



ITSSAT

**INSTITUTO TECNOLÓGICO SUPERIOR
DE SAN ANDRÉS TUXTLA**

Carrera: Ing. en Informática

Módulo: Administración y organización de datos

Catedrático: MTI. Rogelio Enrique Telona Torres

Semestre: 4to. Semestre

Grupo: 410A

Alumno:

Christian Manuel Millán Pólito

Periodo Escolar: Febrero - Julio 2022

San Andrés Tuxtla, Ver.

Introducción

En este documento se presentan la investigación documental de la Unidad 2, iniciando con el tema Unidad 2, se presentarán los SO emergentes para aplicaciones móviles, las características relevantes de los distintos entornos de desarrollo asociados a los sistemas operativos de la actividad anterior, los datos más relevantes de estos entornos y lenguajes, adjuntando imágenes para su mejor comprensión.

SO Emergentes para Aplicaciones Móviles

Los sistemas operativos móviles son plataformas de software diseñadas específicamente para dispositivos móviles, como **teléfonos inteligentes y tabletas**. Estos sistemas operativos están optimizados para ejecutarse en hardware móvil con recursos limitados, pantallas táctiles y capacidades de comunicación inalámbrica:

Android

Android es un **sistema operativo móvil desarrollado por Google**. Se basa en el kernel de Linux y es conocido por su amplia adopción en dispositivos móviles de diversas marcas. Una de las características más destacadas de Android es su naturaleza de **código abierto**, lo que significa que su código fuente está disponible para ser modificado y personalizado por fabricantes y desarrolladores.

Los usuarios pueden descargar aplicaciones, juegos, herramientas y utilidades para personalizar su experiencia en los **dispositivos Android**. Además, Android proporciona una interfaz de usuario intuitiva y altamente personalizable, lo que permite a los usuarios adaptar el aspecto y la configuración de sus dispositivos según sus preferencias.

Cada versión de Android también lleva el nombre de un postre en orden alfabético, como **Cupcake, Donut, KitKat, Oreo, etc.** Los fabricantes de dispositivos Android a menudo personalizan la apariencia y características de Android en sus propios dispositivos a través de capas de personalización llamadas "skins".

iOS

iOS es el sistema operativo desarrollado por Apple exclusivamente para sus dispositivos móviles, como el **iPhone, iPad y iPod Touch**. A diferencia de Android, iOS es un sistema operativo cerrado y propietario, diseñado específicamente para funcionar en hardware fabricado por Apple.

La principal característica distintiva de iOS es su enfoque en la **simplicidad y la elegancia**. La interfaz de usuario de iOS se basa en gestos táctiles y es conocida por su diseño limpio y coherente. Apple también es reconocido por su enfoque en la seguridad y

la privacidad, incorporando características como **Touch ID y Face ID** para la autenticación biométrica.

La App Store es la plataforma oficial de distribución de aplicaciones para iOS, que ofrece una amplia selección de aplicaciones desarrolladas tanto por Apple como por terceros.

BlackBerry OS

BlackBerry OS es el sistema operativo desarrollado por **BlackBerry Limited** para sus dispositivos móviles. Históricamente, BlackBerry fue pionero en el campo de las comunicaciones móviles y se destacó por su enfoque en la seguridad y la productividad.

BlackBerry OS se caracterizaba por su capacidad de administración y seguridad empresarial, lo que lo convirtió en una opción popular para los usuarios corporativos y gubernamentales. Además, los dispositivos BlackBerry eran conocidos por su **teclado físico QWERTY**, que ofrecía una experiencia de escritura cómoda y rápida.

Sin embargo, el sistema operativo BlackBerry OS ha perdido terreno frente a la competencia de Android e iOS. Para adaptarse a las nuevas demandas del mercado, BlackBerry ha migrado hacia el uso de **sistemas operativos basados en Android** en sus dispositivos más recientes.

Tizen

Tizen es un sistema operativo de código abierto impulsado por la **Linux Foundation** y respaldado por Samsung. Aunque Tizen no ha alcanzado la misma popularidad que Android o iOS, se utiliza en una variedad de dispositivos Samsung, como **teléfonos inteligentes, smartwatches, televisores inteligentes y otros dispositivos conectados**.

Tizen se basa en el kernel de **Linux** y se ha desarrollado como una plataforma versátil y de alto rendimiento. Ofrece una interfaz de usuario intuitiva y personalizable, y se integra con servicios y aplicaciones populares.

La flexibilidad de Tizen permite a los fabricantes adaptarlo a sus necesidades específicas, lo que ha llevado a su adopción en diferentes tipos de dispositivos electrónicos. Aunque Tizen todavía enfrenta el desafío de competir con los sistemas operativos más

establecidos, su desarrollo continuo y el **respaldo de Samsung** podrían llevarlo a un mayor crecimiento y expansión en el futuro.

Phone Ubuntu

Utiliza el Framework Qt 5 basado en la interfaz de usuario táctil y varios marcos de software desarrollados originalmente para Maemo y MeeGo como oFono. Además cuenta con un inicio de sesión único, utilizando *libhybris*, sistema que se usa con núcleos Linux utilizadas en Android, lo que facilita que sea portado a los últimos teléfonos inteligentes Android.



Ubuntu Touch utiliza las mismas tecnologías esenciales del Escritorio de Ubuntu, por lo que las aplicaciones diseñadas para esta plataforma pueden ser usada en ambas. Además, los componentes de escritorio de Ubuntu vienen con el sistema Ubuntu Touch, permitiendo que los dispositivos táctiles de Ubuntu puedan proporcionar una completa experiencia de escritorio cuando se conecta a un monitor externo. Los dispositivos táctiles de Ubuntu pueden estar equipados con una sesión completa de Ubuntu y pueden cambiar por completo el escritorio del sistema operativo cuando se conecta a una estación de acoplamiento. Si está conectado el dispositivo se pueden utilizar todas las características de Ubuntu y el usuario puede realizar trabajo de oficina o incluso jugar juegos en ARM mediante el dispositivo.

Algunas de sus características más destacadas son:

- Pantalla de inicio sin sistema de bloqueo/desbloqueo (funciona con un nuevo sistema de gestos y se aprovecha para mostrar notificaciones).

- Ubuntu Touch incluye como aplicaciones centrales de medios sociales y medios de comunicación. Las aplicaciones estándar, tales como una calculadora, un cliente de correo electrónico, un despertador, un gestor de archivos, e incluso un terminal.
- Integración con Ubuntu One.

¿Cómo se clasifican los sistemas operativos móviles?



Los sistemas operativos móviles se pueden clasificar de varias formas, considerando diferentes criterios:

Por cuota de mercado

En la actualidad, los sistemas operativos móviles más utilizados son **Android, iOS y en menor medida, otros sistemas** como Windows Phone, BlackBerry OS y Tizen.

Por licencia

Algunos sistemas operativos son de código abierto, lo que significa que el código fuente es accesible y modificable por cualquier persona. Android es un ejemplo destacado de **sistema operativo de código abierto**. Por otro lado, existen sistemas operativos propietarios como iOS, que son desarrollados y distribuidos exclusivamente por una empresa específica.

Por arquitectura de hardware

Los sistemas operativos móviles también pueden clasificarse en función de la arquitectura de hardware que soportan. Por ejemplo, algunos sistemas operativos están diseñados para funcionar en **procesadores basados en arquitectura ARM**, mientras que otros están orientados a dispositivos con procesadores x86.

Por interfaz de usuario

Algunos sistemas operativos móviles utilizan una interfaz de usuario basada en gestos táctiles, mientras que otros pueden enfocarse en **interfaces más tradicionales** con botones físicos o un teclado QWERTY.

Características relevantes de los distintos entornos de desarrollo asociados a los sistemas operativos de la actividad anterior.

Entornos para crear aplicaciones Android

Desarrollar apps Android sin recurrir a Java es posible y te lo vamos a demostrar en este listado de 6 entornos de programación. Vamos allá.

#1 NativeScript

NativeScript es un **entorno de desarrollo** de código abierto para crear aplicaciones móviles Android e iOS, con Angular, Vue.js, TypeScript o JavaScript. Las principales ventajas de este entorno de programación es el rendimiento nativo. Solo tendrás que definir una vez y NativeScript se adaptará para ejecutarse en todas partes, **adaptando la IU de dispositivos y pantallas concretas.**

Otro de sus puntos a favor es su **escalabilidad**. Tendrás acceso a las APIs de Android, pero podrás reutilizar el SDK de Android, además de encontrar complementos gratuitos o plantillas en el marketplace de NativeScript.

Además, si comienzas a programar apps para Android con NativeScript, verás que **es fácil de aprender**. Podrás aplicar los conocimientos que ya tienes en JavaScript y CSS, pero además, tendrás a tu disposición una gran cantidad de **recursos, tutoriales, ejemplos y una comunidad de usuarios** fuerte con que la podrás ponerte en marcha antes de lo que imaginas.

#2 React Native

React Native es un **framework** desarrollado por Facebook que permite crear aplicaciones para Android nativas utilizando **JavaScript**. Esto es algo que ya permitían otros como **Cordova**, pero con la diferencia de que con React Native las apps se renderizan utilizando las views nativas, mientras que en Cordova se ejecutan dentro de una webview.

React Native tiene acceso a las API de Android, por lo que la experiencia de usuario y **el rendimiento es similar al de una aplicación nativa**. Otra de sus ventajas es la facilidad de ver los cambios con tan solo guardar. Gracias a JavaScript, React Native permite iterar rápidamente, sin tener que esperar a que finalicen las compilaciones nativas.

Sin duda este framework es una apuesta segura, al contar con el respaldo de Facebook pero, sobre todo, con una gran comunidad de usuarios. En 2018, React Native llegó a tener **el segundo mayor número de contribuciones en los repositorios de GitHub**.

#3 Ionic

Ionic es un **framework** de código abierto para desarrollar **aplicaciones híbridas**, en el que se combinan **Html5, CSS y Javascript**. Gratis y de código abierto, Ionic ofrece una biblioteca de componentes, gestos y herramientas de UI optimizados para dispositivos móviles con los que crear aplicaciones rápidas y altamente interactivas.

Si vienes desde el desarrollo web y quieres comenzar a programar en Android, Ionic es la mejor opción. Podrás seguir utilizando las mismas herramientas que ya conoces.

Ionic 5, versión más reciente del framework, está pensada para un **rendimiento rápido en cualquier dispositivo**. Con él podrás crear apps Android rápidas, optimizadas y que integran las mejores prácticas recomendadas. Además, Ionic facilita acceder a las funcionalidades nativas a través de JavaScript. Tienes a tu disposición una **librería de plugins nativos** para acceder a la geolocalización, Bluetooth o a la cámara sin grandes complejidades.

Por otra parte, el lenguaje C# es **más sencillo de aprender** que otros lenguajes de programación. Así que si Java no es tu fuerte, puede que Xamarin sea la mejor solución para crear aplicaciones Android de calidad.

#5 Kotlin

[Kotlin](#) es un **lenguaje de programación** de tipado estático que corre sobre la máquina virtual de Java. Sin embargo, crear aplicaciones Android con Kotlin tiene la ventaja de la simplicidad. En Kotlin se puede hacer más con menos código, lo que se traduce en una menor probabilidad de errores.

Otra de las ventajas de Kotlin es que es mucho más seguro que Java, y el código es más fácil de entender cuando lo lee otra persona. Además el compilador ayuda a detectar errores en la compilación, lo que contribuye a reducir posibles errores. Con Kotlin podrás programar en Android, pero también para iOS puesto que es multiplataforma. Todo ello hace que Kotlin sea **una alternativa segura y fiable frente a Java**.

#6 Flutter

Flutter es un **SDK** de Google pensado para crear aplicaciones Android e iOS, para móvil, web y escritorio, desde una única base de código. Su lenguaje de programación es Dart y se caracteriza porque hace posible un desarrollo rápido gracias a la funcionalidad **Hot Reload**. Con ella, pueden hacerse cambios sin tener que parar y arrancar la aplicación, con lo que la programación es más efectiva ya que se reducen las esperas.

Este **framework** permite crear funcionalidades rápidamente, con una experiencia de usuario nativa. Sin duda es uno de los SDK con más proyección, puesto que todo apunta a que será con el que se desarrollen los nuevos dispositivos Android.

Tizen

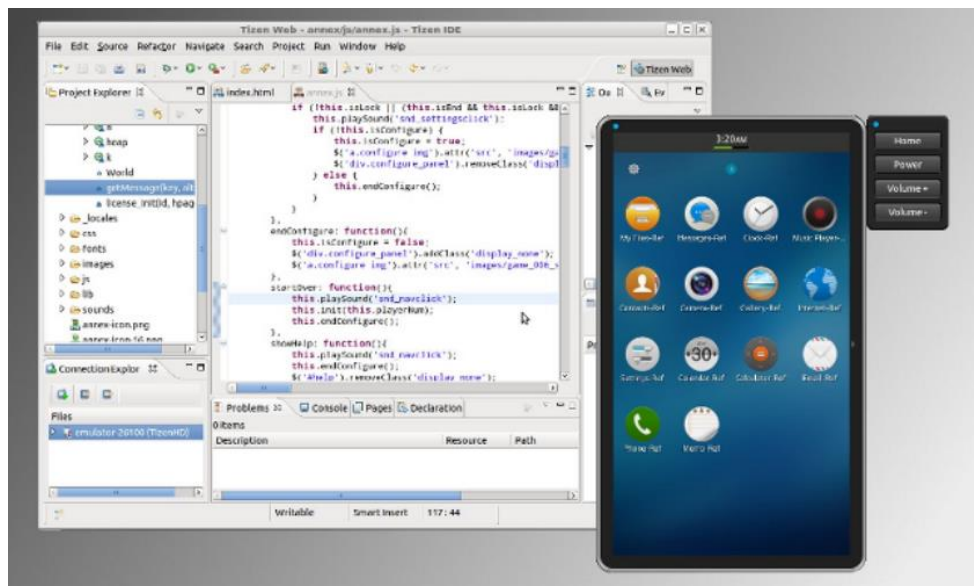
Tizen se fundamenta en estándares y es una plataforma de código abierto basada en Linux. Para el desarrollo de aplicaciones se ha elegido HTML 5 como entorno de desarrollo de sus aplicaciones.

Gracias al Taller de Tizen organizado por MadridJS y impartido por Intel he podido saber las líneas maestras de su plataforma de desarrollo, API y SDK.

Sólo hay que visitar la página de desarrollo para descargarse el SDK y las herramientas de desarrollo. Los sistemas operativos soportados son por el momento Ubuntu (11.10 o 12.04), Windows 7 y Apple Mac OS (10.7, 10.8).

El entorno de desarrollo está basado en Eclipse, por lo que le resultará familiar a muchos. Todo está pensado para facilitar la vida al desarrollador. Se ha hecho especial hincapié en la documentación y ejemplos, de tal manera que no es necesario buscar recursos externos.

En la misma línea de mejorar la experiencia de desarrollo, existen dos herramientas para probar y debugear tus aplicaciones, el simulador y el emulador.



El simulador está basado en Chrome (navegador) y permite desplegar y hacer las pruebas rápidamente. Aquel que sabe lo que puede tardar en arrancar el emulador de Android bendecirá este 'gatchet'. El emulador es una máquina virtual completa de Tizen, más parecido a un dispositivo físico. También se puede probar y depurar en el teléfono directamente. Tanto para el emulador como para desplegar en el teléfono físico es necesario poseer un certificado (para pruebas puede ser autofirmado). Existe una utilidad en el SDK que te permite generarlo.

Xcode como IDE de desarrollo

El entorno de desarrollo oficial de Apple se llama Xcode y es gratuito para cualquier desarrollador. Obtenerlo es tan sencillo como ir a la App Store de Apple y buscar el software por su nombre.

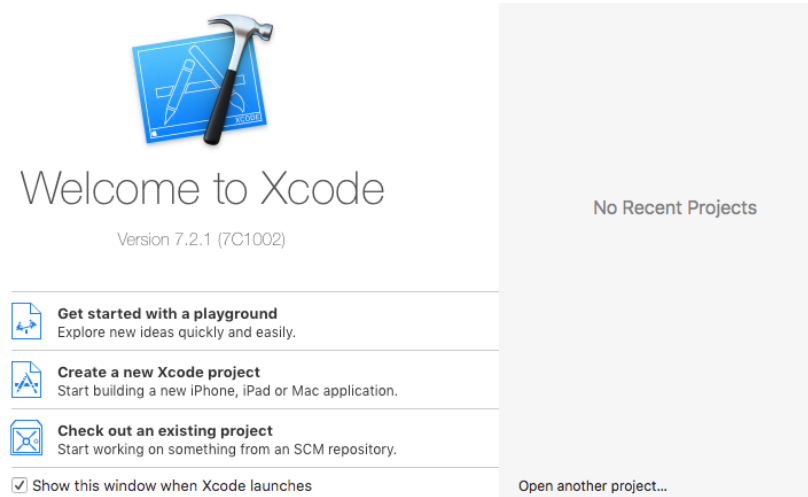
Existen otras opciones de software que se podrían usar para el desarrollo nativo de aplicaciones en iOS, pero no son las más recomendables, dado que el más actualizado y con mayores funciones siempre será el oficial de la compañía. No obstante, por si alguien no se adapta a Xcode podemos destacar AppCode de JetBrains, un desarrollador con una larga trayectoria en la creación de entornos y que ha sabido recoger su experiencia en AppCode.

Encontraremos Xcode como primera de las opciones. Solo necesitamos pulsar el botón de Instalar.



El programa es pesado (varios gigas), por lo que la descarga no será inmediata, así que puedes seguir leyendo mientras tanto (o tomarte un café o las dos cosas). Al arrancar Xcode por primera vez nos pedirá permiso para instalar componentes adicionales, lo otorgamos confirmando con la clave de nuestro usuario Mac.

Una vez instalado ya tenemos todo lo que necesitamos para comenzar a desarrollar para iOS... ¡Pero realmente ahora comienza nuestro trabajo!! La primera ventana que veremos es la siguiente:



Entre las opciones disponibles nos interesarán las dos primeras. La que recomendamos para comenzar a introducirnos en el desarrollo para iOS y el uso de los lenguajes de Apple es la nombrada como Get started with a playground. Esta opción crea un espacio de trabajo meramente experimental, pero perfecto para poder comenzar a probar las herramientas de desarrollo de Apple. Ahora bien, si ya deseamos comenzar a desarrollar nuestro proyecto de aplicación entraremos en la sección Create a new Xcode project, que nos ofrecerá el IDE en su plena funcionalidad.

Nuestro primer ejemplo con Swift

Como comentábamos antes, comenzaremos con la opción playground, que nos permitirá programar con Swift sin abrumarnos con todas las opciones que encontrarás dentro de Xcode. Ahora que estamos aprendiendo y hasta que tomemos algo de soltura, es lo más adecuado.

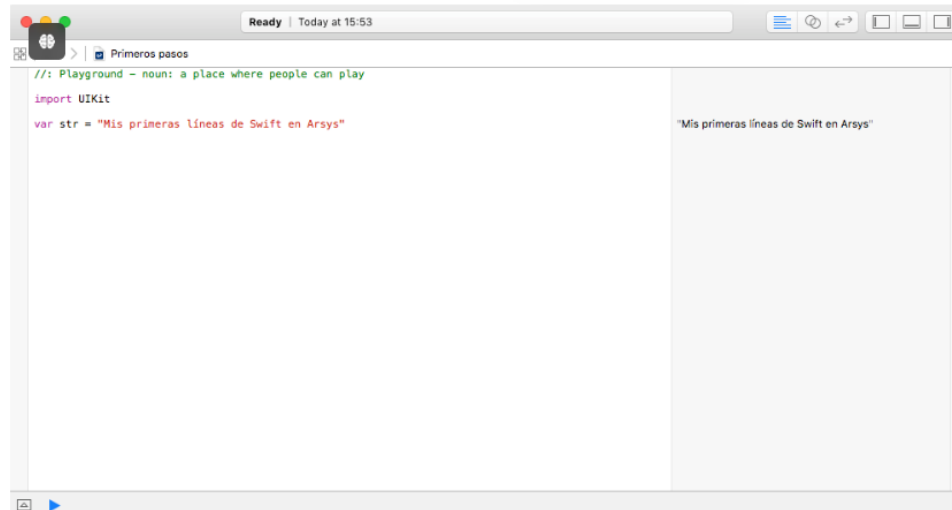
Como lenguaje para el desarrollo nativo en iOS podríamos elegir entre Objective-C y Swift. Éste último está siendo más fuertemente apoyado por Apple últimamente y se espera una mayor trayectoria a largo plazo, por lo que sería nuestra primera recomendación si es que vamos a comenzar a aprender desarrollo para iOS desde cero.

Al escoger la opción Get started with a playground se abre un asistente para indicar los datos de nuestro «espacio de juego». Asignamos un nombre y una plataforma para trabajar.

Luego nos pedirá un lugar en el disco donde guardar los archivos que vamos a producir.

Otra cosa que nos aparecerá es la pantalla para autorizar a Xcode que entre en el Developer mode que permite opciones más avanzadas para desarrollo y depuración. Podemos decirle que sí.

Ahora dentro del playground podremos escribir código en la parte de la izquierda y obtendremos una vista previa en la parte de la derecha. Aparece ya una pequeña muestra de código para poder empezar. Si editamos el valor de la variable podremos ver en la parte de la derecha como también se cambia la salida.



Bibliografía

- [1]. EUROINNOVA. "¿Cuáles son los sistemas operativos móviles más reconocidos?". Todos los Derechos Reservados ©. [En línea]. Disponible en: <https://www.euroinnova.com/programacion-y-desarrollo-de-software/articulos/sistemas-operativos-moviles> [Accedido: 07-Oct-2024].
- [2]. T.C "Desarrolla aplicaciones movil". Todos los Derechos Reservados ©. [En línea]. Disponible: en <https://lariarteagaa98.wixsite.com/misitio/single-post/2016/03/10/sistema-operativo-emergente> [Accedido: 07-Oct-2024].
- [3]. P. S. "Características de Phone". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://ubuntuso1.wordpress.com/ubuntu-movil/caracteristicas-de-phone/> [Accedido: 07-Oct-2024].
- [4]. F. A. "Los mejores entornos para crear aplicaciones Android [sin Java]". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.yeeply.com/blog/desarrollo-de-apps/entornos-programacion-desarrollar-apps-android/> [Accedido: 07-Oct-2024].
- [5]. F. A. "Líneas maestras del desarrollo para Tizen, nuevo SO enfocado a Smartphones". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.logicaalternativa.com/lineas-generales-del-desarrollo-para-tizen-nuevo-so-enfocado-a-smartphones/> [Accedido: 07-Oct-2024].
- [6]. E. B. "Cómo configurar un entorno de desarrollo para iOS". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.logicaalternativa.com/lineas-generales-del-desarrollo-para-tizen-nuevo-so-enfocado-a-smartphones/> [Accedido: 07-Oct-2024].

Conclusión

Podemos concluir que efectivamente se presentó la investigación documental de la Unidad 2, se presentaron los SO emergentes para aplicaciones móviles, las características relevantes de los distintos entornos de desarrollo asociados a los sistemas operativos de la actividad anterior, sus datos más relevantes, adjuntando imágenes, así como también algunas sugerencias de uso de los entornos y lenguajes.

Curso: [Desarrollo de Aplicaciones Móviles](#)

Tarea: [Investigación Unidad 2](#) 

Valor 30

[Ver todos los envíos](#)



Christian Manuel Millán Pólito

211u0383@alumno.itssat.edu.mx

Fecha de entrega: 7 de octubre ...



Cambiar usuario



1 de 2 [Reiniciar preferencias de tabla](#)



◀ Página 1 de 15 ▶



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 11 mins 54 segundos antes de la fecha límite

Los estudiantes pueden editar este envío

 [Inv-U2-MillánPólitoChristianManuel.pdf](#) 7 de octubre de 2024, 23:43

▶ [Comentarios \(0\)](#)

Calificación

Calificación sobre 100



100.00

Calificación actual en el libro

100.00

= 30

Comentarios de retroalimentación

Rich text editor toolbar with icons for undo, font color, bold, italic, bulleted list, numbered list, decrease indent, increase indent, link, unlink, insert image, insert file, insert audio, insert video, insert link, insert table, and a grid icon.

Large empty text area for entering feedback comments.

Notificar a estudiantes [?](#)

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR



ITSSAT

**INSTITUTO TECNOLÓGICO SUPERIOR
DE SAN ANDRÉS TUXTLA**

Carrera: Ing. en Informática

Módulo: Administración y organización de datos

Catedrático: MTI. Rogelio Enrique Telona Torres

Semestre: 4to. Semestre

Grupo: 410A

Alumno:

Christian Manuel Millán Pólito

Periodo Escolar: Febrero - Julio 2022

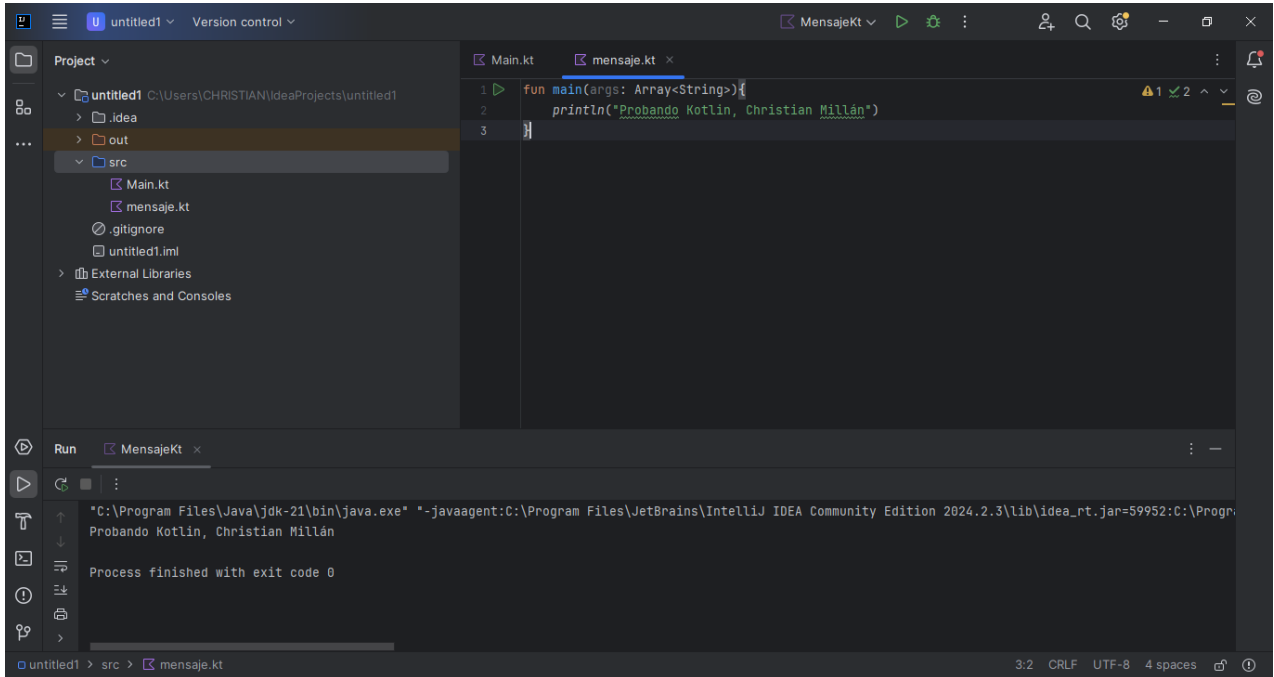
San Andrés Tuxtla, Ver.

Introducción

En este documento se presenta la evidencia de las prácticas donde se realizarán Ejercicios para identificar el funcionamiento de los emuladores. Programando en Kotlin con IntelliJ IDEA para conocer el ambiente de desarrollo y familiarizarse con la sintaxis del lenguaje.

Ejercicios para identificar el funcionamiento de los emuladores

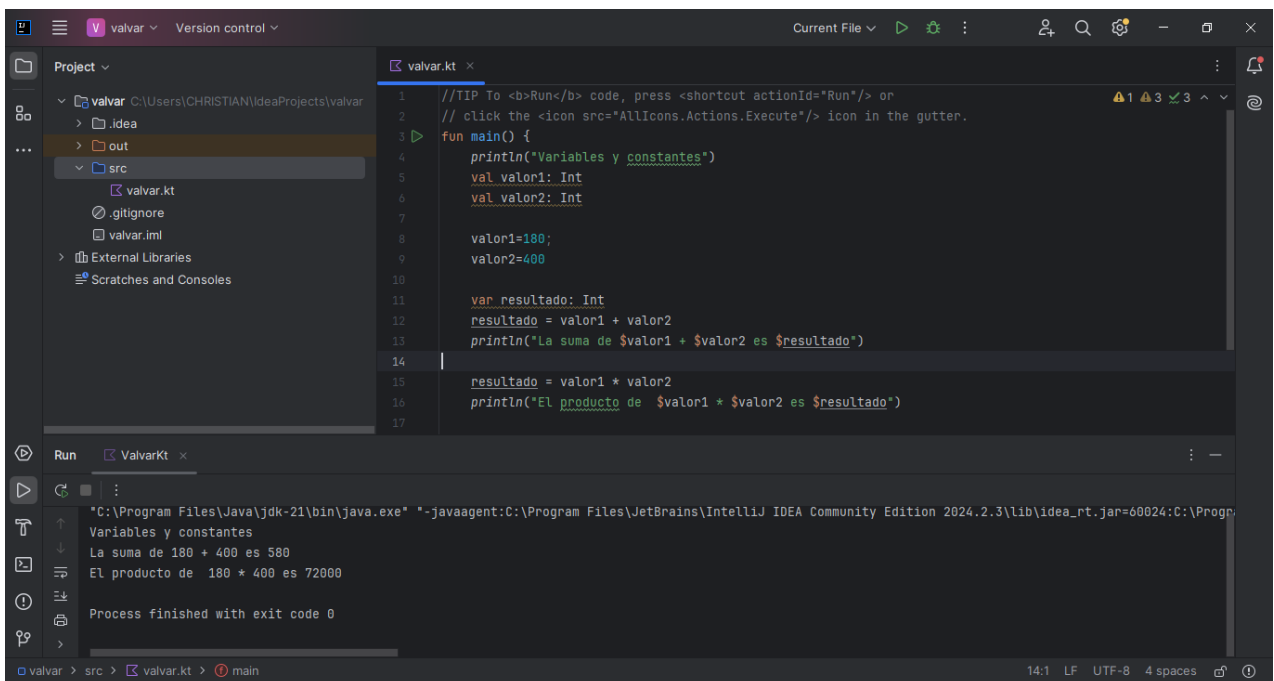
Ejercicio 0



```
1 fun main(args: Array<String>){
2     println("Probando Kotlin, Christian Millán")
3 }
```

```
*C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=59952:C:\Progr
Probando Kotlin, Christian Millán
Process finished with exit code 0
```

Ejercicio 1. Suma de 2 números (Estático)



```
1 //TIP To <b>Run</b> code, press <shortcut actionId="Run"/> or
2 // click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
3 fun main() {
4     println("Variables y constantes")
5     val valor1: Int
6     val valor2: Int
7
8     valor1=180;
9     valor2=400
10
11     var resultado: Int
12     resultado = valor1 + valor2
13     println("La suma de $valor1 + $valor2 es $resultado")
14
15     resultado = valor1 * valor2
16     println("El producto de $valor1 * $valor2 es $resultado")
17 }
```

```
*C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=60024:C:\Progr
Variables y constantes
La suma de 180 + 400 es 580
El producto de 180 * 400 es 72000
Process finished with exit code 0
```

Alumno: Christian Manuel Millán Pólito.

Ejercicio 2. Producto y Suma de 2 números (interactivo)

```
1 fun main(){
2     //Suma y producto
3     print("Ingrese el primer numero: ")
4     val valor1 = readLine()!!.toInt()
5
6     print("Ingrese el primer numero: ")
7     val valor2 = readLine()!!.toInt()
8
9     val suma = valor1 + valor2
10
11     println("La suma de $valor1 + $valor2 = $suma")
12
13     val producto = valor1 * valor2
14     println("El producto de $valor1 * $valor2 = $producto")
15
16
17
```

Run Console Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=60136:C:\Progrn
Ingrese el primer numero: 5
Ingrese el primer numero: 2
La suma de 5 + 2 = 7
El producto de 5 * 2 = 10
Process finished with exit code 0
```

Ejercicio 3. Perímetro de un cuadrado

```
1 fun main(args: Array<String>){
2     print("Ingrese la medida del lado cuadrado: ")
3     val lado = readLine()!!.toInt()
4     val perimetro = lado * 4
5     println("El perimetro del cuadrado es $perimetro")
6
7
```

Run Console Output:

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=49321:C:\Progrn
Ingrese la medida del lado cuadrado: 4
El perimetro del cuadrado es 16
Process finished with exit code 0
```

Ejercicio 4. Compra

```
1 fun main() {  
2     print("Ingrese el precio del producto: ")  
3     val precio = readLine()!!.toDouble()  
4     print("Ingrese la cantidad de productos: ")  
5     val cantidad = readLine()!!.toInt()  
6     val total = precio * cantidad  
7  
8     println("El total a pagar es $total")  
9 }
```

Run Comprakt x

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=49387:C:\Progr  
Ingrese el precio del producto: 12.84  
Ingrese la cantidad de productos: 6  
El total a pagar es 77.03999999999999  
Process finished with exit code 0
```

Ejercicio 5. Cálculo de la suma de dos números y el producto de otros dos

```
1 fun main() {  
11     print("Ingrese el cuarto numero: ")  
12     val valor4 = readLine()!!.toInt()  
13  
14     val suma = valor1 + valor2  
15     println("La suma de $valor1 + $valor2 es $suma")  
16  
17     val producto = valor3 * valor4  
18     println("El producto de $valor3 * $valor4 es $producto")  
19  
20 }
```

Run SumaProductoKt x

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=49478:C:\Progr  
Ingrese el primer numero: 6  
Ingrese el segundo numero: 44  
Ingrese el tercero numero: 3  
Ingrese el cuarto numero: 21  
La suma de 6 + 44 es 50  
El producto de 3 * 21 es 63  
Process finished with exit code 0
```

Conclusión

Podemos concluir que efectivamente en este documento se presentó la evidencia de las prácticas con 6 pruebas de ejecución de los Ejercicios para identificar el funcionamiento de los emuladores conociendo el ambiente de desarrollo de IntelliJ IDEA programando en Kotlin, que tiene una sintaxis sencilla.

Curso: [Desarrollo de Aplicaciones Móviles](#)

Tarea: **Prácticas Unidad 2** ⚙️

Valor 30

[Ver todos los envíos](#)



Christian Manuel Millán Pólito

211u0383@alumno.itssat.edu.mx

Fecha de entrega: 31 de octubre ...



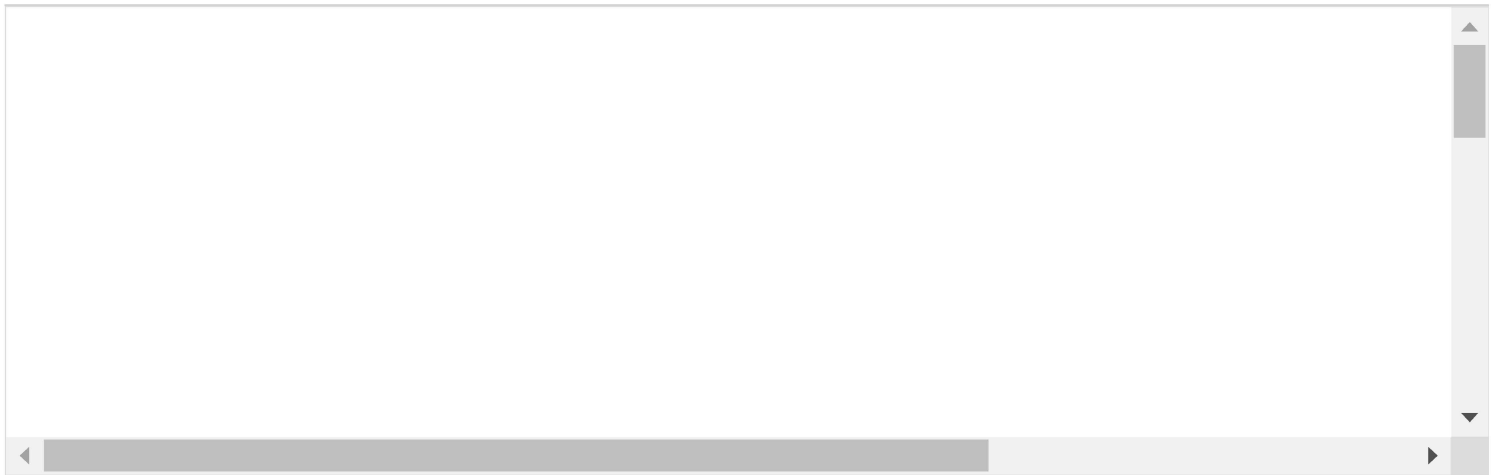
Cambiar usuario



1 de 2 [Reiniciar preferencias de tabla](#)



◀️ Página 1 de 6 ▶️



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 21 horas 24 mins antes de la fecha límite

Los estudiantes pueden editar este envío

[Prácticas-U2-MillánPólitoChristianManuel.pdf](#) 31 de octubre de 2024, 02:30

▶ [Comentarios \(0\)](#)

Calificación

Calificación sobre 100



100.00

Calificación actual en el libro

100.00 = 30

Comentarios de retroalimentación

Rich text editor toolbar with icons for undo, font color, bold, italic, bulleted list, numbered list, decrease indent, increase indent, link, unlink, insert image, insert file, insert audio, insert video, insert link, insert table, and a grid icon.

Large empty text area for providing feedback comments.

Notificar a estudiantes [?](#)

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR



ITSSAT

**INSTITUTO TECNOLÓGICO SUPERIOR
DE SAN ANDRÉS TUXTLA**

Carrera: Ing. en Informática

Módulo: Administración y organización de datos

Catedrático: MTI. Rogelio Enrique Telona Torres

Semestre: 4to. Semestre

Grupo: 410A

Alumno:

Christian Manuel Millán Pólito

Periodo Escolar: Febrero - Julio 2022

San Andrés Tuxtla, Ver.

Índice

Introducción	3
Sistemas Operativos para Dispositivos Móviles	4
Bibliografía	5
Conclusión.....	6

Introducción

En este documento se presenta el cuadro comparativo de Sistemas Operativos para Dispositivos Móviles que destaca los datos y características que diferencian entre sí a cada uno de los sistemas operativos.

Sistemas Operativos para Dispositivos Móviles

Sistema Operativo	Desarrollado por	Año de lanzamiento	Licencia	Aplicaciones	Compatibilidad de apps	Seguridad	Actualizaciones	Popularidad actual	Compatibilidad de hardware
Android	Google	2008	Open source (Apache) con componentes cerrados	Google Play Store	Apps de Android	Alta, Google Play Protect	Irregulares, depende del fabricante	Alta	Variada, disponible en múltiples fabricantes (Marcas)
iOS	Apple	2007	Cerrado, propietario	Apple App Store	Exclusivas de iOS	Alta, actualizaciones rápidas	Frecuentes y directas de Apple	Alta en mercados desarrollados	Exclusiva de dispositivos Apple
BlackBerry OS	BlackBerry Ltd.	1999	Cerrado	BlackBerry World	Aplicaciones propias de BlackBerry	Alta, enfocada en seguridad	Infrecuentes	Muy baja, descontinuado	Exclusiva de dispositivos BlackBerry
Tizen	Samsung, Linux Foundation	2012	Open source (Apache 2.0)	Tizen Store	Apps Tizen (compatibles algunas de Android)	Media, con enfoque en IoT	Irregulares, depende del fabricante	Baja, principalmente en dispositivos Samsung	Principalmente dispositivos Samsung
Phone Ubuntu	Canonical	2013	Open source	Ubuntu Store	Apps propias de Ubuntu Touch	Alta, acceso a software libre	Irregulares	Muy baja, en nichos específicos	Limitada a dispositivos con soporte de Ubuntu

Bibliografía

- [1]. EUROINNOVA. "¿Cuáles son los sistemas operativos móviles más reconocidos?". Todos los Derechos Reservados ©. [En línea]. Disponible en: <https://www.euroinnova.com/programacion-y-desarrollo-de-software/articulos/sistemas-operativos-moviles> [Accedido: 07-Oct-2024].
- [2]. T.C "Desarrolla aplicaciones movil". Todos los Derechos Reservados ©. [En línea]. Disponible: en <https://lariarteagaa98.wixsite.com/misitio/single-post/2016/03/10/sistema-operativo-emergente> [Accedido: 07-Oct-2024].
- [3]. P. S. "Características de Phone". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://ubuntuso1.wordpress.com/ubuntu-movil/caracteristicas-de-phone/> [Accedido: 07-Oct-2024].
- [4]. F. A. "Los mejores entornos para crear aplicaciones Android [sin Java]". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.yeeply.com/blog/desarrollo-de-apps/entornos-programacion-desarrollar-apps-android/> [Accedido: 07-Oct-2024].
- [5]. F. A. "Líneas maestras del desarrollo para Tizen, nuevo SO enfocado a Smartphones". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.logicaalternativa.com/lineas-generales-del-desarrollo-para-tizen-nuevo-so-enfocado-a-smartphones/> [Accedido: 07-Oct-2024].
- [6]. E. B. "Cómo configurar un entorno de desarrollo para iOS". Todos los Derechos Reservados© 2018. [En línea]. Disponible en: <https://www.logicaalternativa.com/lineas-generales-del-desarrollo-para-tizen-nuevo-so-enfocado-a-smartphones/> [Accedido: 07-Oct-2024].

Conclusión

Podemos concluir que efectivamente se presentó el cuadro comparativo de Sistemas Operativos para Dispositivos Móviles que destaca los datos y características que diferencian entre sí a cada uno de los sistemas operativos, entre ellos Android, iOS, BlackBerry OS, Tizen y Phone Ubuntu .

Curso: [Desarrollo de Aplicaciones Móviles](#)

Tarea: [Examen unidad 2](#) 

Valor 40

[Ver todos los envíos](#)



Christian Manuel Millán Pólito

211u0383@alumno.itssat.edu.mx

Fecha de entrega: 1 de noviembre ...



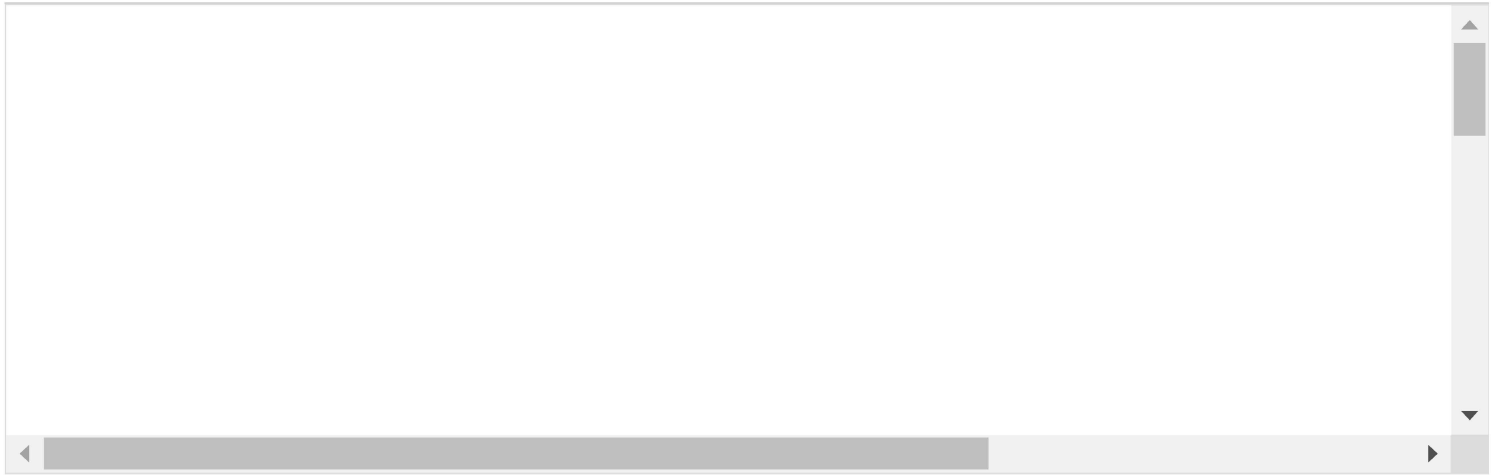
Cambiar usuario



1 de 2 [Reiniciar preferencias de tabla](#)



◀ Página 1 de 6 ▶



Entrega

Enviado para calificar

Calificado

La tarea fue enviada 22 horas 52 mins antes de la fecha límite

Los estudiantes pueden editar este envío

 [Examen-U2-MillánPólitoChristianManuel.pdf](#) 1 de noviembre de 2024, 01:02

▶ [Comentarios \(0\)](#)

Calificación

Calificación sobre 100



100.00

Calificación actual en el libro

100.00 = 40

Comentarios de retroalimentación

Rich text editor toolbar with icons for undo, bold, italic, bulleted list, numbered list, decrease indent, increase indent, link, unlink, insert image, insert file, insert audio, insert video, insert link, insert table, and a grid icon.

Notificar a estudiantes [?](#)

GUARDAR CAMBIOS

GUARDAR Y MOSTRAR SIGUIENTE

REINICIAR