manualmente.
- **internet en todas partes:** cambia automáticamente entre redes [WiFi y 4G/LTE](#), no necesitas hacer nada, esto 'simplemente funciona'
- **cualquier orientación y posición:** el iPad funciona muy bien sin un teclado [externo](#) o mouse, lo que te permite usarlo en muchos escenarios diferentes (la pantalla también ajusta automáticamente su brillo y balance de blancos para que coincida con tu entorno).

Resulta que lograr que estas cosas funcionen a la perfección con Debian GNU/Linux en Surface es difícil. Mi primer sistema operativo para tableta requería mucha interacción manual y, por lo tanto, era engorroso en comparación con el iPad Pro.

Segundo intento: Surface Go 2

Microsoft lanzó una nueva tableta Surface Go este año, la Surface Go 2. Tiene un procesador más rápido (¡hasta un 64%!), biseles más delgados y una pantalla mejorada. Es una actualización muy decente que pone a la Surface Go 2 mucho más a la par con el iPad Pro de 11 pulgadas en términos de hardware. Para mi segundo intento de crear la mejor experiencia de tableta, decidí usar el modelo superior de Surface Go 2:

- Pantalla táctil de 10.5" (1920x1250, 220PPI), relación de aspecto 3:2, Corning® Gorilla® Glass 3
- Procesador Intel Core m3 (8.^a generación)
- 8GB RAM
- Wifi 802.11a/b/g/n/ac/ax
- LTE Advanced (módem Qualcomm® Snapdragon™ X16 LTE)
- Bluetooth 5.0
- Gráficos Intel UHD 615
- Dimensiones: 245 mm x 175 mm x 8.3 mm
- SSD de 256 GB
- Tarjeta MicroSDXC de 512 GB
- 1 puerto USB-C y 1 puerto Surface Connect



Microsoft Surface Go 2 con teclado Type Cover negro y Surface Pen: ¡una pieza de hardware excelente!

Instalación de Debian GNU/Linux en Surface Go 2

La Surface Go 2 es muy similar a su predecesora cuando se trata de instalar una instalación básica de tu distribución GNU/Linux favorita. Seguí mis propios pasos de la instalación original de Surface Go [aquí](#).



Instalar Debian GNU/Linux en Surface Go 2 es muy similar al Surface Go original

En resumen, debes seguir los siguientes pasos al instalar Linux en tu Surface:

- 1) Actualiza Windows 10 e instala todos los parches/actualizaciones de firmware
- 2) Crea un medio de instalación de arranque que contenga tu distribución GNU/Linux favorita
- 3) (Opcional) conecta una conexión ethernet por cable
- 4) Arranca la Surface desde la unidad USB
- 5) Instala el sistema operativo usando el instalador
- 6) Ajusta y optimiza la experiencia del software



Debian GNU/Linux con i3wm (“también conocido como WillemOS”) ejecutándose en Surface Go 2

Refinando la experiencia

El último paso de la instalación es donde puedes marcar la diferencia entre una experiencia de usuario fantástica y pésima. Para *“alentarme”* a seguir con este experimento, decidí vender mi iPad Pro: ¡quemando las naves (ya que no habría vuelta atrás)!



Vender tu iPad Pro es fácil si conservas las cajas originales; yo vendí el mío en solo unos días: ¡obligándome a mí mismo a hacer que la nueva tableta funcione!

En una pantalla pequeña, ejecutar aplicaciones en pantalla completa tiene mucho sentido. Sé que el concepto convencional de PC de administrar ventanas de aplicaciones distintas no proporciona la mejor experiencia de usuario en una tableta pequeña. Por lo tanto, decidí probar un gestor de ventanas en mosaico: [i3wm](#).

Gestor de ventanas en mosaico

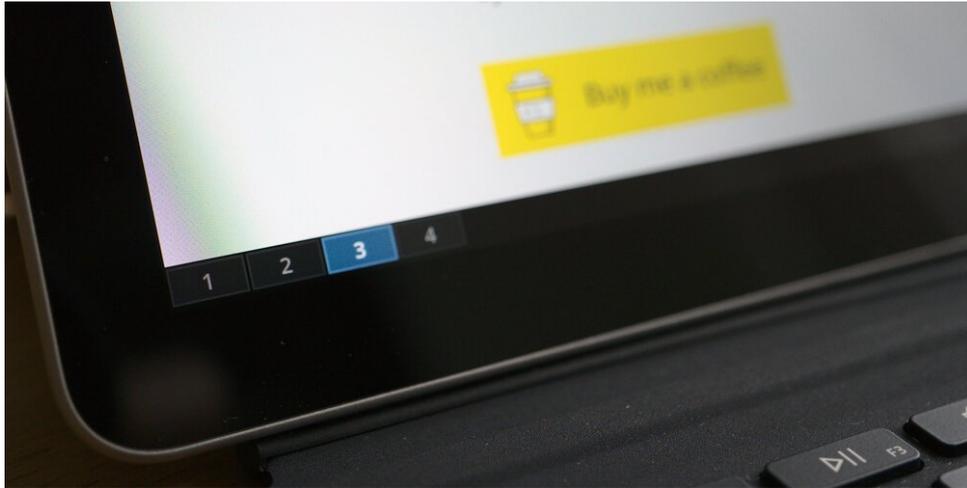
El principio de un gestor de ventanas en mosaico es simple: las aplicaciones ocupan tanto espacio como haya disponible. Esto significa que si ejecutas solo una aplicación, estará en pantalla completa. Dos aplicaciones se ejecutan en pantalla dividida, etc., etc. Puedes definir reglas que automaticen la disposición de las aplicaciones. Usando atajos de teclado, puedes cambiar rápidamente entre aplicaciones. Es posible que necesites algo de tiempo para descubrir qué funciona mejor para *ti* cuando usas un gestor de ventanas en mosaico, ¡pero al final puede ahorrarte mucho tiempo!



Ejecutando Firefox (izquierda) y una ventana de terminal MOSH/tmux (derecha) en pantalla dividida usando el gestor de ventanas en mosaico i3wm

Espacios de trabajo

Otra característica poderosa de i3wm es que puedes tener diferentes 'espacios de [trabajo](#)'. Son una forma fácil de organizar grupos de aplicaciones que se ejecutan juntas. Utilizo diferentes espacios de trabajo para diferentes cosas que hago. Un enfoque común es colocar el navegador web en un espacio de trabajo, las aplicaciones de correo en otro y aquellas con las que trabajas, en el tercero.



Los espacios de trabajo múltiples son muy potentes: es como cambiar de canal de televisión

Lo bueno de los múltiples espacios de trabajo es que una aplicación (por ejemplo, tu navegador) puede ejecutar múltiples instancias en diferentes espacios de trabajo. Esto permite muchas combinaciones diferentes de aplicaciones. Aquí es realmente donde el gestor de ventanas i3wm es *mucho* mejor que la versión multitarea de iPadOS.

El gestor de ventanas i3wm admite muy bien los monitores externos, utilizando el mismo concepto de un espacio de trabajo. Simplemente designas uno o varios espacios de trabajo para que se muestren en la pantalla adicional/externa. Esto funciona bien con Surface Dock y los accesorios USB-C a HDMI para Surface. La compatibilidad con varios monitores es realmente excelente.



Deslizar tres dedos para cambiar de espacio de trabajo, un truco de macOS, usando fusuma puedes usar gestos multitáctiles en i3wm

Una cosa que deberías probar es usar [fusuma](#) para habilitar gestos de tres dedos en

el trackpad. Si alguna vez has trabajado en una MacBook, sabrás que los trackpads multitáctiles habilitan gestos útiles para cambiar de escritorio (o espacios de trabajo en i3wm). ¡Funciona realmente bien!

Almacenamiento

Para [administrar mis fotos usando Shotwell](#) quería un almacenamiento serio en mi tableta. Aunque tengo el modelo superior de Surface Go 2 con la unidad SSD de 256 GB, utilicé la ranura para tarjetas MicroSD para ampliar mi capacidad de almacenamiento en otros 512 GB. La tarjeta MicroSD es más lenta que la SSD integrada, pero para archivos que no lees todos los días/horas/minutos es lo suficientemente rápida.



Ampliar el almacenamiento en un Surface Go 2 es posible usando una tarjeta MicroSD

Mi instalación de Debian GNU/Linux no tuvo problemas para reconocer la tarjeta MicroSD. Cifré toda la tarjeta MicroSD para evitar que mis fotos se filtraran si alguna vez perdía mi tarjeta. Siempre debes considerar esto, ¡toma las medidas adecuadas para proteger y respaldar tus datos!

Conectividad WiFi/4G/LTE

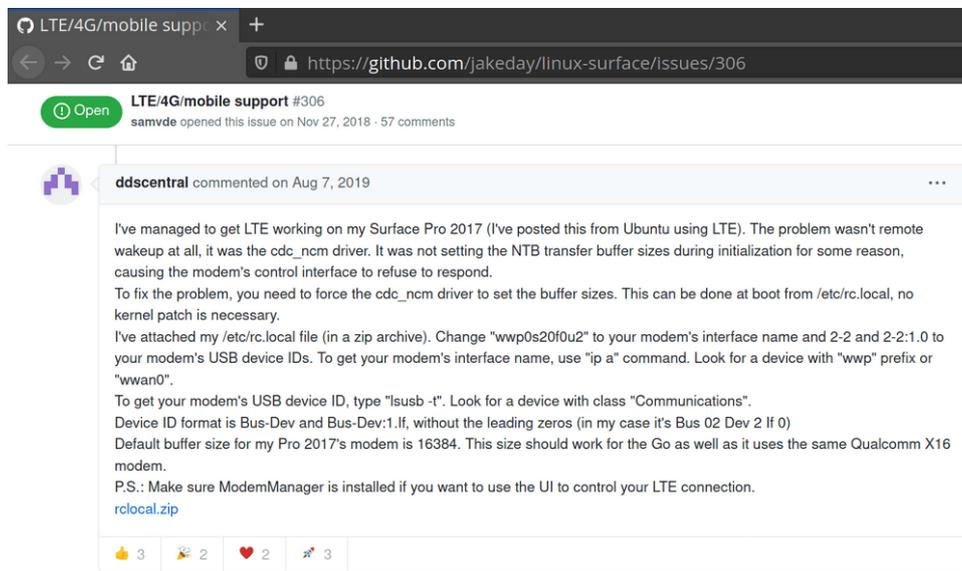
Después de instalar el firmware inalámbrico "no libre" en Debian, el WiFi funciona bien en Surface Go 2. ¡Hacer que LTE/4G funcionara fue mucho más difícil! Por alguna razón, la conexión WWAN no se inició, aunque inserté una tarjeta SIM que funcionaba en la Surface Go. Por alguna razón misteriosa, el módem no se inició correctamente, ¡4G/LTE no funcionó!



Instalando la SIM en la tableta Surface Go 2

Una de las mejores cosas del iPad es que comparte la lógica de conectividad del iPhone. Administra automáticamente diferentes redes inalámbricas y conexiones de banda ancha móvil. ¡Quería que esto también funcionara en mi tableta Linux!

Cuando buscaba una pista para que 4G/LTE funcionara, busqué informes sobre Surface Pro. La Surface Go 2 es técnicamente muy similar a la Pro más grande. Pensé que podría aprender de los esfuerzos de las personas con banda ancha móvil en la Surface Pro. Después de muchas horas de búsqueda y probando diferentes cosas, encontré [esta solución para hacer que 4G funcione en Surface](#):



¡Restablecer los búferes del módem de banda ancha Surface permitió que funcionara con Debian GNU/Linux!

Puedes usar el Network Manager (y Modem Manager) para automatizar la gestión de la conectividad en Linux. De forma predeterminada, preferirá las conexiones WiFi sobre la banda ancha móvil. Puedes hacer que se conecte automáticamente a través de 4G/LTE si no hay WiFi disponible. ¡Ahora funciona muy bien en combinación con el hardware de Surface Go 2!



Network Manager (nm-applet) cambia automáticamente entre conexiones WiFi y de banda ancha móvil

Suspender/Hibernar/Reanudar

Otra cosa en la que ningún usuario de iPad piensa jamás es en lo bien que Apple hizo las cosas en lo que respecta a suspender y reanudar. Cuando tienes un iPad, puedes tenerlo en modo de espera durante muchos días y, sin embargo, siempre es muy rápido continuar donde lo dejaste una vez que lo vuelves a recoger.

Para que mi Surface con GNU/Linux hiciera lo mismo, necesitaba una forma de suspender y reanudar rápidamente. En el mundo de Linux/PC, básicamente hay cuatro modos de energía:

- **encendido:** las aplicaciones se están ejecutando. El sistema está completamente encendido, consume mucha energía
- **suspender (a RAM):** los estados de las aplicaciones se conservan en la memoria RAM. El sistema aún está encendido, pero todos los componentes se apagan tanto como sea posible, aún consume energía (aproximadamente un 10 % de descarga de batería por hora en mi caso con Surface Go 2)
- **suspender (a disco)/hibernar:** los estados de las aplicaciones se conservan en el disco. El sistema está apagado y no consume energía.
- **apagado:** las aplicaciones están cerradas. El sistema está apagado, no se consume energía.

Para "reanudar instantáneamente donde lo dejaste", debes conservar los estados de las aplicaciones. Las reanudaciones más rápidas (y casi instantáneas) son posibles cuando el estado de la aplicación se almacena en la memoria RAM. El problema con la memoria RAM es que necesita corriente (¡energía!) para mantener su valor. Una vez que se pierde la energía, se pierden los estados de las aplicaciones.

La alternativa es suspender a disco (o hibernar), donde el estado se almacena en el dispositivo de almacenamiento permanente de la computadora. Permitirá que la computadora se apague por completo. El problema con la suspensión a disco es que la

reanudación es más lenta y requerirá volver a ingresar frases de contraseña de cifrado para el cifrado completo del disco.

La buena noticia es que el GNU/Linux moderno viene con `systemd` que te permite configurar cómo maneja la computadora la suspensión y la reanudación. Puedes configurar tu tableta para que primero suspenda a RAM (permitiendo la reanudación instantánea) y luego, después de un período de tiempo, suspenda a disco.

```
# See systemd-sleep.conf(5) for details

[Sleep]
AllowSuspend=yes
AllowHibernation=yes
AllowSuspendThenHibernate=yes
#AllowHybridSleep=yes
#SuspendMode=
#SuspendState=mem standby freeze
#HibernateMode=platform shutdown
#HibernateState=disk
#HybridSleepMode=suspend platform shutdown
#HybridSleepState=disk
#HibernateDelaySec=86400      # 24 hour
HibernateDelaySec=7200
```

La configuración de `systemd` "AllowSuspendThenHibernate" en `sleep.conf` te permite definir un tiempo de espera entre la suspensión a RAM y la suspensión a disco (hibernación)

En un día laboral normal, me despierto (muy) temprano para trabajar en mis proyectos más desafiantes desde casa. Luego [voy en bicicleta al trabajo](#) y sigo trabajando desde allí. Calculé el tiempo de mi viaje y pensé que si mi tableta permaneciera encendida durante al menos dos horas, me permitiría reanudar el trabajo instantáneamente. Si no continúo trabajando en mi tableta dentro de dos horas, ahorrará energía de la batería suspendiendo a disco. De esta manera tengo la mejor experiencia para mi situación particular.

```
# See logind.conf(5) for details.

[Login]
HandleLidSwitch=suspend-then-hibernate
HandleLidSwitchExternalPower=suspend-then-hibernate
```

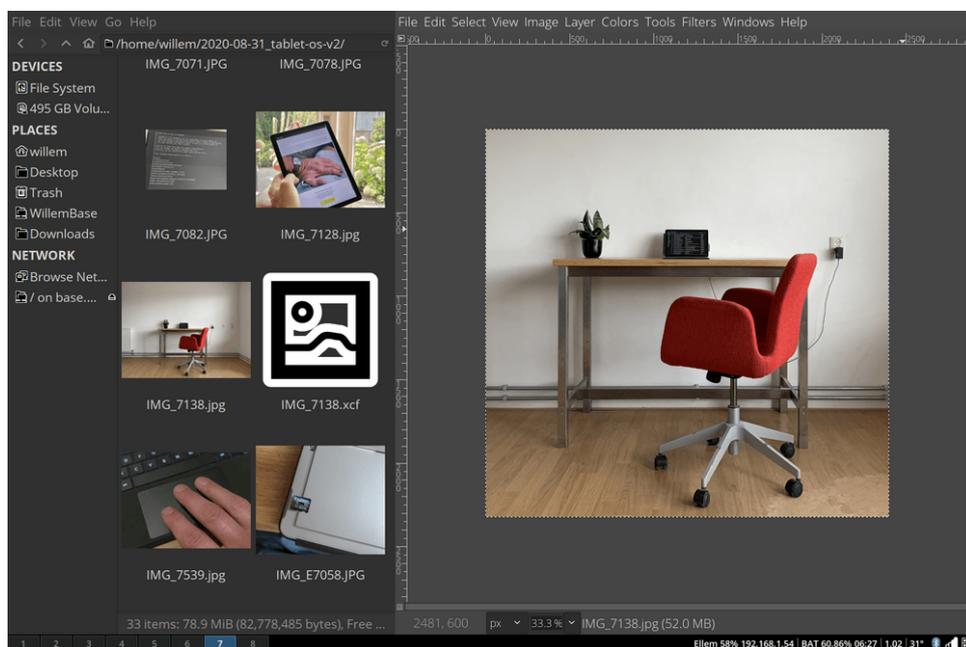
*Configura `LidSwitch` (en `login.conf` de `systemd`) para suspender y reanudar automáticamente cuando cierras la *Type Cover* de Surface, yo lo he configurado para suspender y luego hibernar usando un tiempo de espera*

Apariencia uniforme

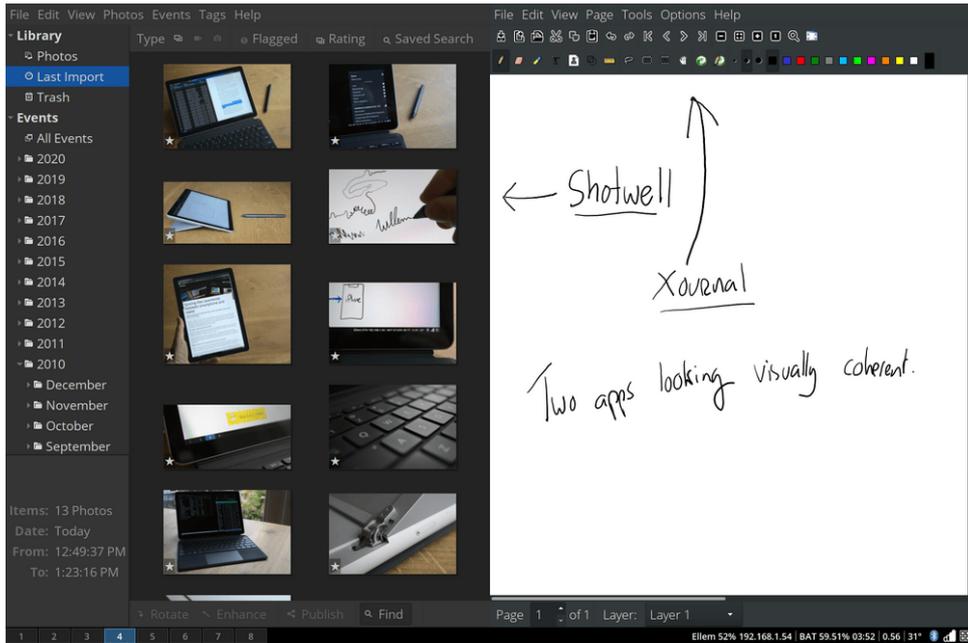
A diferencia del iPad, estrechamente controlado, el software en GNU/Linux se crea utilizando muchos enfoques diferentes para el diseño de la interfaz de usuario. No existe tal cosa como las [Pautas de la interfaz de usuario](#) que Apple dicta para iOS y iPadOS. Este es un problema, ya que hace que el software se vea y se sienta inconsistente.

Una apariencia uniforme es importante para permitir un entorno y una experiencia de usuario sin fricciones. Por lo tanto, hice muchos pequeños ajustes en el color, las fuentes, los iconos y las barras de herramientas para que todas mis aplicaciones se vieran y se sintieran similares. Aunque no es perfectamente armonioso, ya no se siente como si estuviera trabajando con "Frankenstein OS".

Prefiero un tema oscuro, ya que es más agradable para mis ojos, pero debes decidir por ti mismo. Usa herramientas como **lxappearance** para realizar ajustes para múltiples aplicaciones. Pero, probablemente también necesites realizar ajustes en cada aplicación individual. Utilizo el tema **adwaita-dark** y los iconos (monocromáticos) **HighContrast**, con la fuente **Open Sans Light**.



Explorador de archivos Thunar (izquierda) y GIMP (derecha) en armonía visual: crea una apariencia uniforme combinando colores, iconos y fuentes



Shotwell (izquierda) y Xournal (derecha): dos aplicaciones muy diferentes ejecutándose una al lado de la otra, con una apariencia visualmente coherente

Usándolo como controlador diario

Construir un sistema hermoso es una cosa, ¡pero usarlo como controlador diario es otra cosa! La Surface Go es mi controlador diario y durante las últimas semanas he estado experimentando con diferentes cosas y configuraciones. Aquí hay algunos aspectos destacados.



Surface Pen es genial y cuando configuras Xournal para usar Xinput, la sensibilidad a la presión funciona perfectamente: puedes ver cómo varía el grosor de la línea, como un bolígrafo real



El soporte es brillante en combinación con Xournal y Surface Pen: ningún iPad puede hacer esto



El soporte es muy resistente ya que utiliza dos fuertes bisagras de metal



Escribir en la Type Cover es un encanto con teclas reales (retroiluminadas), incluyendo ESC y una fila de teclas de función



También puedes usar un teclado Bluetooth, lo que permite una distancia cómoda entre tus ojos y la pantalla



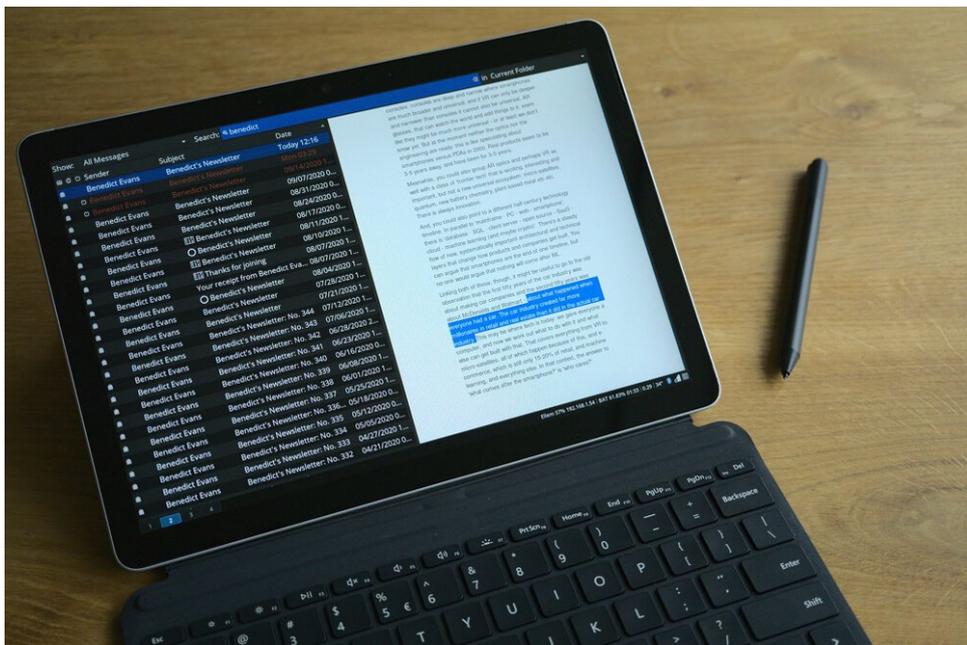
Si bien USB-C es el futuro, el puerto conector Surface es bastante útil ya que funciona con Surface Dock 2 (que ofrece una gran cantidad de conectividad)



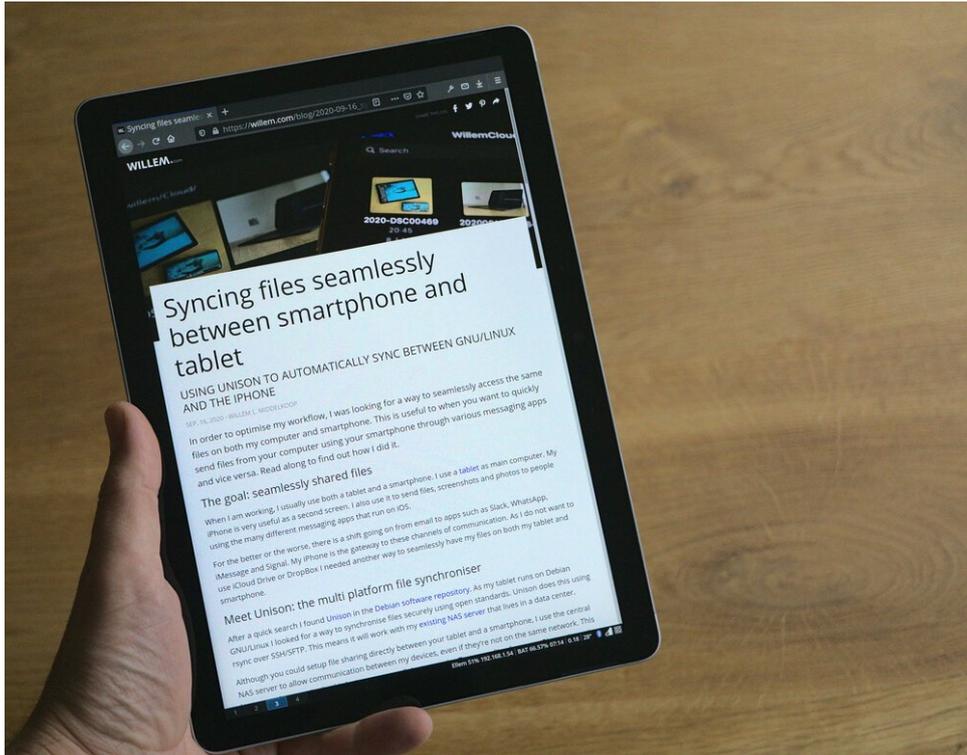
La parte posterior de la Surface no tiene una lente de cámara que sobresalga, lo que permite que la tableta se apoye perfectamente plana sobre una mesa, ¡como debería ser!



¡Es una computadora pequeña, pero es increíblemente potente y versátil!



¡Leer correos electrónicos en Evolution funciona bien!



Es lo suficientemente liviana como para manejarla con una mano; puedes leer el sensor de giroscopio para rotar automáticamente la pantalla, útil cuando lees un artículo (interesante)



La Surface Go 2 es una herramienta seria, capaz de cosas poderosas cuando te conectas a la nube (usando una terminal, MOSH y tmux)

Conclusión

Construir mi propio sistema operativo para tableta es un verdadero desafío. Mi primer intento no fue lo suficientemente bueno como para reemplazar mi iPad. Sin embargo, este segundo intento es diferente. El tiempo y el esfuerzo que dediqué a refinar la experiencia del usuario dan como resultado un sistema mucho mejor. Ahora lo uso como mi computadora principal a diario.

Aunque siempre aprecié el trabajo que Apple ha hecho en el iPad, sigue estando limitado por su software cerrado. Especialmente ahora, cuando Apple ha decidido [duplicar su control de la AppStore](#). La única forma de controlar el futuro es crearlo tú mismo: ¡creo que es el momento adecuado para convertirme en capitán de mi propio barco!

Actualización de enero de 2021

[Después de meses de uso intenso he decidido abandonar mi propio sistema operativo para tableta](#)