**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | ADSO |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501096. Desarrollar la solución de *software* de acuerdo con el diseño y metodologías de desarrollo. | RESULTADOS DE APRENDIZAJE | Crear componentes *front-end* del *software* de acuerdo con el  diseño. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | CF015 |
| NOMBRE DEL COMPONENTE FORMATIVO | Arquitectura aplicaciones móviles nativas |
| BREVE DESCRIPCIÓN | En este componente formativo se abordan los conceptos básicos de las aplicaciones móviles nativas, los cuales servirán para identificar los tipos de metodologías usadas en el desarrollo móvil, en plataformas nativas, en entornos de desarrollo y su configuración. |
| PALABRAS CLAVE | Android, metodologías ágiles, Android Studio, refactorización, Scrum. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**
2. Enfoque metodológico para el desarrollo de aplicaciones móviles nativas
3. Plataformas de desarrollo móvil nativas
4. Entornos de desarrollo
   1. Plataforma Android
   2. Android Studio
   3. Arquitectura de una aplicación en Android
5. Instalación y configuración entorno Android Studio
6. Componentes de una aplicación Android
7. **INTRODUCCIÓN**

Sin duda, los dispositivos móviles son parte de nuestro día a día y cada vez son más avanzados. La creciente demanda de *software* específico para estos dispositivos ha generado nuevos desafíos para los desarrolladores, ya que este tipo de aplicaciones posee características propias, restricciones y necesidades únicas.

|  |  |
| --- | --- |
| A hand holding a cell phone  Description automatically generated | Es importante definir cuáles serán las tecnologías más adecuadas para la programación de móviles, y una de las arquitecturas más implantadas es la proporcionada por el sistema Android. En los dispositivos móviles actuales pueden utilizarse tres tipos de aplicaciones diferentes: **nativas, híbridas y web.** En este componente se tratarán las aplicaciones móviles nativas. |

Las aplicaciones nativas son aquellas que enfocan su desarrollo en una plataforma específica como **Android, iOS y Windows**, haciendo uso de su correspondiente lenguaje de programación, el cual permite tener compatibilidad con sus características. En el caso de Android, se utilizan lenguajes como Java y Kotlin; para la plataforma iOS, Objective-C o Swift; y para la plataforma Windows, los lenguajes C++ o C#.

Desde el punto de vista de los programadores, uno de los mayores inconvenientes es que para que una misma aplicación pueda utilizarse en dos o más plataformas distintas, hay que desarrollar la misma aplicación en los diferentes lenguajes, iniciando desde cero y necesitando más tiempo de desarrollo y mayor costo.

1. **DESARROLLO DE CONTENIDOS:**
2. **Enfoque metodológico para el desarrollo de aplicaciones móviles nativas**

Actualmente, las metodologías ágiles tienen una gran notoriedad, ya que ofrecen una buena solución para desarrollar proyectos en tiempos más cortos, especialmente aquellos proyectos que constantemente experimentan cambios en sus requisitos. Las aplicaciones para dispositivos móviles son un buen ejemplo de esto, ya que requieren una variedad de características como **portabilidad, movilidad y adaptación,** entre otras. Aunque hay muchas aplicaciones para dispositivos móviles que funcionan en diferentes plataformas, no siempre cumplen totalmente las expectativas de los usuarios debido a su baja calidad en el desarrollo. Esto se debe a que el uso de metodologías no es una prioridad en este tipo de aplicaciones, ignorando los métodos de ingeniería que son los que garantizan el mantenimiento y calidad de las aplicaciones.

|  |  |
| --- | --- |
|  | Según con Letelier y Canos (2003)  Las metodologías ágiles son procesos para desarrollar *software* de manera rápida y con gran facilidad de adopción por los equipos de trabajo. Se han seleccionado las metodologías ágiles más referenciadas para emplearlas en el desarrollo de aplicaciones móviles, lo cual resulta muy conveniente teniendo en cuenta el crecimiento de los proyectos para estas tecnologías. |

A continuación, se describen los conceptos principales para la introducción al uso de metodologías de desarrollo:

Entre las principales metodologías ágiles aplicadas en el desarrollo de aplicaciones móviles, se destacan las siguientes:

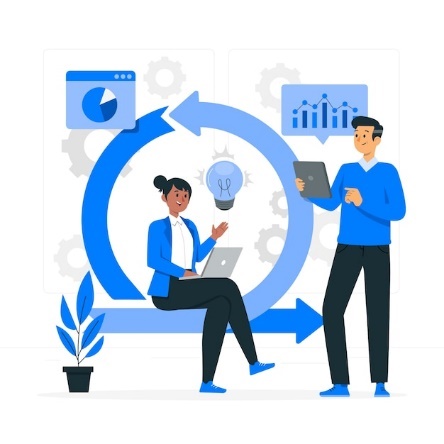
**Extreme Programming (XP)**

Es una metodología liviana, aplicable en grupos de desarrollo pequeños y medianos que se centra en requerimientos no exactos y en continuo cambio. Se caracteriza por tener un ciclo de vida dinámico, a través de ciclos de desarrollo cortos denominados iteraciones. Sus fases son análisis, diseño, desarrollo y pruebas, en las cuales se aplican 12 prácticas.

**Figura 1.** Prácticas de XP



Así pues, XP no es solo un conjunto de técnicas, sino un conjunto de principios probados y fiables con características de las metodologías convencionales, pero llevadas a un nivel extremo.

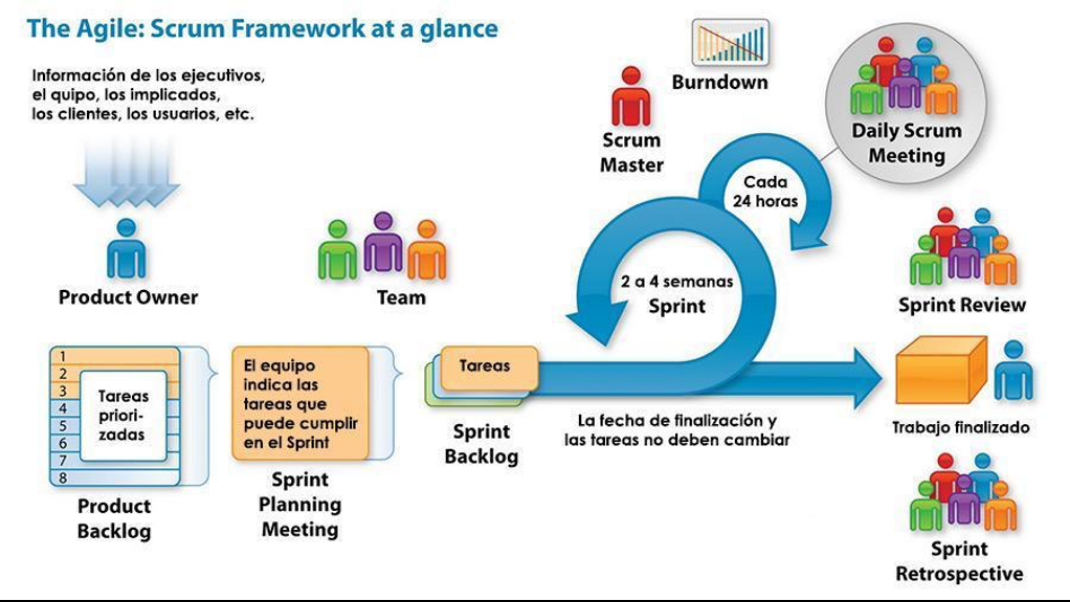
***SCRUM***

*Scrum,* según Takeuchi & Nonaka (1986), presenta un proceso adaptativo, rápido y auto-organizado para el desarrollo de productos, centrado en la gestión de proyectos en situaciones que son difíciles de planificar a futuro. El término "Scrum" proviene del famoso juego de rugby y hace referencia a la forma en que se reintroduce un balón que ha salido del campo al terreno de juego de manera colectiva.

Este modelo se basa en el desarrollo incremental, es decir, conforme avanzan las fases y la iteración, mayor es el tamaño del proyecto que se está desarrollando. Los procesos que utiliza son:

Consiste en realizar un análisis de los requerimientos del sistema (*Product Backlog*), señalar cuáles serán los objetivos a corto o mediano plazo dentro de un *sprint*, es decir, en la fase de desarrollo. Posteriormente, los desarrolladores realizan su trabajo, se efectúan algunas pruebas y se retroalimenta de acuerdo con lo conseguido al terminar la última fase**.**

**Figura 2**. Modelo ágil *Scrum*



A continuación, se explora el enfoque iterativo e incremental que Scrum ofrece para la gestión de proyectos de *software*:

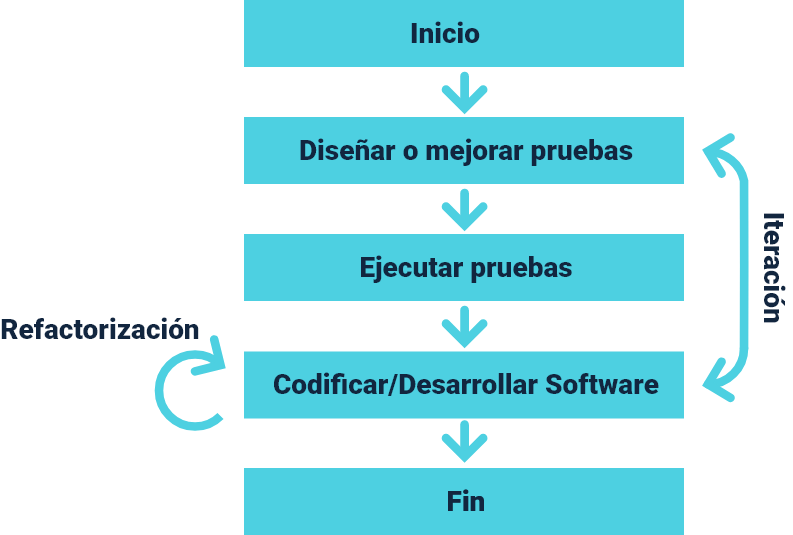


<https://youtu.be/DSO1YzSAx4s?feature=shared>

***Test Driven Development (TDD)***

Según Astel (2003), el *Test Driven Development* es un enfoque de desarrollo que mantiene un exhaustivo conjunto de pruebas realizadas por el programador. Su objetivo principal es que ninguna parte del código avance a producción sin antes realizar las pruebas asociadas y obtener la aprobación correspondiente. Se trata de un desarrollo orientado a las pruebas, que cambia la mentalidad del equipo de desarrollo, agilizando los resultados y aumentando la calidad del producto. La tendencia actual es integrar TDD en cualquier metodología, ya sea ágil (Scrum Alliance - TDD and Scrum, 2011) o tradicional (Letelier et al., s. f.), para aprovechar los beneficios de practicar una metodología que permite corregir errores constantemente, asegurar la calidad del producto y protegerse de errores, tanto malintencionados como humanos.

**Figura 3.** *Test Driven (TDD)*



**Mobile-D.**

Según Abrahamsson (2004) es una metodología de desarrollo diseñada específicamente para el desarrollo de aplicaciones móviles, que integra prácticas ágiles como *Extreme Programming y Crystal*. Presenta características tales como la orientación a pruebas, la programación en parejas, la integración continua y la refactorización, además de las tareas de mejora del proceso de *software.* La metodología incluye las fases de exploración, inicialización, producción, estabilización y pruebas. Cada fase cuenta con un día de planificación y un día de entrega.

**Figura 4.** Ciclo de desarrollo *Mobile-D*

A chart with yellow arrows and black background

Description automatically generated

A continuación, se explica cada fase:

|  |
| --- |
| Pestañas  CF015\_\_1\_ Enfoque metodológico para el desarrollo de aplicaciones móviles nativas |

1. **Plataformas de desarrollo móvil nativas**

Las plataformas de desarrollo móvil nativas, son aquellas que tienen un desarrollo específico para cada uno de sus sistemas operativos; *IOS, Android, o Windows Phone*, lo que quiere decir que para cada uno de estos Sistemas Operativos se adapta un lenguaje de desarrollo así:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Sistema Operativo Móvil: iOS, Lenguajes: Objetive-C, Swift | Sistema Operativo Móvil:  Android, Lenguajes: Java, Kotlin | Sistema Operativo Móvil:  Windows Phone  Lenguajes: C++, C# |

Una de las principales ventajas de las aplicaciones nativas es que pueden aprovechar todas las funcionalidades avanzadas de los dispositivos de cada plataforma, incluidos los procesadores gráficos. Los mayores beneficios de este tipo de aplicaciones siempre serán la flexibilidad y la usabilidad, que son superiores a las de las aplicaciones híbridas. Uno de los principales inconvenientes del desarrollo de las aplicaciones nativas es que tanto el desarrollo como las actualizaciones de estas apps tienen un alto costo.

**Rendimiento**

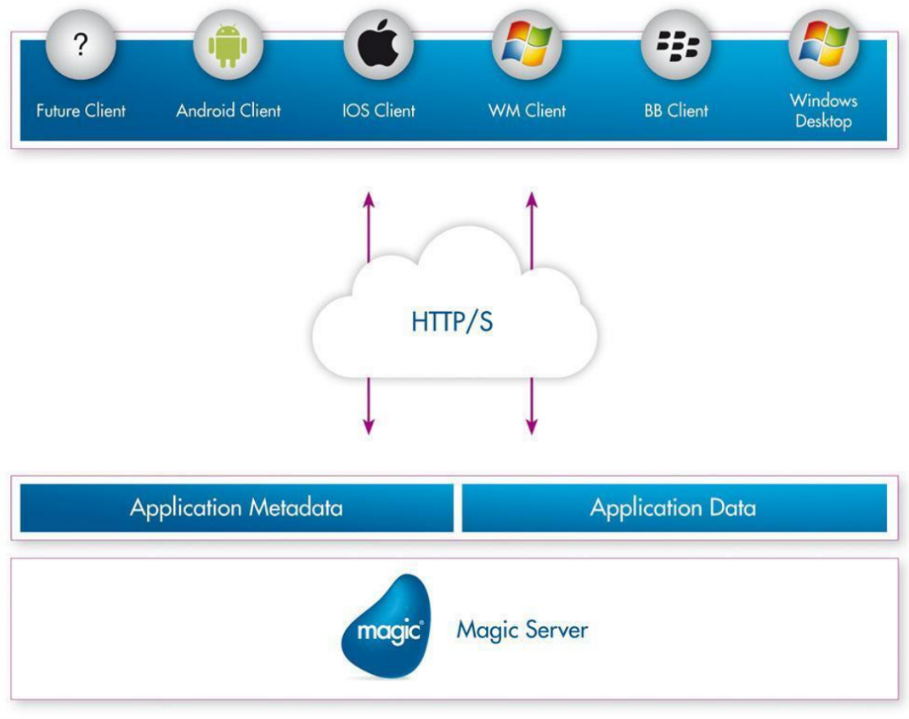
El rendimiento de las aplicaciones nativas es otra de sus características principales, ya que ofrecen mayor velocidad, animaciones avanzadas, transiciones complejas, aprovechamiento total del hardware (GPS, cámara, sensores, etc.) y un menor consumo de memoria.

Ventajas de las aplicaciones nativas:

**Arquitectura de las aplicaciones móviles**

Básicamente, el desarrollo de las *apps* móviles se compone de dos partes, las cuales se diferencian por su código: el *front-end* y la parte de *web services.*

**Figura 5.** Arquitectura de una *app*



1. **Entornos de desarrollo**

Un entorno de desarrollo es un *software* que facilita un conjunto de herramientas que permiten crear una aplicación. Estos entornos contienen soporte para uno o varios lenguajes de programación y son los encargados de depurar, compilar, realizar pruebas y convertir nuestro código en una aplicación ejecutable.

Los entornos más utilizados para el desarrollo de aplicaciones nativas son: Xcode para la plataforma iOS y Android Studio para la plataforma Android.

|  |
| --- |
| Tarjetas  CF015\_3\_Entornos de desarrollo |

El desarrollo de aplicaciones móviles se ha convertido en una necesidad debido a la gran utilización y evolución de estas plataformas. Por esta razón, es importante definir cuáles serán las tecnologías más adecuadas para la programación móvil. Una de las arquitecturas más implantadas es la proporcionada por el sistema Android.

**3.1. Plataforma Android**

Es un sistema operativo móvil desarrollado por Google, de código abierto y basado en Linux. Ofrece una optimización de la máquina virtual denominada Dalvik VM y soporta la mayoría de las API de Java SE para el desarrollo de aplicaciones. Está enfocado en ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos.

Sus características son:

**Versiones**

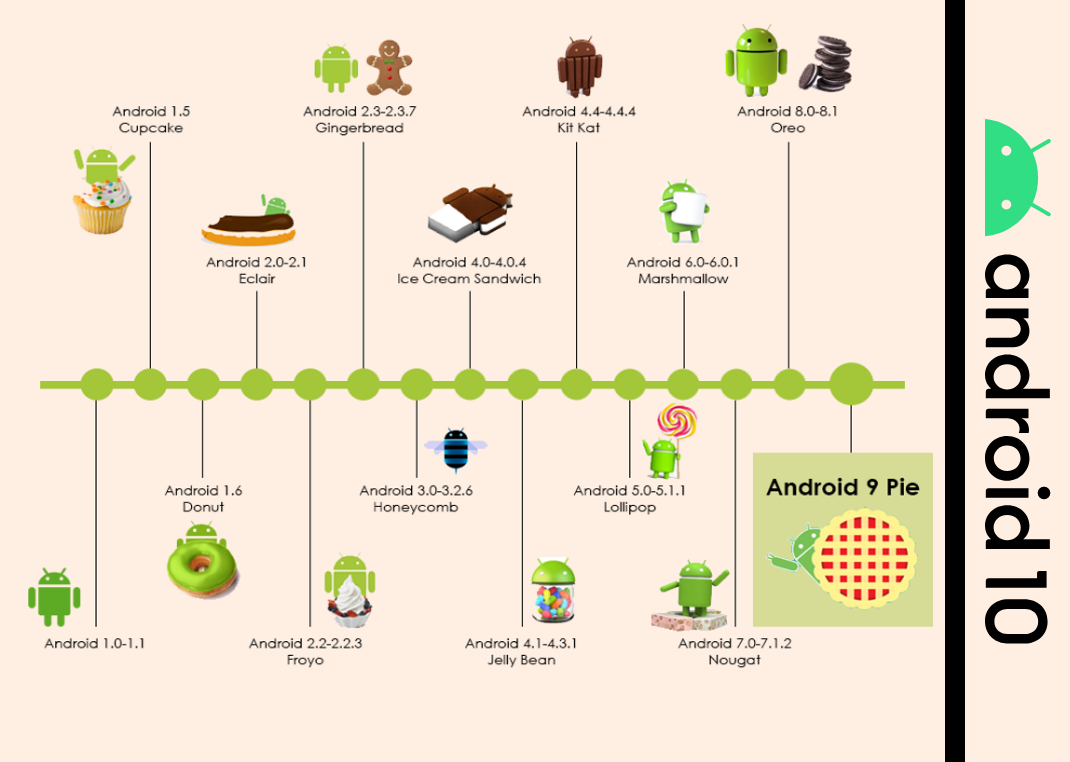
Android ha lanzado muchas versiones desde su inicio, con numerosas actualizaciones. Cada nueva versión corrige fallos detectados en las anteriores e incluye nuevas funcionalidades con soporte para las nuevas tecnologías. Curiosamente, las versiones de Android reciben el nombre de postres o dulces, cuyos nombres van en orden alfabético, una tradición que se mantuvo hasta la versión 9.

**Tabla 1.**  Nombres versiones y lanzamiento de sistema operativo Android

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Versión** | **Lanzamiento** |
| Android Apple Pie | Versión 1.0 | 23 de septiembre de 2008 |
| Android Banana Bread | Versión 1.1 | 9 de febrero de 2009 |
| Android Cupcake | Versión 1.5 | 25 de abril de 2009 |
| Android Donut | Versión 1.6 | 5 de septiembre de 2009 |
| Android Éclair | Versión 2.0 | 26 de octubre de 2009 |
| Android Froyo | Versión 2.2 | 20 de mayo de 2010 |
| Android Gingerbread | Versión 2.3 | 6 de diciembre de 2010 |
| Android Honeycomb | Versión 3.0 | 22 de febrero de 2011 |
| Android Ice Cream Sandwich | Versión 4.0 | 18 de octubre de 2011 |
| Android Jelly Bean | Versión 4.1 | 9 de julio de 2012 |
| Android KitKat | Versión 4.4 | 31 de octubre de 2012 |
| Android Lollipop | Versión 5.0 | 12 de noviembre de 2014 |
| Android Marshmallow | Versión 6.0 | 5 de octubre de 2015 |
| Android Nougat | Versión 7.0 | 15 de junio de 2016 |
| Android Oreo | Versión 8.0 | 21 de agosto de 2017 |
| Android Pie | Versión 9.0 | 6 de agosto de 2018 |
| Android 10 | Versión 10 | 3 de septiembre de 2019 |
| Android 11 | Versión 11 | 8 de septiembre de 2020 |
| Android 12 | Versión 12 | 4 de octubre de 2021 |
| Android 12 Beta 2 | Versión 12 | 9 de junio de 2021 |
| Android 13 | Versión 13 | 15 de agosto de 2022 |

A continuación, se presenta un esquema que resume lo anterior:

**Figura 6.** Versiones de Android



*Nota.* Tomado de El origen de Android (Características y Costos, por F. Olmos, 2020

**Arquitectura**

Los componentes principales de la plataforma Android están diseñados para una variedad de dispositivos. Android es de código abierto y está basado en Linux.

|  |
| --- |
| Infografía interactiva  CF015\_3.1.\_Plataforma Android |

A continuación, se presenta una descripción completa del IDE Android Studio, que como se mencionó anteriormente, es el entorno oficial para el desarrollo de aplicaciones móviles nativas en Android.

* 1. **Android Studio**

Android Studio está basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de *software* libre Apache 2.0, está programado en Java y es multiplataforma. Fue presentado por Google el 16 de mayo de 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado exclusivamente a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Actualmente, es el IDE recomendado por Google.

Las características son:

La realización de aplicaciones en Android Studio se enfoca en 4 fases de desarrollo:

|  |
| --- |
| Infografía  CF015\_3.2\_Android Studio |

**3.3 Arquitectura de una aplicación en Android**

Cada componente depende solo del componente que está un nivel más abajo. Por ejemplo, las actividades y los fragmentos solo dependen de un modelo de vista. El repositorio es la única clase que depende de otras clases. En este ejemplo, el repositorio depende de un modelo de datos persistente y de una fuente de datos de *back-end* remota.

Este diseño crea una experiencia de usuario consistente y agradable. Independientemente de que el usuario vuelva a la *app* varios minutos después de cerrarla por última vez o varios días más tarde, verá instantáneamente la información del usuario que la *app* persiste a nivel local. Si estos datos están inactivos, el módulo de repositorio comienza a actualizar los datos en segundo plano.

**Figura 7**. Arquitectura de una aplicación en Android

A diagram of a computer

Description automatically generated

**Versiones**

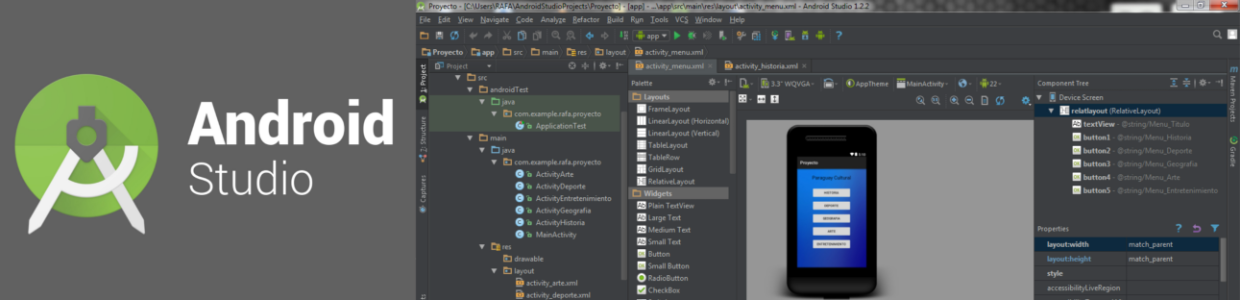
Así como el sistema operativo Android tiene una gran variedad de actualizaciones, el ritmo de actualizaciones de Android Studio es bastante alto. A continuación, se listan las últimas versiones desde la más actual hasta la más antigua.

**Tabla 2.** Nombres versiones y lanzamiento de sistema operativo Android

|  |  |
| --- | --- |
| **Versiones Android Studio** | **Fecha de lanzamiento** |
| Android Studio Bumblebee (2021.1.1) | 27 de junio de 2021 |
| Android Studio Arctic Fox (2020.3.1) Beta 3 | 27 de mayo de 2021 |
| Android Studio 4.2.1 | 13 de mayo de 2021 |
| Android Studio 4.2.0 | 4 de mayo de 2021 |
| Android Studio 4.1.3 | 18 de marzo de 2021 |
| Android Studio 4.1 | 12 de octubre de 2020 |
| Android Studio 4.0 | 28 de mayo de 2020 |
| Android Studio 3.6.3 | 17 de abril de 2020 |
| Android Studio 3.0 | 25 de octubre de 2017 |
| Android Studio 2.3.2 | 11 de mayo de 2017 |
| Android Studio Chipmunk (2021.2.1) | 9 de mayo de 2022 |
| Android Studio Dolphin (2021.3.1) | 8 de agosto de 2022 |
| Android Studio Electric Eel (2022.1.1) | 27 de marzo de 2023 |

1. **Instalación y configuración entorno Android Studio**

Para la instalación, lo primero que se debe tener en cuenta son los requisitos mínimos que debe cumplir el equipo. Para un mejor rendimiento y para asegurar que el emulador funcione correctamente, se necesita un equipo de trabajo adecuado que cuente con las propiedades recomendadas para cada sistema operativo.



Requerimientos de sistema para instalación de Android Studio:

Además de estos requisitos del sistema, para una instalación exitosa y un funcionamiento correcto del *software*, se deben considerar otros elementos. Dependiendo del sistema operativo que se utilice, ya sea Ubuntu, Linux, Windows 10 o Mac OS, es necesario seguir unas instrucciones específicas para cada uno de ellos. Se puede encontrar información al respecto en el enlace proporcionado.

|  |  |
| --- | --- |
|  | **Instalación**  La instalación de Android en este componente se realizará para Windows, para más información de cómo instalar en Mac, Linux, o Chrome ingresa al enlace:  <https://developer.android.com/studio/install?hl=es-419> |

Después de verificar los requisitos del sistema se deben seguir los siguientes pasos para la instalación.

1. **Descargar e instalar Android Studio.** Ingresa a la página oficial *Android developers* y selecciona la opción Android Studio o ingresa al siguiente enlace: <https://developer.android.com/studio?hl=es-419>

**Figura 8.** Descarga de Android Studio

A screenshot of a computer program

Description automatically generated

1. La página detecta el sistema operativo y descarga el instalador correspondiente, seleccionar la opción de: Leí y acepto los términos y condiciones anteriores; y hacer clic en descargar Android Studio.

**Figura 9.** Términos y condiciones de la instalación

A screenshot of a computer

Description automatically generated

1. Para instalar la aplicación, se debe descargar y ejecutar el instalador. Al iniciar, se sigue el asistente de instalación.

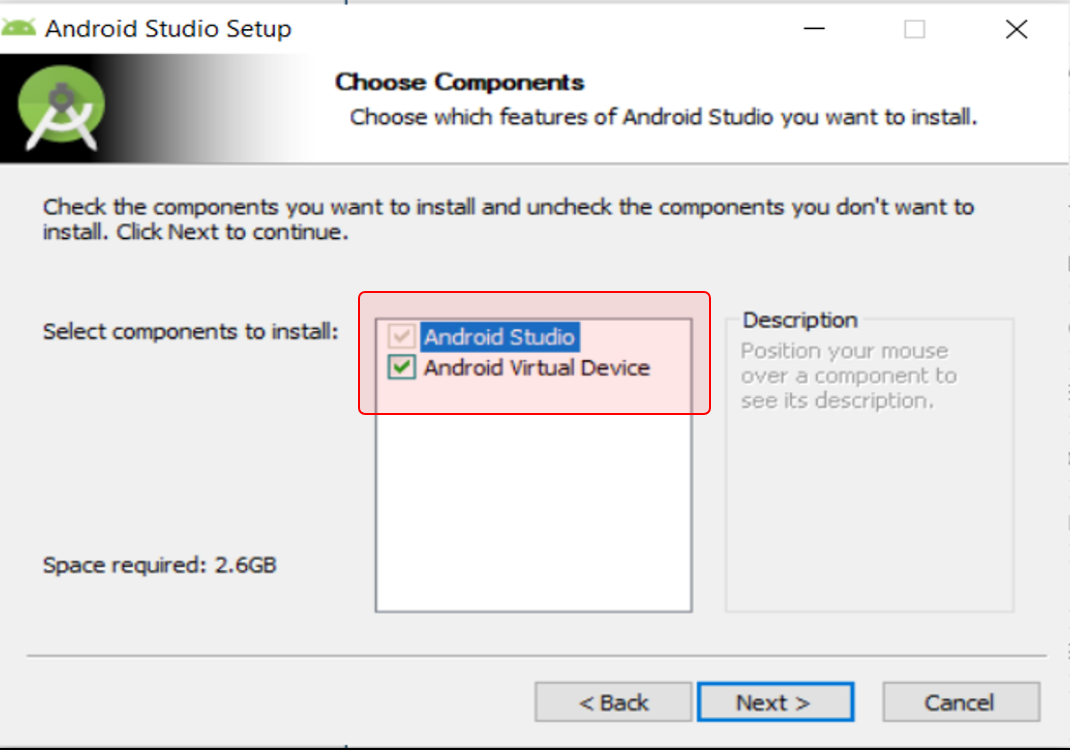
**Figura 10.** Asistente de Instalación Android Studio

A screenshot of a computer

Description automatically generated

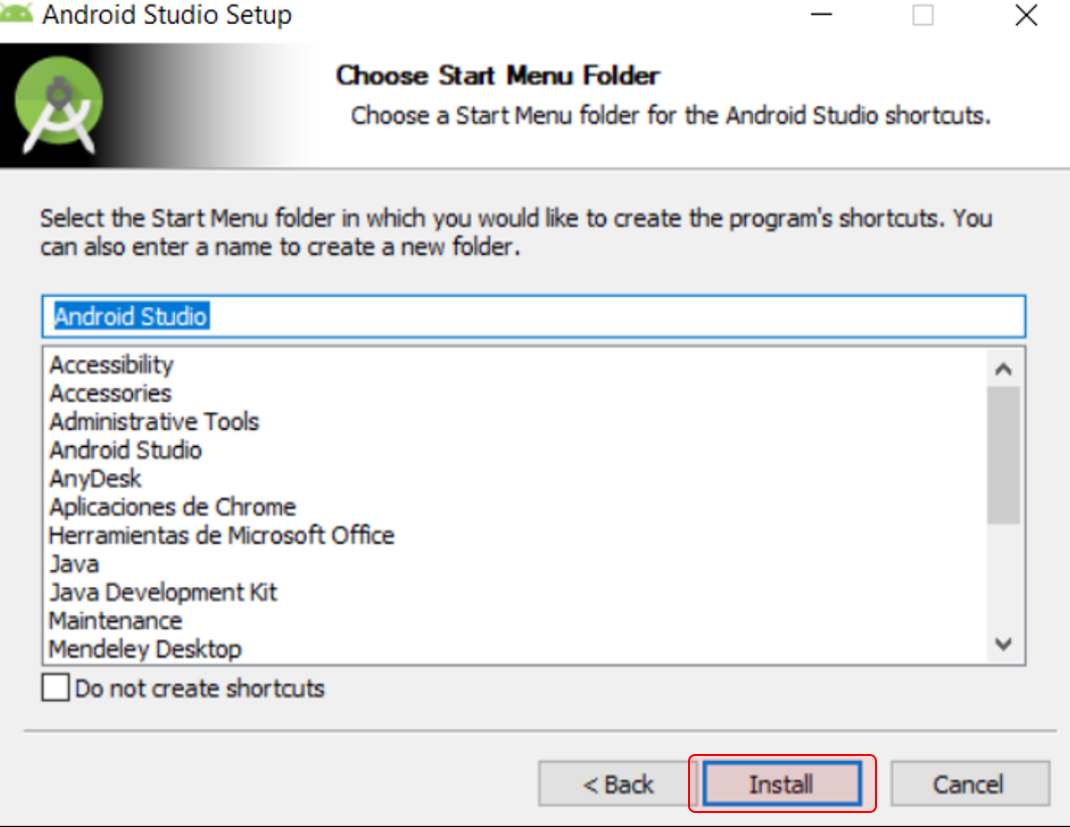
1. En este paso se muestra la ruta donde quedará instalado Android Studio, hacer clic en *Next.*

**Figura 11.** Asistente Instalación Selección componentes



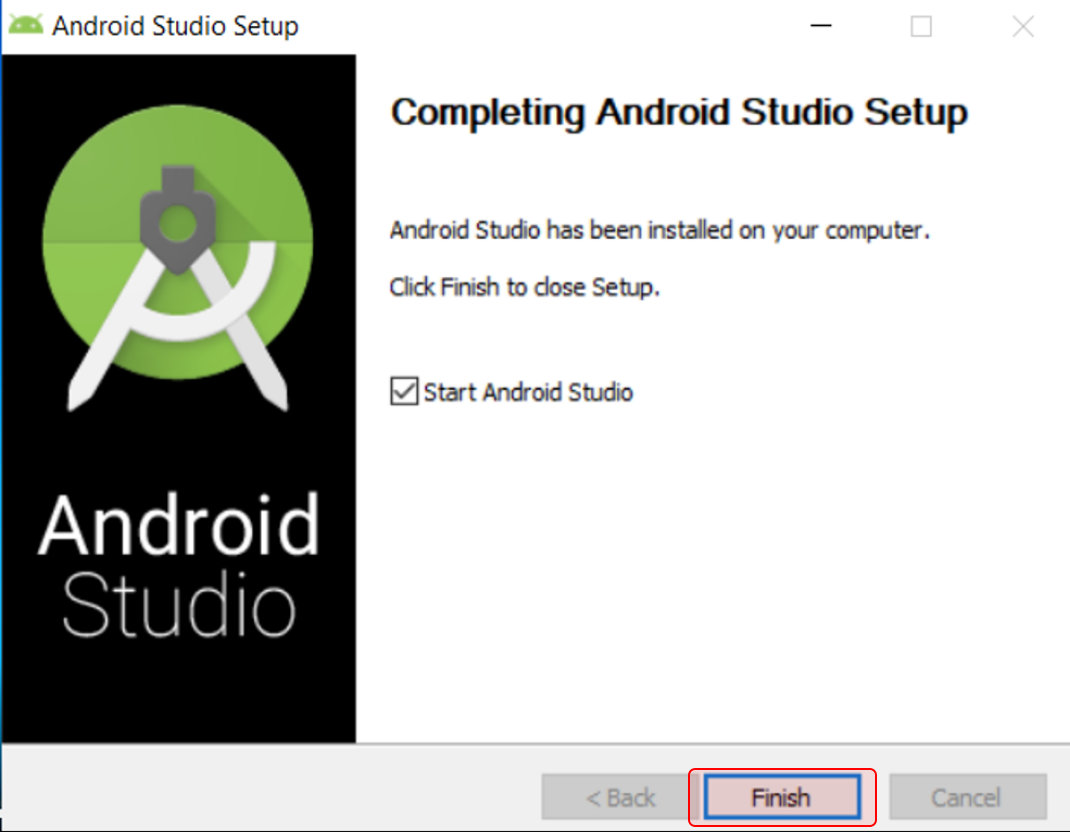
1. En este paso del asistente, se deben seleccionar los componentes a instalar. Se recomienda dejar marcados, en este caso, dos: el propio IDE Android Studio y un *Virtual Device.*

**Figura 12.** Asistente de Instalación



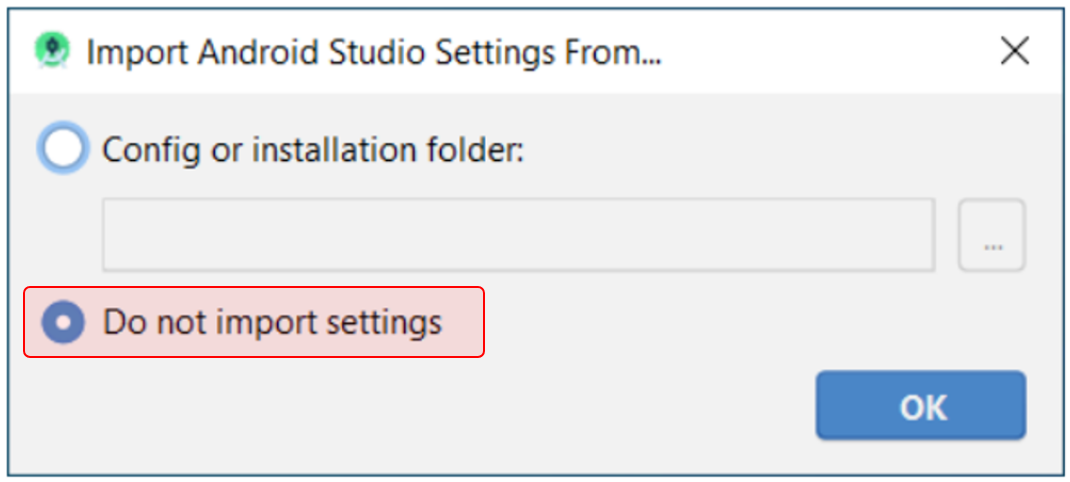
Para el resto de pasos de este asistente inicial, se deben aceptar sin modificar ninguna opción por defecto, hasta llegar al último paso donde se selecciona la opción *"Start Android Studio"* y se pulsa el botón *"Finish*". De esta manera, la aplicación se iniciará automáticamente.

**Figura 13.** Asistente de Instalación Iniciar Android Studio



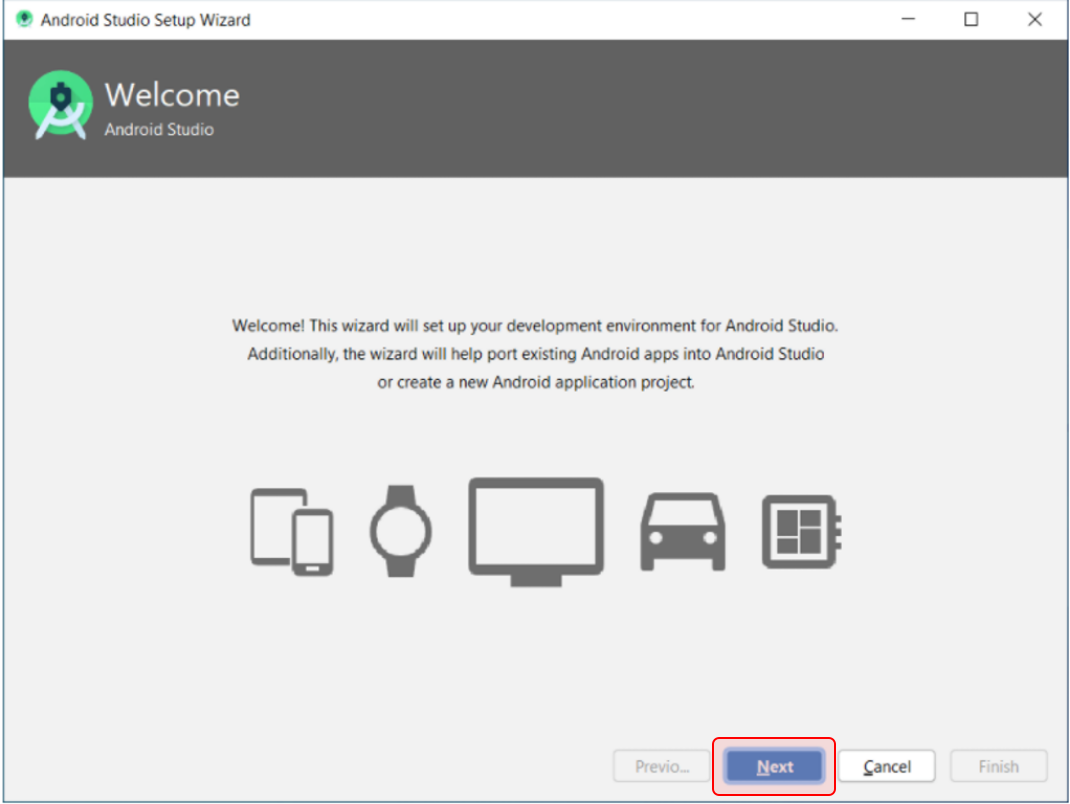
1. Seleccionar: no importar configuraciones

**Figura 14.** Asistente de Instalación Importar



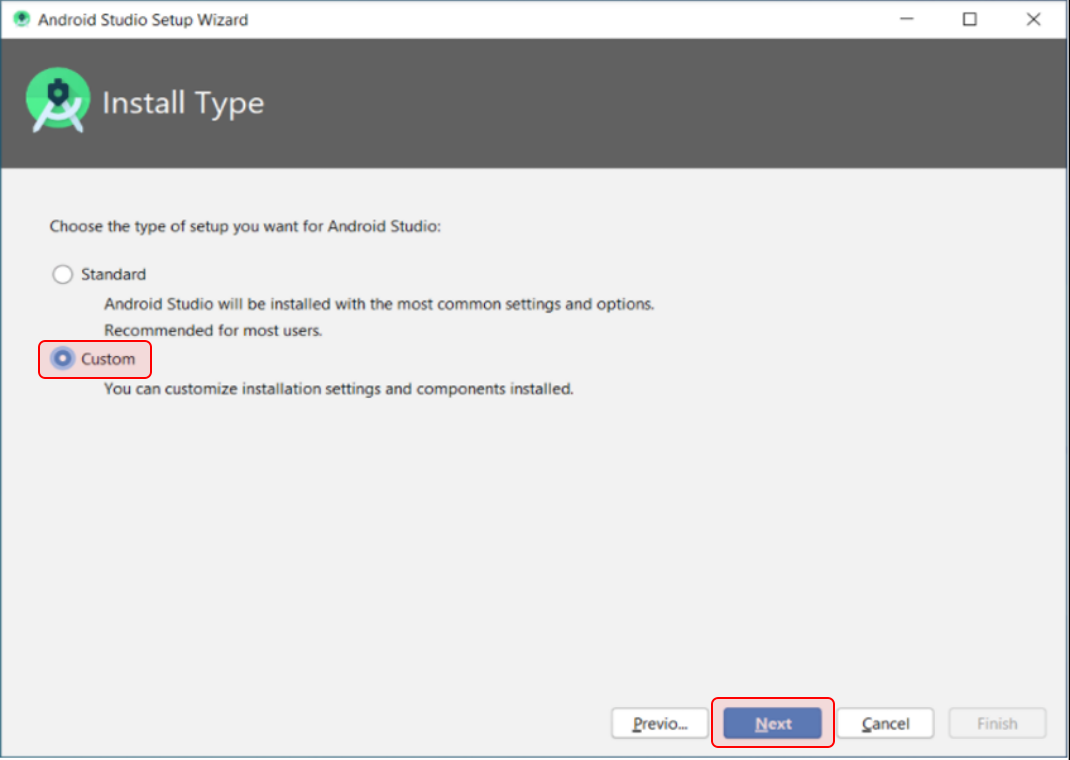
1. Instalación de Wizard Android Studio

**Figura 15**. Asistente de Instalación “Wizard Android Studio”



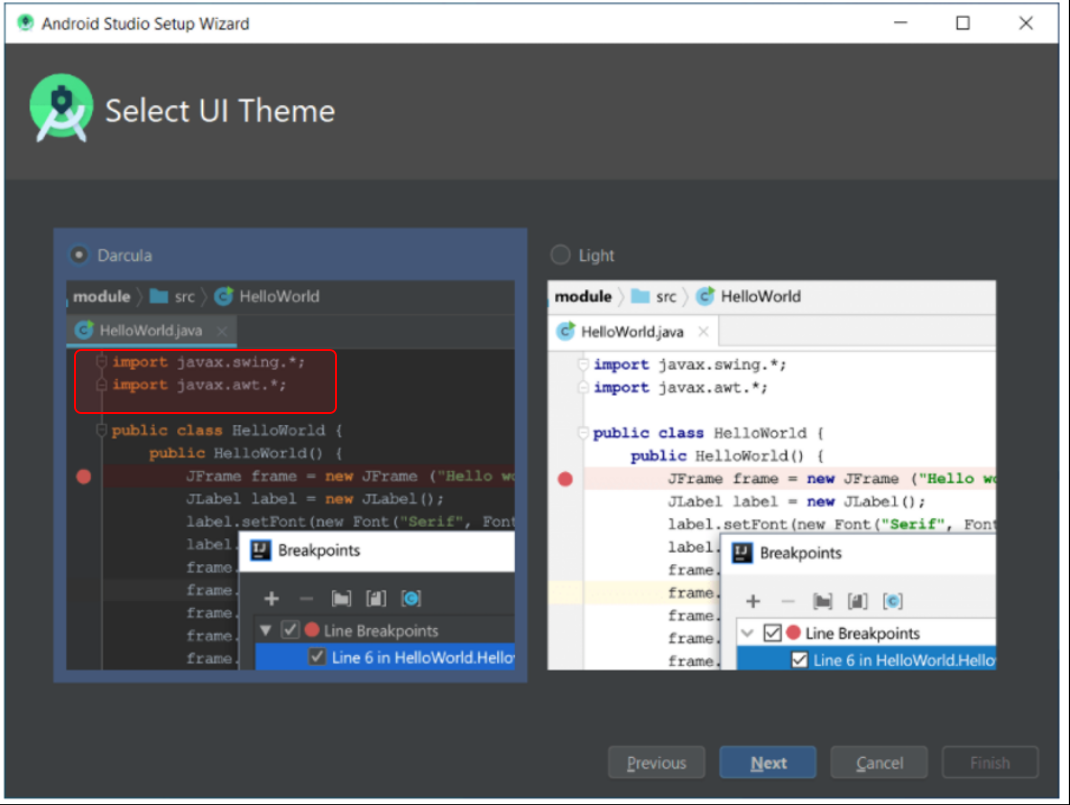
1. Clic en *Next*

**Figura 16.** Selección tipo configuración



1. Para el tipo de configuración marcar la opción  *Custom* (Personalizado) y clic en Next

**Figura 17.** Selección tema visual



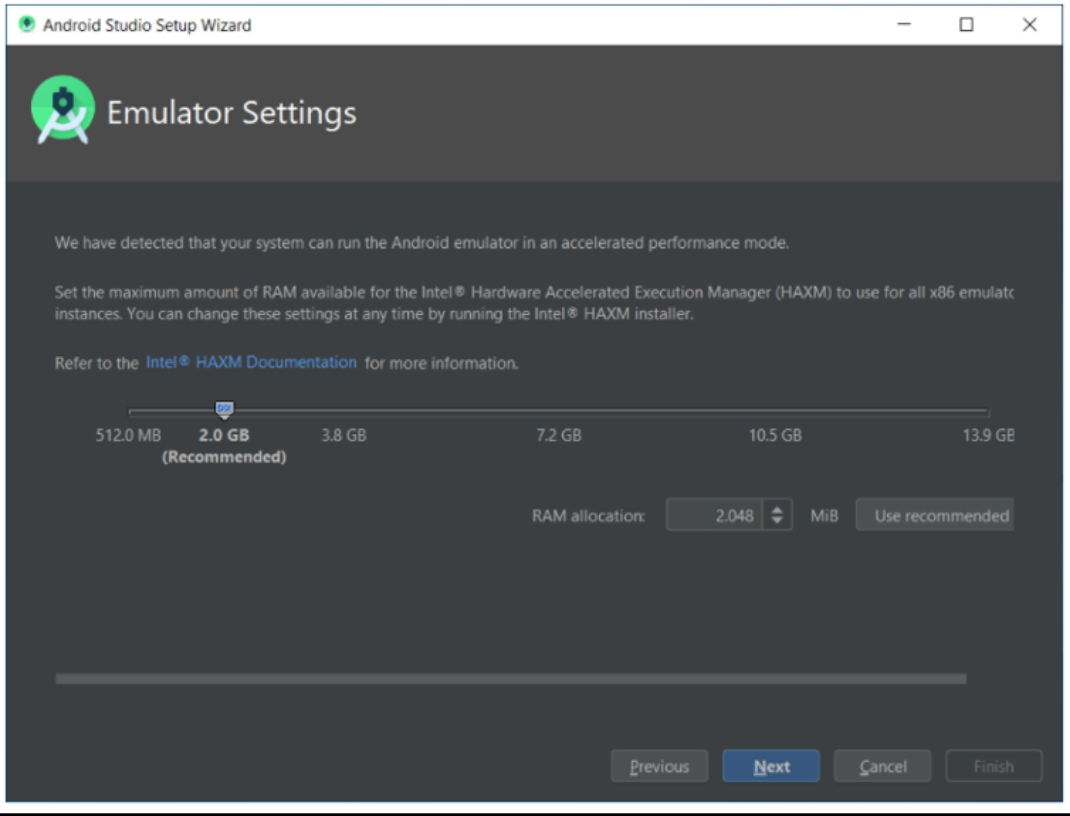
**I)** En la siguiente pantalla del asistente marcar los componentes adicionales a instalar. En el campo Android SDK Location se indica la ruta donde se desea instalar el SDK de Android.

**Figura 18.** Configuración de componentes



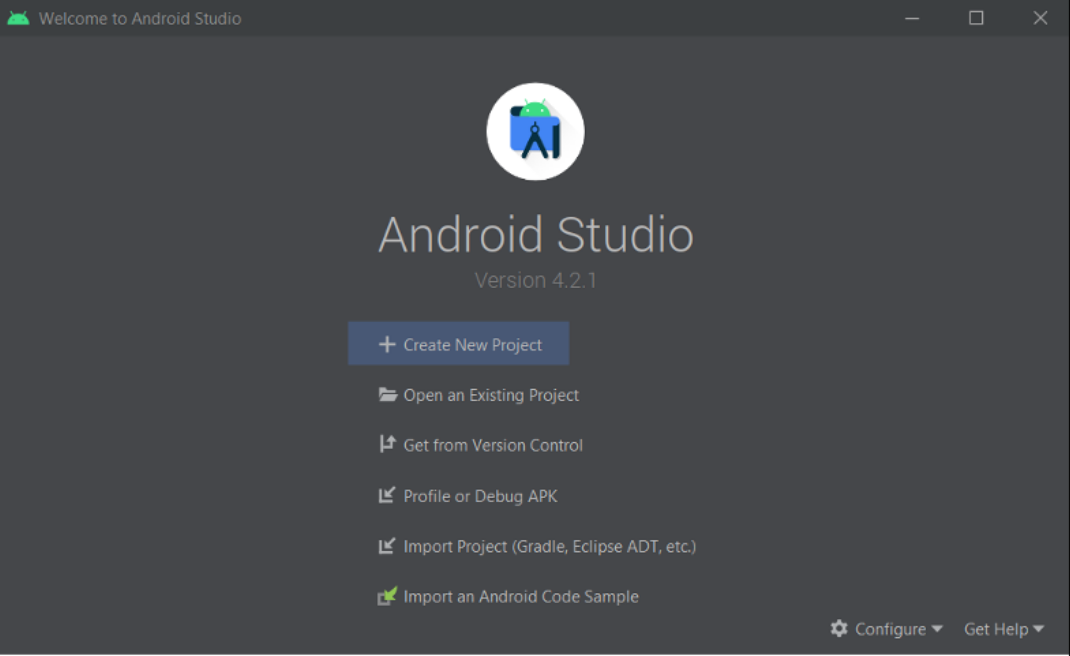
**J)** Indicar la cantidad de memoria que se utilizará para el emulador y clic en “Next”

**Figura 19.**  Configuración del Emulador



1. Tras finalizar el asistente de inicio aparecerá la pantalla de bienvenida de Android Studio para este ejemplo en su versión 4.2.1

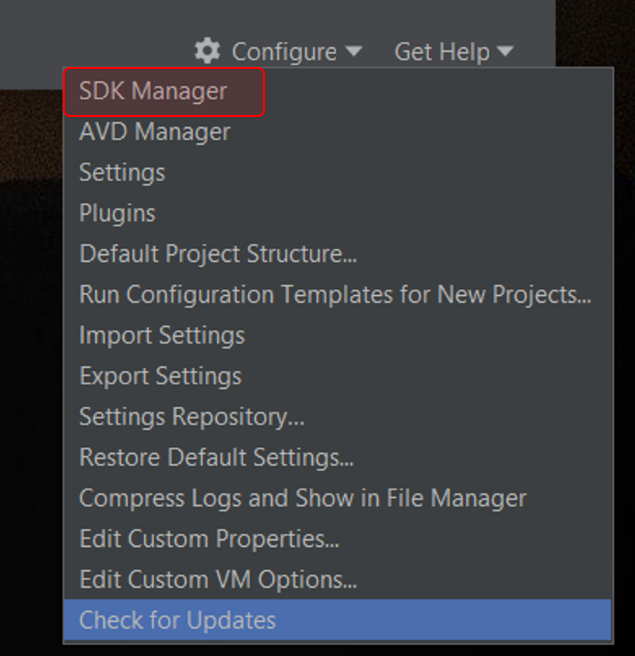
**Figura 20.**  Bienvenida Android Studio



***SDK Manager***

El *SDK Manage*r permite gestionar las versiones de Android que están instaladas, así como otras herramientas necesarias para el desarrollo de aplicaciones móviles. El siguiente paso será revisar los componentes que se han instalado del SDK de Android Studio e instalar o actualizar componentes adicionales si fuera necesario para el desarrollo de las aplicaciones. En la pantalla de inicio de Android Studio, seleccione la opción "Configure" y haga clic en *SDK Manager.*

**Figura 22.** *SDK Manager*

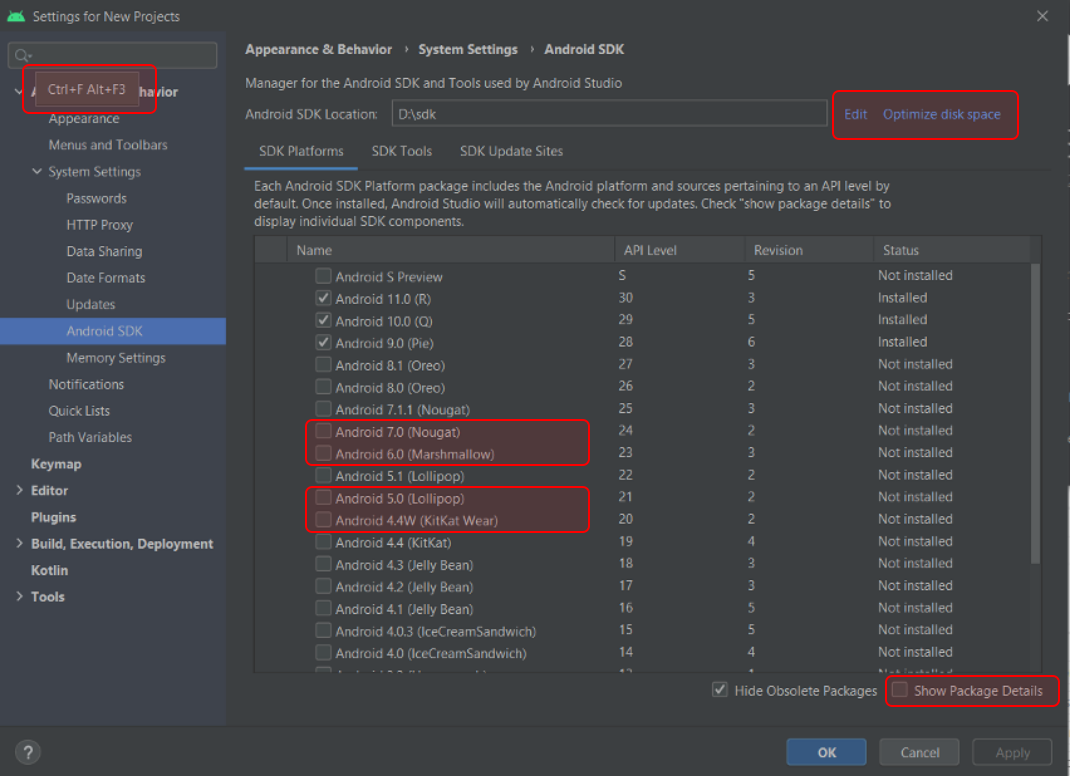


***SDK Platforms***

*SDK Platforms* permite seleccionar los componentes y librerías necesarios para desarrollar sobre cada una de las versiones de Android. Por ejemplo, si se desea probar la aplicación en dispositivos con Android 8 y Android 10, se deben descargar las plataformas correspondientes a las versiones 26 y 29, respectivamente. Para ver los subcomponentes de cada plataforma, seleccione la opción "Show Package Details" situada en la parte inferior de la ventana. Para cada versión de Android instalada, se deben tener al menos los siguientes dos elementos:

* Android SDK Platform
* Google APIs
* Intel x86 Atom System Image

**Figura 23.** *SDK Platforms*



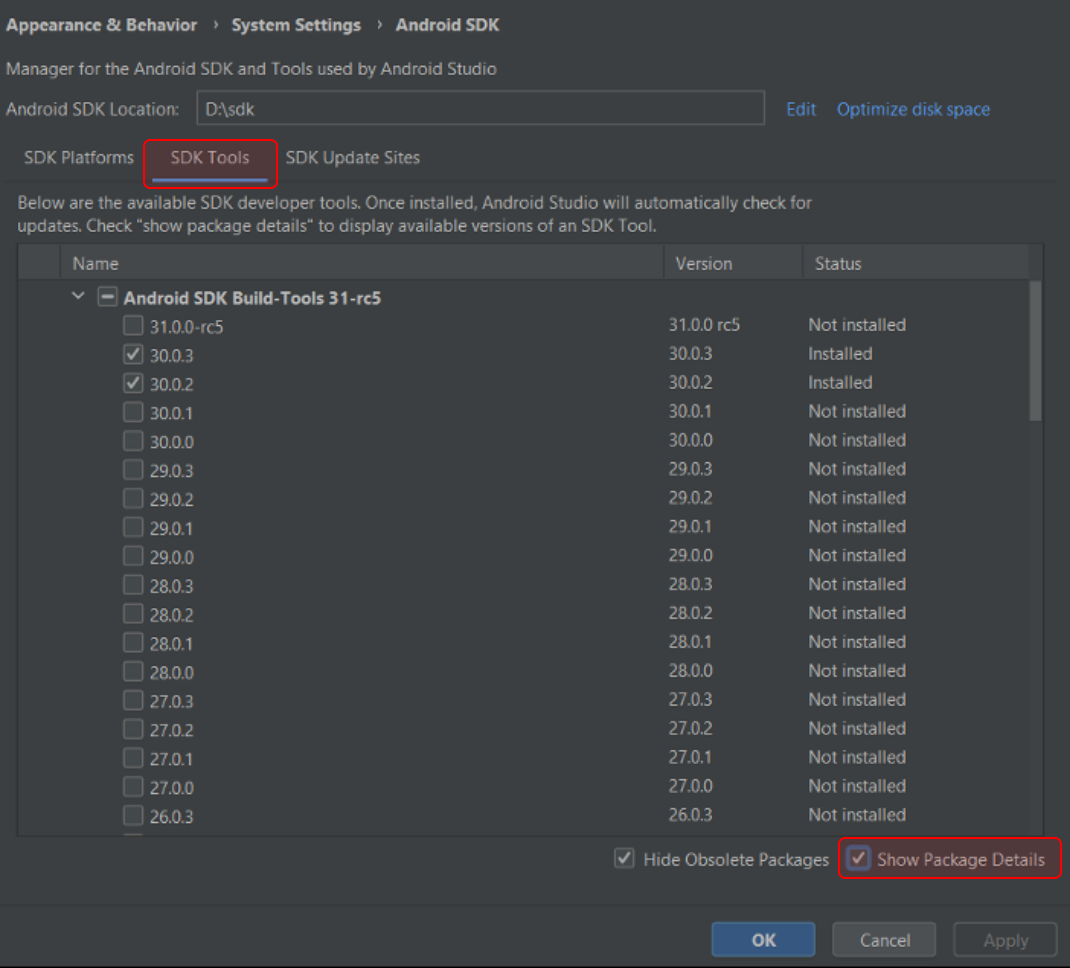
***SDK Tools***

*SDK Tools* son componentes para el SDK que incluyen herramientas para el desarrollo y la depuración. Los indispensables por el momento, que ya deberían aparecer instalados por defecto, son los siguientes:

* Android SDK Build-Tools
* Android SDK Platform-Tools
* Android Emulator

Con estos pasos, ya se tendría configurado Android Studio para dar inicio al desarrollo de las aplicaciones.

Figura 24. SDK Tool*s*



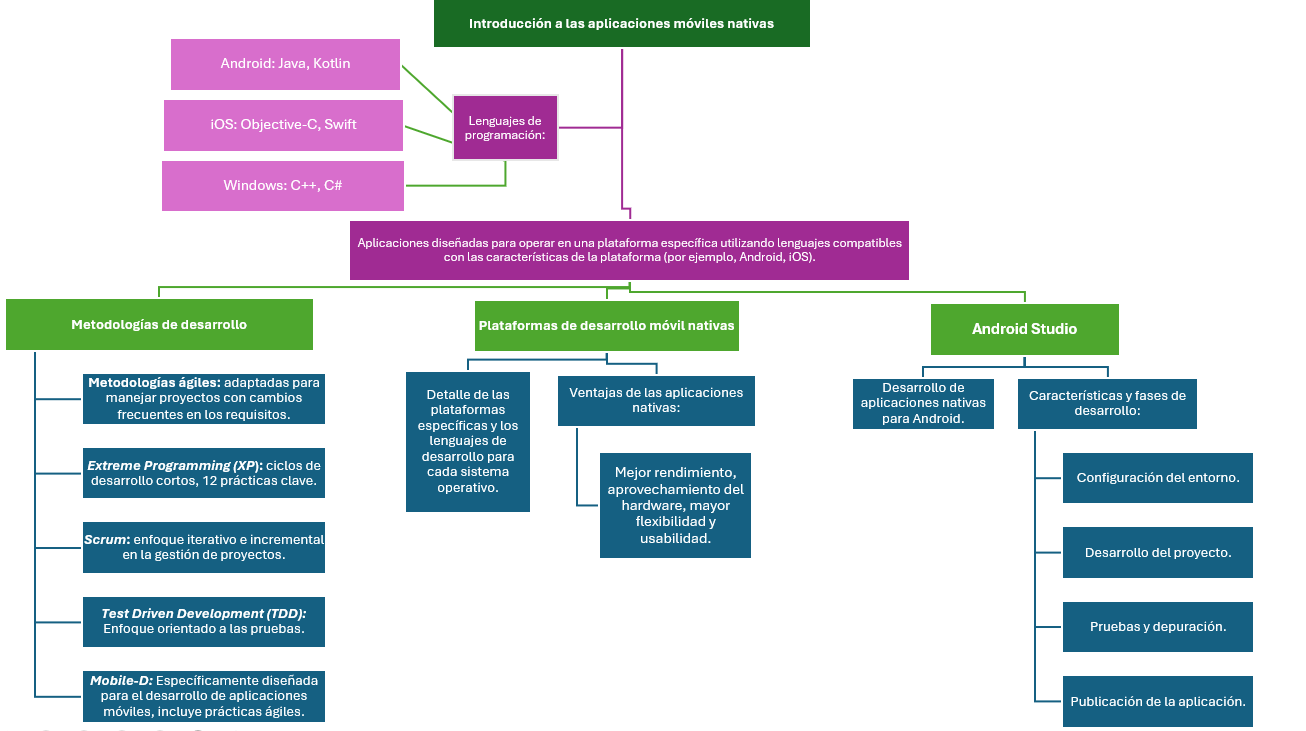
**5. Componentes de una aplicación Android**

Los componentes de la aplicación son mecanismos de creación básicos de una aplicación para Android. Cada componente es un punto de entrada por el que el sistema o un usuario ingresan a una aplicación. Algunos componentes dependen de otros.

|  |
| --- |
| Slide  CF015\_5\_Componentes de una aplicación Android |

1. **SÍNTESIS**

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.

****

1. **ACTIVIDADES DIDÁCTICAS (Se debe incorporar mínimo 1, máximo 2)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Desarrollo de aplicaciones móviles nativas |
| Objetivo de la actividad | Vincular términos clave con sus definiciones apropiadas para reforzar el entendimiento del desarrollo de aplicaciones móviles nativas. |
| Tipo de actividad sugerida | Relacionar conceptos |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | CF015\_Actividad didactica |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Metodología XP | Perú, M. H. (2019, 15 de diciembre). ¿Qué es la programación extrema? Metodología Ágil XP (Ciclo de Vida XP, Prácticas Básicas de XP) [Video]. YouTube. | Video YouTube | <https://www.youtube.com/watch?v=tCl33R9jHBk> |
| Metodología Scrum | Henao, C. (2018, 27 de junio). #3. SCRUM en 6 minutos Metodologías Ágiles [Video]. YouTube. | Video YouTube | <https://www.youtube.com/watch?v=HhC75IonpOU> |
| Mobile-D | Montoya, J. L. (2021, 15 de abril). Exposición metodologías de desarrollo de software. | Mobile-D | <https://www.youtube.com/watch?v=I3yMHHgQCEk> |
| Entornos de Desarrollo | Peñalba, I. (2021). Hacer aplicaciones para Android: ¿Qué es un entorno de desarrollo y cuál es el mejor? El Español. | Página | <https://www.elespanol.com/elandroidelibre/20200518/hacer-aplicaciones-android-entorno-desarrollo-mejor/490952339_0.html> |
| Visual Studio | Microsoft Build. (2023). Información general sobre Visual Studio. | Página | <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2019> |
| Android | Developers. (2020, 7 de mayo). Arquitectura de la plataforma. | Página Oficial | <https://developer.android.com/guide/platform?hl=es-419> |
| Versiones Android | ADSLZone. (2021, 3 de marzo). Qué es Android. | Página | <https://www.adslzone.net/reportajes/software/que-es-android> |
| Versiones Android Studio | Developer. (2021, 14 de junio). Archivos de descarga de Android Studio. | Página | <https://developer.android.com/studio/archive?hl=es-419> |
| Instalación Android | Developer. (2020, 21 de diciembre). Cómo instalar Android Studio. | Página Oficial | <https://developer.android.com/studio/install?hl=es-419> |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Dalvik: | máquina virtual de Android. |
| Manifiesto ágil: | es un documento redactado con principios para mejorar la forma de desarrollar. |
| Wereable: | dispositivo conectado que se puede llevar puesto y que se conecta al teléfono móvil. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., & Salo, O. (2004). Mobile-D: An agile approach for mobile application development. En Companion to the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications, 174-175. ACM.

Astel, D. (2003). Test-driven development: A practical guide. Prentice Hall PTR.

Avison, D. E., & Fitzgerald, G. (2006). Information system development. McGraw-Hill Education.

Balafuera, Y. D. A. (2015). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles: Estado actual. Revista de tecnología, 12(2). <https://doi.org/10.18270/rt.v12i2.1291>

Letelier, P., Canós, J. H., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. Presentado en VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Alicante, España, 1-8. Addison Wesley.

Olmos, F. (2020, 19 de octubre). El origen de Android: Características y costos. Fredy Olmos. <https://fredyolmos.com/ciencia-y-tecnologia/el-origen-de-android-costos-caracteristicas/>

Takeuchi, H., & Nonaka, I. (1986). The new new product development game. Harvard Business Review.

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor (es) | Zulema Yidney León Escobar | Experto Temático | Regional Distrito Capital - Grupo De Apoyo Administrativo - CGI | Noviembre 2022 |
| Paola Alexandra Moya | Evaluadora instruccional | Regional Antioquia - Centro de Servicios de Salud | Mayo de 2024 |
|  | Olga Constanza Bermúdez Jaimes | Responsable Línea de Producción Antioquia | Regional Antioquia - Centro de Servicios de Salud | Mayo de 2024 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |