



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA:

Principios de la programación para dispositivos

Wearables

PRESENTADO POR:

Hernández Miranda Rafael Francisco

GRUPO:

9B

MATERIA:

Desarrollo para Dispositivos Inteligentes

PROFESOR:

Ray Brunett Parra Galaviz

FECHA:

11/09/2024.

Desarrollo de aplicaciones para smartwatch y wearable Android e iOS

Vincular diferentes dispositivos como teléfonos móviles, relojes inteligentes, televisores, maquinaria en empresas o incluso electrodomésticos en los hogares nunca había sido tan sencillo. La integración de apps con sistemas IoT ofrece mejores experiencias a los usuarios, mayor valor y nuevas funcionalidades.

Hoy en día, las aplicaciones ya no son solo cosa de smartphones, ahora el desarrollo de apps para smartwatch y wearable suponen una nueva forma de comunicación y permiten crear oportunidades únicas para llegar a los usuarios.

La tecnología nos ha permitido aprovechar al máximo las funcionalidades de los dispositivos móviles y poder trabajar el desarrollo de aplicaciones para smartwatch y wearables adaptándonos a cada cliente y los principales principios que rigen en esta rama son:

1. **Portabilidad:** Los dispositivos wearables deben ser livianos y cómodos, ya que están diseñados para ser usados por largos periodos.
2. **Durabilidad:** Los wearables deben ser resistentes al desgaste, al agua y otros factores ambientales que puedan afectar su funcionamiento.
3. **Interfaz de Usuario (UI) Simples:** Dado que los wearables tienen pantallas pequeñas o ninguna, las interfaces deben ser intuitivas y simples de usar. La interacción mediante gestos, toques y comandos de voz es común.
4. **Conectividad:** La mayoría de los wearables necesitan una conexión constante a Internet, ya sea a través de Wi-Fi, Bluetooth o redes celulares para sincronizarse con otros dispositivos o plataformas.
5. **Bajo consumo de energía:** Los wearables suelen tener baterías pequeñas, por lo que es importante optimizar su consumo de energía para que puedan durar todo el día o más sin necesidad de recarga.
6. **Seguridad y privacidad:** Dado que los wearables a menudo recopilan datos sensibles, como datos de salud o ubicación, es fundamental garantizar la seguridad y privacidad de estos datos.
7. **Compatibilidad:** Los wearables deben poder integrarse fácilmente con una variedad de sistemas operativos, aplicaciones y dispositivos, como smartphones o plataformas de fitness.
8. **Monitoreo y Sensores:** La mayoría de los wearables están equipados con sensores (por ejemplo, sensores de ritmo cardíaco, giroscopios, acelerómetros) que les permiten recolectar datos sobre la actividad del usuario.
9. **Actualizaciones de Software:** Es esencial que los wearables puedan recibir actualizaciones de software para mejorar el rendimiento, añadir nuevas funciones y mantener la seguridad del sistema.

Diferencias en la Arquitectura de Software para Wearables vs. Otros Dispositivos Inteligentes

1. Optimización para Recursos Limitados

- **Wearables:** La arquitectura de software debe estar optimizada para trabajar con recursos limitados como batería, procesador y memoria. Las aplicaciones tienden a ser ligeras, con procesos eficientes en segundo plano para maximizar la duración de la batería.
 - **Ejemplo:** Un smartwatch debe priorizar procesos de baja energía y minimizar las conexiones a redes para ahorrar batería.
- **Otros Dispositivos Inteligentes:** Dispositivos como tabletas, consolas de videojuegos o sistemas domésticos inteligentes (por ejemplo, asistentes de voz como Alexa) tienen más capacidad de procesamiento, mayor espacio de almacenamiento y pueden ejecutar tareas más complejas sin un enfoque tan estricto en el consumo de energía.
 - **Ejemplo:** Un asistente de voz inteligente en el hogar tiene más potencia y puede ejecutar procesos complejos como la integración de varios servicios a la vez sin preocuparse mucho por el consumo de energía.

2. Interacción de Usuario y UI/UX

- **Wearables:** La interfaz de usuario está diseñada para pantallas pequeñas o incluso para dispositivos sin pantalla (como audífonos). Las interacciones suelen ser gestuales, táctiles, o por comandos de voz, lo que requiere una arquitectura que soporte interfaces simplificadas y ágiles.
 - **Ejemplo:** En un reloj inteligente, el software debe optimizar la navegación por una pantalla pequeña con gestos intuitivos y botones grandes.
- **Otros Dispositivos Inteligentes:** En dispositivos como televisores inteligentes, asistentes de hogar o consolas de juegos, la interfaz puede ser más rica, con pantallas grandes y opciones de interacción más complejas, como controles remotos, teclados o mandos de juego.
 - **Ejemplo:** Un televisor inteligente ofrece una interfaz completa con múltiples aplicaciones y secciones organizadas para navegación remota o táctil.

3. Procesamiento de Datos

- **Wearables:** Muchos wearables necesitan procesar datos en tiempo real, como la actividad física o datos biométricos, lo que requiere una arquitectura de software eficiente que priorice el procesamiento rápido y seguro de información sensible, sin sobrecargar los recursos del dispositivo.
 - **Ejemplo:** Una pulsera de fitness debe procesar continuamente los datos del ritmo cardíaco y los pasos del usuario, sin agotar la batería.

- **Otros Dispositivos Inteligentes:** Los dispositivos inteligentes como cámaras de seguridad, termostatos o consolas de juegos pueden almacenar o procesar grandes cantidades de datos localmente o en la nube, con menos limitaciones en términos de tiempo real.
 - **Ejemplo:** Una cámara de seguridad inteligente graba y almacena datos en la nube, pero no necesita la misma capacidad de procesamiento en tiempo real que un dispositivo wearable que monitoriza el pulso.

4. Conectividad

- **Wearables:** Los wearables suelen depender de la conectividad constante con otros dispositivos, como smartphones, mediante Bluetooth o Wi-Fi. La arquitectura de software está diseñada para sincronizar datos entre el wearable y el dispositivo vinculado.
 - **Ejemplo:** Un reloj inteligente debe sincronizarse continuamente con un smartphone para mostrar notificaciones y actualizar datos.
- **Otros Dispositivos Inteligentes:** En otros dispositivos, como altavoces inteligentes o termostatos, la conectividad también es importante, pero estos pueden ser más autónomos. No dependen necesariamente de otro dispositivo para funcionar de manera óptima.
 - **Ejemplo:** Un altavoz inteligente puede funcionar de manera independiente para reproducir música o controlar otros dispositivos domésticos sin necesidad de conexión constante con otro dispositivo.

Diferencias en la Arquitectura de Software para Wearables vs. Otros Dispositivos Inteligentes

1. Optimización para Recursos Limitados

- **Wearables:** La arquitectura de software debe estar optimizada para trabajar con recursos limitados como batería, procesador y memoria. Las aplicaciones tienden a ser ligeras, con procesos eficientes en segundo plano para maximizar la duración de la batería.
 - **Ejemplo:** Un smartwatch debe priorizar procesos de baja energía y minimizar las conexiones a redes para ahorrar batería.

- **Otros Dispositivos Inteligentes:** Dispositivos como tabletas, consolas de videojuegos o sistemas domésticos inteligentes (por ejemplo, asistentes de voz como Alexa) tienen más capacidad de procesamiento, mayor espacio de almacenamiento y pueden ejecutar tareas más complejas sin un enfoque tan estricto en el consumo de energía.
 - **Ejemplo:** Un asistente de voz inteligente en el hogar tiene más potencia y puede ejecutar procesos complejos como la integración de varios servicios a la vez sin preocuparse mucho por el consumo de energía.

2. Interacción de Usuario y UI/UX

- **Wearables:** La interfaz de usuario está diseñada para pantallas pequeñas o incluso para dispositivos sin pantalla (como audífonos). Las interacciones suelen ser gestuales, táctiles, o por comandos de voz, lo que requiere una arquitectura que soporte interfaces simplificadas y ágiles.
 - **Ejemplo:** En un reloj inteligente, el software debe optimizar la navegación por una pantalla pequeña con gestos intuitivos y botones grandes.
- **Otros Dispositivos Inteligentes:** En dispositivos como televisores inteligentes, asistentes de hogar o consolas de juegos, la interfaz puede ser más rica, con pantallas grandes y opciones de interacción más complejas, como controles remotos, teclados o mandos de juego.
 - **Ejemplo:** Un televisor inteligente ofrece una interfaz completa con múltiples aplicaciones y secciones organizadas para navegación remota o táctil.

3. Procesamiento de Datos

- **Wearables:** Muchos wearables necesitan procesar datos en tiempo real, como la actividad física o datos biométricos, lo que requiere una arquitectura de software eficiente que priorice el procesamiento rápido y seguro de información sensible, sin sobrecargar los recursos del dispositivo.
 - **Ejemplo:** Una pulsera de fitness debe procesar continuamente los datos del ritmo cardíaco y los pasos del usuario, sin agotar la batería.
- **Otros Dispositivos Inteligentes:** Los dispositivos inteligentes como cámaras de seguridad, termostatos o consolas de juegos pueden almacenar o procesar grandes cantidades de datos localmente o en la nube, con menos limitaciones en términos de tiempo real.
 - **Ejemplo:** Una cámara de seguridad inteligente graba y almacena datos en la nube, pero no necesita la misma capacidad de procesamiento en tiempo real que un dispositivo wearable que monitoriza el pulso.

4. Conectividad

- **Wearables:** Los wearables suelen depender de la conectividad constante con otros dispositivos, como smartphones, mediante Bluetooth o Wi-Fi. La arquitectura de software está diseñada para sincronizar datos entre el wearable y el dispositivo vinculado.
 - **Ejemplo:** Un reloj inteligente debe sincronizarse continuamente con un smartphone para mostrar notificaciones y actualizar datos.
- **Otros Dispositivos Inteligentes:** En otros dispositivos, como altavoces inteligentes o termostatos, la conectividad también es importante, pero estos pueden ser más autónomos. No dependen necesariamente de otro dispositivo para funcionar de manera óptima.
 - **Ejemplo:** Un altavoz inteligente puede funcionar de manera independiente para reproducir música o controlar otros dispositivos domésticos sin necesidad de conexión constante con otro dispositivo.

References

- (N.d.-a). Abamobile.com. Retrieved September 12, 2024, from <https://abamobile.com/web/desarrollo-de-apps-para-smartwatches-y-wearables/>
- (N.d.-b). Iberdrola.com. Retrieved September 12, 2024, from <https://www.iberdrola.com/innovacion/tecnologia-wearable>
- (N.d.-c). Com.Mx. Retrieved September 12, 2024, from <https://sg.com.mx/sgvirtual/slides/2014-Wearables.pdf>
- (N.d.-d). Iberdrola.com. Retrieved September 12, 2024, from <https://www.iberdrola.com/innovacion/tecnologia-wearable#:~:text=La%20gran%20diferencia%20respecto%20a,estar%20a%20flor%20de%20piel.>
- (N.d.-e). Ujaen.Es. Retrieved September 12, 2024, from https://crea.ujaen.es/bitstream/10953.1/19965/1/David_TFM.pdf
- (N.d.-f). Mokosmart.com. Retrieved September 12, 2024, from <https://www.mokosmart.com/es/comparison-of-smart-wearable-on-different-technologies/>