



# DESARROLLO WEB FULLSTACK -INTERMEDIO

Autor de contenido

**Andrés Fernando Pineda Guerra**



# Tabla de Contenido



## Presentación

En el curso de desarrollador Full Stack como componente intermedio, podrán adquirir las habilidades y lenguajes necesarios para el desarrollo web, enfocándose en sus grandes pilares, como lo son Front End, Back End, Diseño y modelamiento de aplicaciones y documentación de código.

El curso trata temas emergentes tales como, la seguridad informática, desarrollo de aplicaciones móviles, gestión de base de datos, todo esto basado en la metodología Scrum. De la misma manera, se hace énfasis en el manejo de proyectos tanto en los módulos de desarrollo como los módulos de gestión de proyectos de TI.

## Objetivos del curso (competencias)



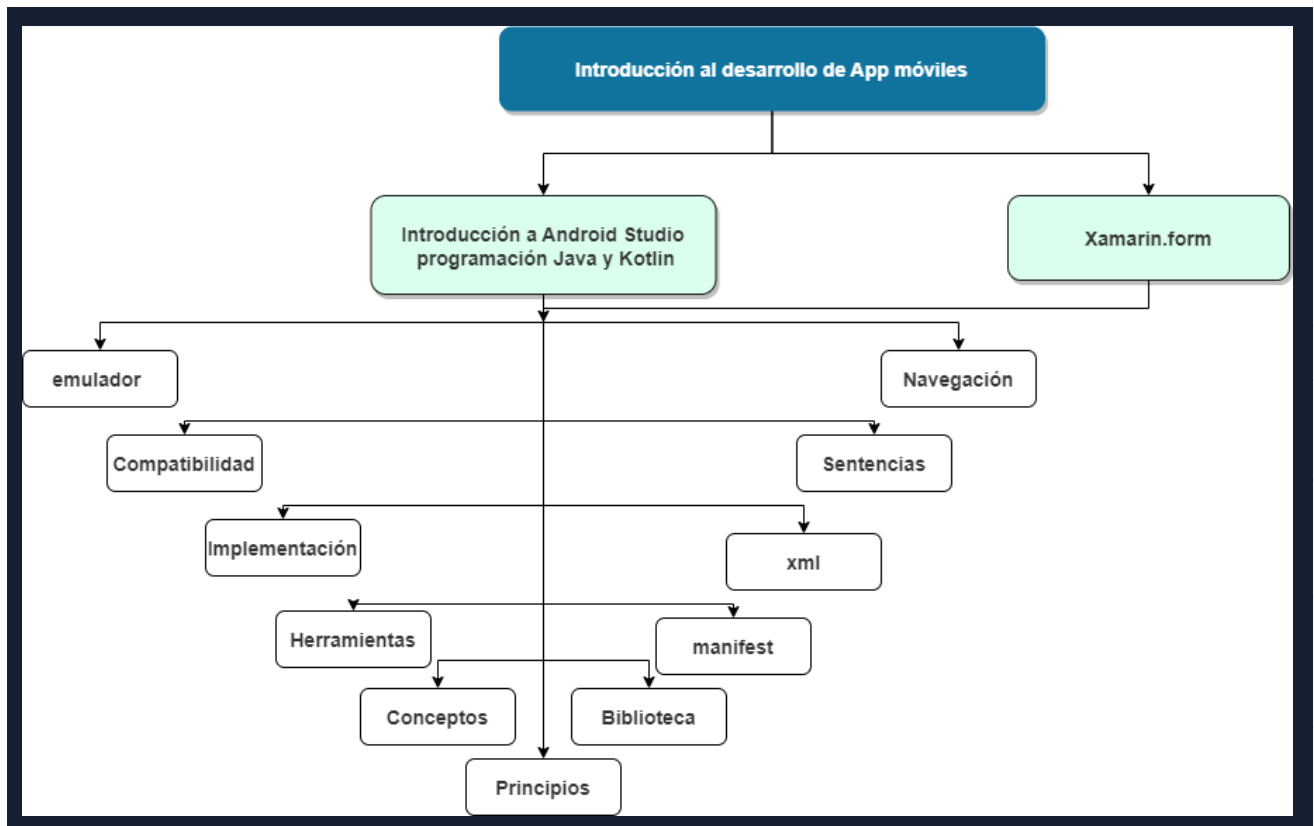
### Objetivo general

Formar a los participantes en el desarrollo web en todo el ciclo de vida del software, en donde adquieran los conocimientos básicos para implementar soluciones web.

### Objetivo específico

- Conocer los conceptos y teoría básica del desarrollo web.
- Identificar y conocer los diferentes lenguajes de programación y herramientas para el desarrollo web.
- Aplicar las diferentes tecnologías web, tendencias y herramientas en el desarrollo de soluciones web enfocadas a proyectos.
- Diseñar, desarrollar e implementar soluciones web básicas en donde se integren los componentes de Front End, Back End, seguridad, redes y buenas prácticas utilizando metodologías ágiles.
- Identificar y conocer los conceptos básicos para el desarrollo móvil, así como aplicar su desarrollo en aplicaciones básicas.

## Mapa de contenido de la unidad



## Módulo 8 Introducción al desarrollo de App móviles

### Ideas clave

Introducción a kotlin y Java para programación de aplicaciones móviles, interfaz y principales herramientas.

Introducción a xamarin y xamarin forms, características y funcionalidades.

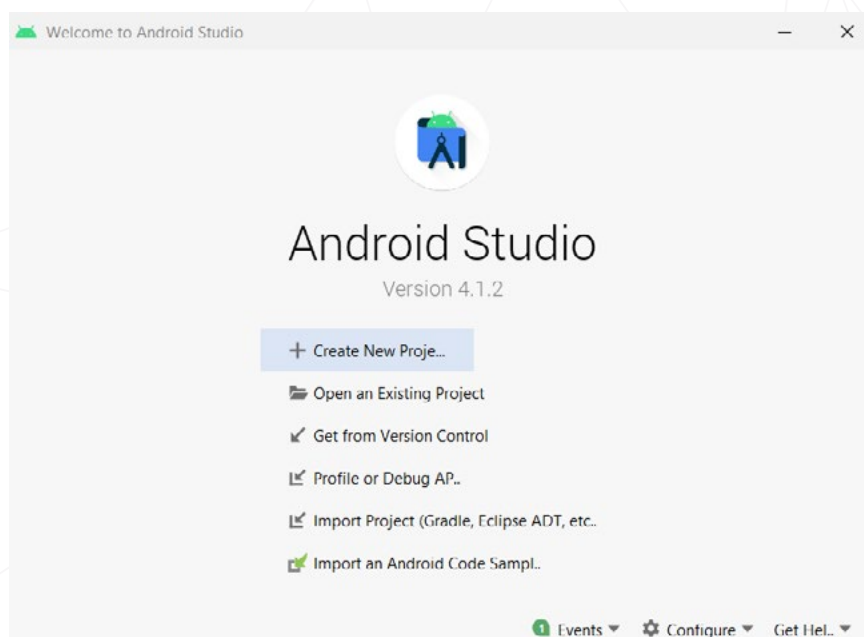
## 8.1. Introducción a Android Studio programación Java y Kotlin

### Tipos de datos

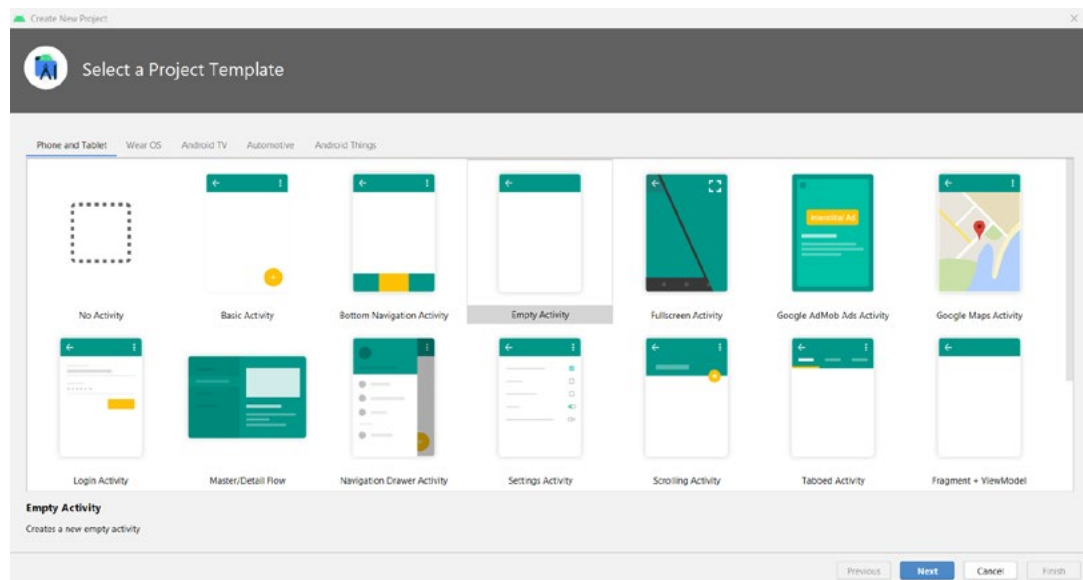
Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan tu productividad cuando desarrollas apps para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle
- Un emulador rápido y cargado de funciones
- Un entorno unificado donde puedes desarrollar para todos los dispositivos Android
- Aplicación de cambios para insertar cambios de código y recursos a la app en ejecución sin reiniciarla
- Integración con GitHub y plantillas de código para ayudarte a compilar funciones de apps comunes y también importar código de muestra
- Variedad de marcos de trabajo y herramientas de prueba
- Herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de versiones, entre otros
- Compatibilidad con C++ y NDK
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine

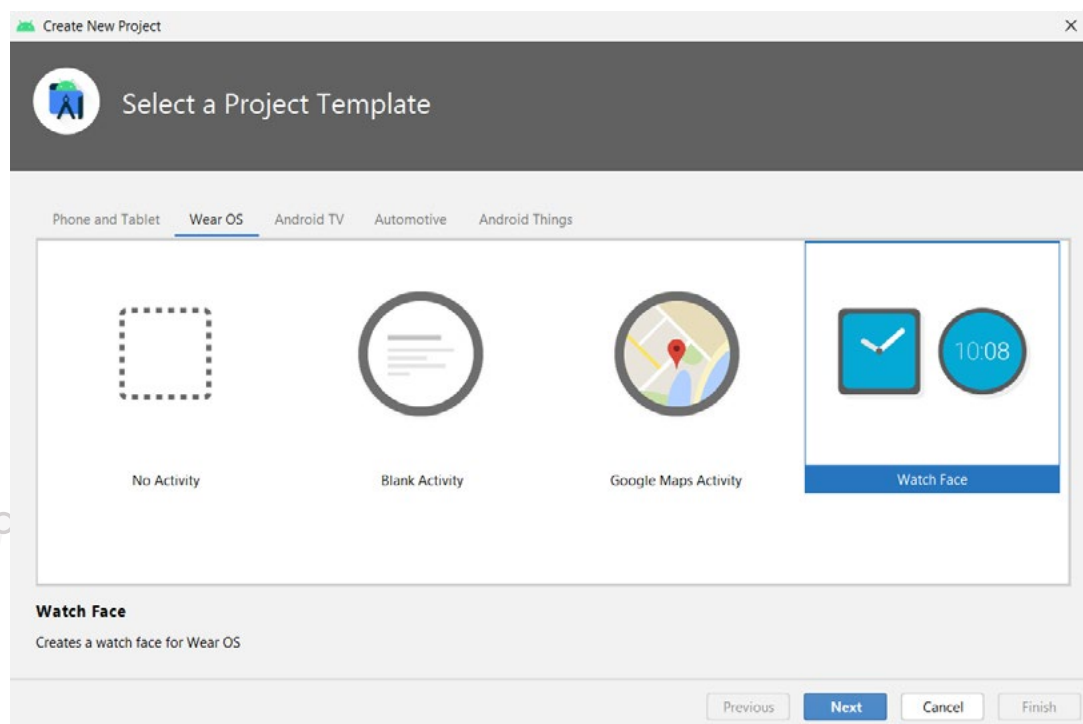
### Creación de nuevos proyectos



- Permite la creación de proyectos con bocetos bases con el fin de crear aplicaciones para celulares o tablets.

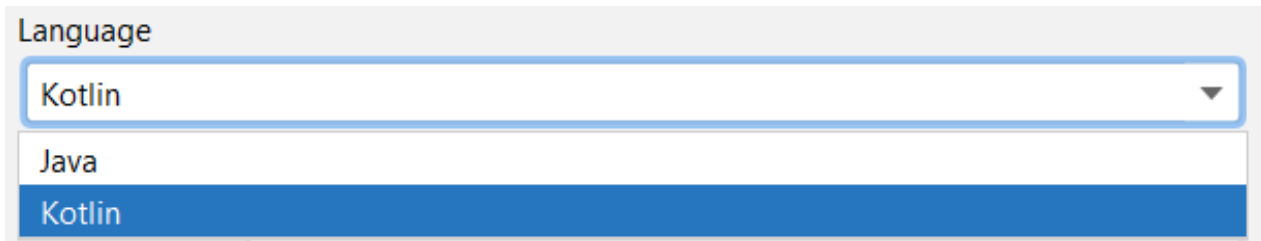


- Permite crear aplicaciones para wear OS, generador plantillas de mapas activos, actividades en blanco, entre otros.



- De la misma manera permite crear aplicaciones para Android TV o Android Things.

## Lenguajes



### Kotlin

Kotlin es un lenguaje de programación expresivo y conciso que permite reducir los errores de código comunes y se integra fácilmente en las apps existentes. Si quieres crear una app para Android, se recomienda comenzar con Kotlin a fin de aprovechar sus mejores funciones.

- Expresivo y conciso: Se puede hacer más con menos. Expresa tus ideas y reduce la cantidad de código estándar. El 67% de los desarrolladores profesionales que usan Kotlin afirman que aumentó su productividad.
- Código más seguro: Kotlin cuenta con muchas funciones de lenguaje que te ayudarán a evitar errores de programación comunes, como excepciones de puntero nulo. Las apps para Android que contienen código de Kotlin tienen un 20% menos de probabilidades de fallar.
- Interoperabilidad: Puedes llamar al código basado en Java desde Kotlin, o viceversa. Como Kotlin es 100% interoperable con el lenguaje de programación Java, puedes implementarlo en tus proyectos de la forma que quieras.
- Simultaneidad estructurada: Las corrutinas de Kotlin hacen que sea tan fácil trabajar con código asíncrono como con código de bloqueo. Las corrutinas simplifican en gran medida la administración de tareas en segundo plano para todo, desde las llamadas de red hasta el acceso a los datos locales.

### Java

Java es caracterizado por ser un lenguaje de programación de propósito general, tipado y orientado a objetos, el cual permite el desarrollo desde aplicaciones básicas, pasando por aplicaciones empresariales hasta aplicaciones móviles.

Java es un lenguaje con diversas características, entre ellas tenemos:

### **Orientado a objetos**

En Java cualquier elemento es un objeto, y dichos objetos se encapsulan los datos los cuales son accedidos mediante métodos.

### **Seguro**

Ningún objeto puede protegerse contra una modificación externa, duplicación o engaño. Es que un lenguaje con apuntadores poderosos tal vez tengan errores serios que comprometen la seguridad. Java elimina estas amenazas de un golpe al suprimir por completo los apuntadores de lenguaje.

### **Arquitectura neutral**

Independientemente de que se ejecute en una arquitectura de 32bits o de 64bits. En Java los tipos de datos siempre ocupan lo mismo.

### **Portable**

Java no tiene nada que dependa de la plataforma, lo cual le hace que sea portable a diferentes plataformas.

### **Robusto**

El lenguaje Java intenta controlar las situaciones de error en los procesos de compilación y de ejecución, reduciendo de esta manera el riesgo de fallo.

### **Alto rendimiento**

Debido al manejo de Just-In-Time permite en java tener un alto rendimiento.

### **Entornos de funcionamiento**

#### **Sistemas de servidor**

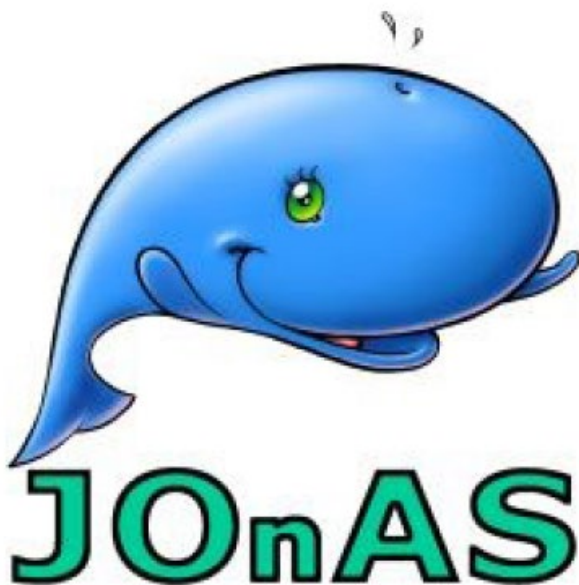
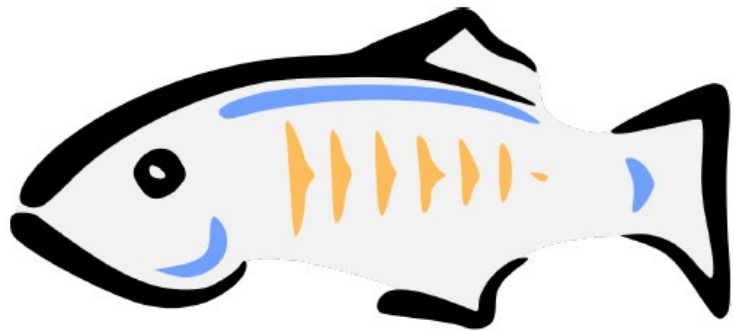
Se conoce como servidor java de aplicaciones, en donde sus aspectos fundamentales se encuentran distribuidos, un servidor de java permite que sus aplicaciones funcionen todo el tiempo, es decir, 24 horas al día los 365 días del año, para esto se debe contar con un equipo con las características adecuadas para que la aplicación corra de manera correcta, en esto se deben tener en cuenta las variables de durabilidad, mantenimiento en el tiempo.

A continuación, se muestran algunos tipos de servidores en Java:



## ClassFish

ClassFish es un servidor de aplicaciones de código abierto iniciado por Sun Microsystems para la plataforma Java EE y ahora patrocinado por la corporación Oracle.



## JOnAS

JOnAS es una implementación de software libre de la especificación de JEE para servidores de aplicación, desarrollada y alojada en el consorcio ObjectWeb. Proporciona un contenedor de EJB a través de EasyBeans y esta disponible con un servidor Tomcat y un servidor Jetty. JVM 6 esta soportado y los intentos de ejecutar en una pila libre con el classpath GNU son prometedores. Además de la pila de Java EE, JOnAS proporciona mecanismos de alto nivel para cluster y así proporcionar escalabilidad y alta disponibilidad.

## Apache TomEE

Apache TomEE es la edición Java EE de Apache Tomcat, el cual combina varios proyectos Java incluyendo Apache OpenEJB, ApacheOpenWebBeans, Apache OpenJPA, ApacheMyFaces y otros. El proyecto tiene una certificación de la corporación Oracle como un servidor compatible del perfil web Java EE 6, que es una parte de toda la especificación Java EE



## Dispositivos móviles y sistemas embebidos

La creación de J2ME es una versión creada específicamente para el desarrollo en dispositivos móviles y electrodomésticos.

Java comenzó su andadura como lenguaje de programación a mediados de la década de los noventa del siglo pasado. Originalmente fue concebido como un lenguaje para poder programar un amplio rango de aparatos electrónicos con capacidades de conectividad a partir de otro dispositivo del tipo de un asistente personal digital. El espíritu inicial era realizar una adaptación de C++, tomando lo mejor de él y a la vez mejorándolo y que se adecuara a las restrictivas condiciones ofrecidas por los chips de los aparatos a programar, teniendo como principales objetivos la fiabilidad y la seguridad. Además, se intentaba que una vez que fuera desarrollado el programa, éste se pudiera ejecutar en varios tipos diferentes de aparatos sin necesidad de volver a compilarlo.

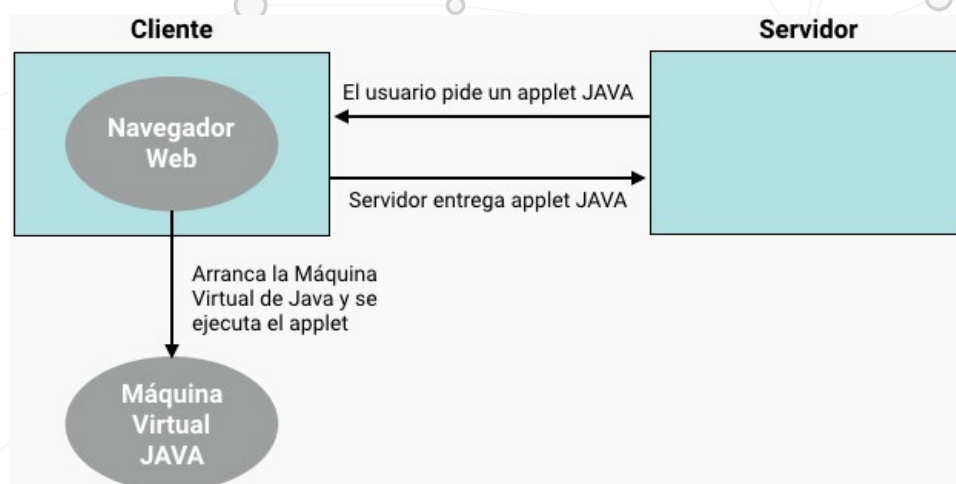
Con la llegada de Internet y los primeros navegadores para la World Wide Web, los desarrolladores de Java se dieron cuenta de su aplicabilidad a este nuevo medio, naciendo así la tecnología de los applets de Java, que permite, de nuevo, poder desarrollar una aplicación una única vez y ejecutar tantas veces como se desee en un conjunto heterogéneo de ordenadores conectados a la Red.

## Navegador web

Java permite el desarrollo web mediante el uso de Applets, que permiten ser embebidas en HTML, las cuales son ejecutadas por el navegador.

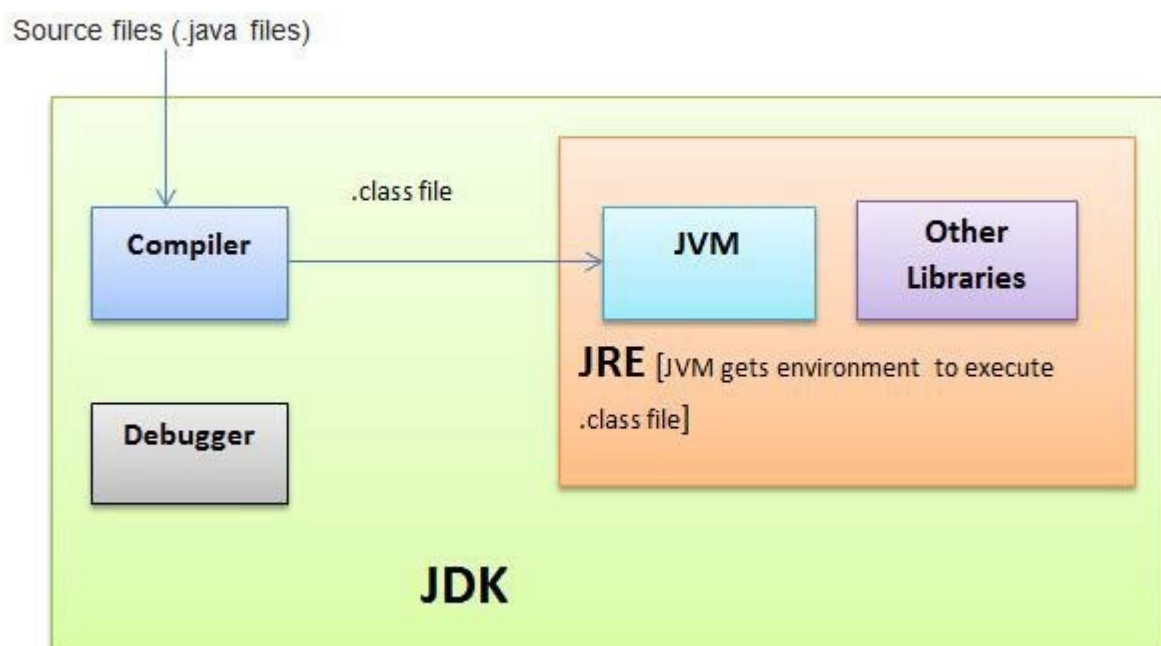
Las Applets son ejecutadas de la siguiente manera:

- Se realiza una consulta en el navegador.
- El navegador procesa y carga el HTML
- El navegador identifica las clases del Applet de java
- Se ejecuta el Applet java.



## Aplicaciones de escritorio

Es una de las aplicaciones de java más utilizadas en el mercado, basados en el entorno de ejecución JRE el cual, es un software que los programas de Java necesitan para ejecutarse correctamente. Java es un lenguaje de computación en el que se basan numerosas aplicaciones web y móviles actuales. El JRE es una tecnología subyacente que comunica el programa de Java con el sistema operativo. Actúa como traductor y facilitador, y brinda todos los recursos de modo que, una vez que escribe un software de Java, se ejecuta en cualquier sistema operativo sin necesidad de más modificaciones.



## Líneas de comandos básicos

Se muestran los comandos básicos ejecutados en java con el entorno de desarrollo NetBeans, en el siguiente ejemplo se crea la clase "HelloWorld" en donde se maneja una sintaxis específica y vemos la línea de comando "System.out.println", esta sentencia permite la impresión de un texto en pantalla.

```

class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World");
    }
}
  
```

## void

void indica que no se va a pasar ningún dato y en su lugar se deja vacía la ubicación correspondiente.

## args

args se utiliza para pasar valores en forma de matriz desde la línea de comandos al método main de Java.

## string

```
1 import java.util.Scanner;
2 class AreaOfRectangle
3 {
4     public static void main(String args[])
5     {
6         Scanner s= new Scanner(System.in);
7
8         System.out.println("Enter the length:");
9         double l= s.nextDouble();
10        System.out.println("Enter the breadth:");
11        double b= s.nextDouble();
12
13
14
15        double area=l*b;
16        System.out.println("Area of Rectangle is: " + area);
17    }
18 }
```

string define una cadena en Java.

## System.out.println

Con este comando tienes una salida de texto a través de la consola. Después de la salida, también se inserta una nueva línea.

## private

private garantiza que una función sólo se ejecute dentro de la clase correspondiente.

## int (integer)

int especifica números enteros en Java, que pueden ser positivos o negativos.

## double

double especifica números decimales y almacena datos con doble precisión.

if ( condición ) {Primer indicación}else {Segunda indicación}

Con if y else se puede comprobar si una variable tiene un valor determinado. Si cumple la condición, se ejecuta la primera indicación. Si la condición no se cumple, se ejecuta la segunda indicación.

## While ( condición de terminación ) {indicación;}

Mientras la condición de terminación sea verdadera, la indicación se ejecuta una y otra vez. Si la condición deja de cumplirse, el programa se termina.

## do {indicación;} while ( condición de terminación )

A diferencia del while, el do while es un bucle controlado por el pie. Esto significa que la indicación se ejecuta primero y el programa solo comprueba si la condición de terminación es "true" o "false".

## for ( int i = 1;i <= 5;i++ {System.out.println}

También con for, una sentencia se ejecuta mientras la condición de terminación sea "true". Sin embargo, a diferencia del bucle while, el for tiene una variable de ejecución que puede utilizarse para determinar de antemano la frecuencia con la que se va a repetir la sentencia.

## Realizar solicitud de entradas al usuario:

```
1 import java.util.Scanner;
2 class AreaOfRectangle
3 {
4     public static void main(String args[])
5     {
6
7         Scanner s= new Scanner(System.in);
8
9         System.out.println("Enter the length:");
10        double l= s.nextDouble();
11        System.out.println("Enter the breadth:");
12        double b= s.nextDouble();
13
14
15        double area=l*b;
16        System.out.println("Area of Rectangle is: " + area);
17    }
18 }
```



## Estructura del proyecto en Android Studio

Cada proyecto de Android Studio incluye uno o más módulos con archivos de código fuente y archivos de recursos. Entre los tipos de módulos se incluyen los siguientes:

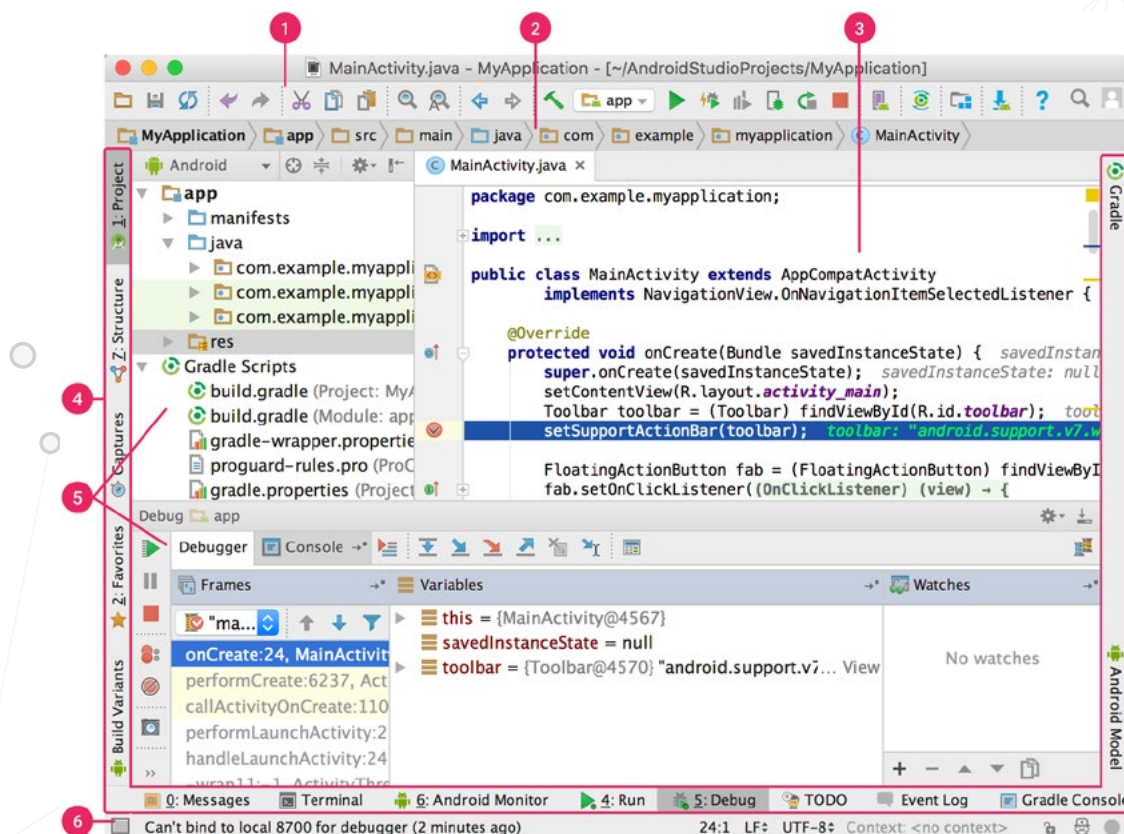
- Módulos de apps para Android
- Módulos de biblioteca
- Módulos de Google App Engine

De manera predeterminada, Android Studio muestra los archivos de tu proyecto en la vista de proyecto de Android, como se ve en la figura 1. Esta vista está organizada en módulos para que puedas acceder rápidamente a los archivos fuente clave de tu proyecto.

Puedes ver todos los archivos de compilación en el nivel superior de Secuencias de comando de Gradle y cada módulo de app contiene las siguientes carpetas:

- manifests: contiene el archivo AndroidManifest.xml.
- java: contiene los archivos de código fuente Java, incluido el código de prueba de JUnit.
- res: contiene todos los recursos sin código, como diseños XML, strings de IU e imágenes de mapa de bits.

## Interfaz de usuario



1. La barra de herramientas te permite realizar una gran variedad de acciones, como ejecutar tu app e iniciar las herramientas de Android.
2. La barra de navegación te ayuda a explorar tu proyecto y abrir archivos para editar. Proporciona una vista más compacta de la estructura visible en la ventana Project.
3. La ventana del editor es el área en la que puedes crear y modificar código. Según el tipo de actividad actual, el editor puede cambiar. Por ejemplo, cuando ves un archivo de diseño, el editor muestra el Editor de diseño.
4. La barra de la ventana de herramientas se encuentra afuera de la ventana del IDE y contiene los botones que te permiten expandir o contraer ventanas de herramientas individuales.
5. Las ventanas de herramientas te brindan acceso a tareas específicas, como la administración de proyectos, la búsqueda, el control de versiones, entre otras. Puedes expandirlas y contraerlas.
6. En la barra de estado, se muestra el estado de tu proyecto y el IDE, además de advertencias o mensajes.

## 8.2. Xamarin.form

### ¿Qué es xamarin?

Xamarin es una plataforma de código abierto para compilar aplicaciones modernas y con mejor rendimiento para iOS, Android y Windows con .NET. Xamarin es una capa de abstracción que administra la comunicación de código compartido con el código de plataforma subyacente. Xamarin se ejecuta en un entorno administrado que proporciona ventajas como la asignación de memoria y la recolección de elementos no utilizados.

Xamarin permite a los desarrolladores compartir un promedio del 90 % de la aplicación entre plataformas. Este patrón permite a los desarrolladores escribir toda la lógica de negocios en un solo lenguaje (o reutilizar el código de aplicación existente), pero conseguir un rendimiento y una apariencia nativos en cada plataforma.

Las aplicaciones de Xamarin se pueden escribir en PC o Mac, y compilar en paquetes de aplicación nativos, como un archivo .apk en Android o .ipa en iOS.

### Xamarin Forms

Xamarin.Forms es un marco de interfaz de usuario de código abierto. Xamarin.Forms permite a los desarrolladores compilar aplicaciones en Xamarin.Android, Xamarin.iOS y Windows desde un único código base compartido.

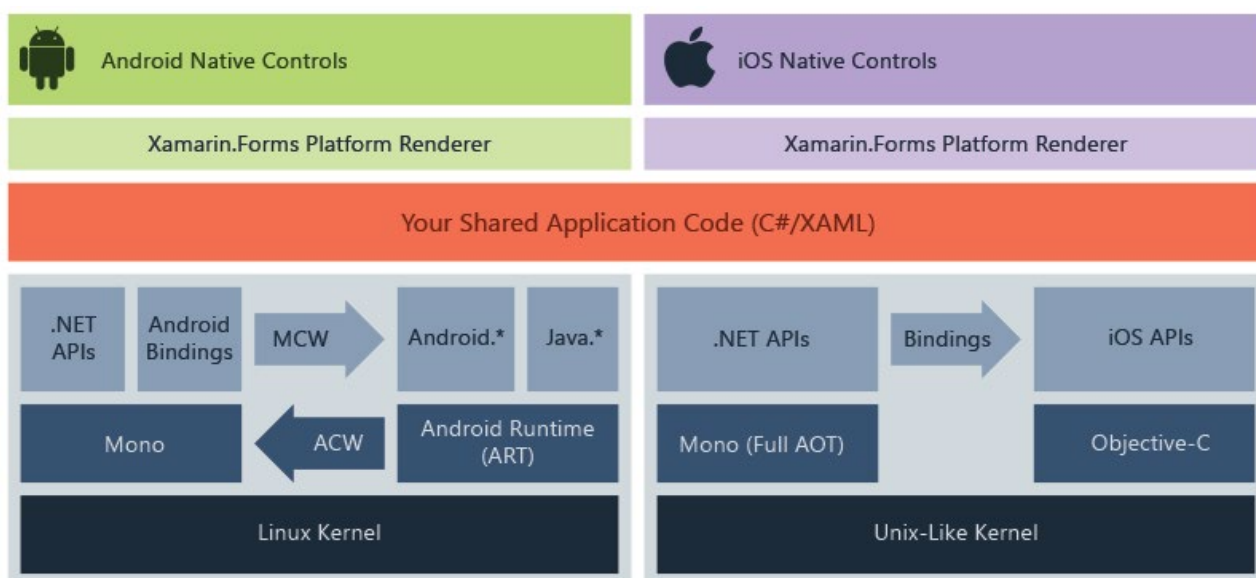
Xamarin.Forms permite a los desarrolladores crear interfaces de usuario en XAML con código subyacente en C#. Estas interfaces se representan como controles nativos con mejor rendimiento en cada plataforma.

## A quién va destinado Xamarin.Forms

Xamarin.Forms es para desarrolladores con los siguientes objetivos:

- Compartir el diseño de la interfaz de usuario entre plataformas.
- Compartir código, pruebas y lógica de negocios entre plataformas.
- Escribir aplicaciones multiplataforma en C# con Visual Studio.

## Cómo funciona Xamarin.Forms



Xamarin.Forms proporciona una API coherente para crear elementos de interfaz de usuario entre plataformas. Esta API se puede implementar en XAML o C#, y admite el enlace de datos para patrones como Model-View-ViewModel (MVVM).

En tiempo de ejecución, Xamarin.Forms usa los representadores de plataforma para convertir los elementos de la interfaz de usuario multiplataforma en controles nativos en Xamarin.Android, Xamarin.iOS y UWP. Esto permite a los desarrolladores obtener una apariencia y un funcionamiento nativos, además de las ventajas del uso compartido de código entre plataformas.

Las aplicaciones Xamarin.Forms suelen estar compuestas de una biblioteca compartida de .NET Standard y proyectos de plataforma individuales. La biblioteca compartida contiene las vistas XAML o C#, y cualquier lógica de negocios, como servicios, modelos



u otro código. Los proyectos de plataforma contienen cualquier lógica o paquete específico de la plataforma que la aplicación necesite.

## Características de xamarin

Xamarin combina las capacidades de las plataformas nativas, mientras agregar características que incluyen lo siguiente:

- Enlaces completos para los SDK subyacentes: Xamarin contiene enlaces para casi todos los SDK de plataforma subyacentes en iOS y Android. Además, estos enlaces están fuertemente tipados, lo que significa que la navegación y el uso son fáciles y que proporcionan una sólida comprobación de tipos en tiempo de compilación y durante el desarrollo. Los enlaces fuertemente tipados producen menos errores en tiempo de ejecución y aplicaciones de mayor calidad.
- Interoperabilidad con Objective-C, Java, C y C++ : Xamarin ofrece funciones para invocar directamente bibliotecas de Objective-C, Java, C y C++, lo que permite usar una amplia variedad de código de terceros. Estas funciones permiten usar bibliotecas existentes de iOS y Android escritas en Objective-C, Java, C o C++. Además, Xamarin ofrece proyectos de enlace que permiten enlazar bibliotecas nativas de Objective-C y Java mediante una sintaxis declarativa.
- Construcciones de lenguaje moderno: las aplicaciones Xamarin se escriben en C#, un lenguaje moderno que incluye mejoras considerables con respecto a Objective-C y Java, como características de lenguaje dinámico, construcciones funcionales como lambdas, LINQ, características de programación en paralelo, genéricos y mucho más.
- Biblioteca de clases base (BCL) sólida: las aplicaciones Xamarin usan la BCL de .NET, una gran colección de clases con características completas y optimizadas, como una compatibilidad eficaz con XML, bases de datos, serialización, E/S, cadenas, redes, etc. El código de C# existente se puede compilar para usarlo en una aplicación, lo que proporciona acceso a miles de bibliotecas que agregan funcionalidad más allá de la BCL.
- Entorno de desarrollo integrado (IDE) moderno: Xamarin usa Visual Studio, un IDE moderno que incluye características como la finalización automática de código, un sofisticado sistema de administración de proyectos y soluciones, una exhaustiva biblioteca de plantillas de proyecto, control de código fuente integrado, etc.
- Compatibilidad multiplataforma móvil: Xamarin ofrece una compatibilidad multiplataforma sofisticada con las tres principales plataformas: iOS, Android y Windows. Se pueden escribir aplicaciones de modo que compartan hasta el 90 % del

código, y Xamarin.Essentials ofrece una API unificada para acceder a recursos comunes de las tres plataformas. El código compartido puede reducir considerablemente los costos de desarrollo y el tiempo de salida al mercado para los desarrolladores para dispositivos móviles.

¿Que más funcionalidades tiene xamarin.forms?

### **Datos y Azure Cloud Services**

- Control de archivos
- Bases de datos locales
- Consumo de un servicio web basado en REST
- Autenticación con Azure Active Directory B2C

### **Implementación y prueba**

- Pruebas automatizadas con Visual Studio App Center
- Mejorar el rendimiento de las aplicaciones
- Publicación de aplicaciones iOS
- Publicación de aplicaciones Android

## Otros materiales para profundizar

### Recursos de video



Codigo 369 (Director). (2022, marzo 8). Xamarin forms desde cero con Visual studio 2022. <https://www.youtube.com/watch?v=GtrDcqRiDIA>

EDteam (Director). (2018, febrero 27). 10 cosas que debes saber sobre Kotlin. <https://www.youtube.com/watch?v=OgH6-ikynw8>

Yo Androide (Director). (2021, febrero 18).  
ANDROID STUDIO DESDE CERO PARA PRINCIPIANTES 2023 | CREAR UNA APLICACION. <https://www.youtube.com/watch?v=juO5H1PBell>

### Referencias bibliográficas de la unidad



Hébuterne, S. (2018). Desarrolle una aplicación Android: Programación en Java con Android Studio. Ediciones ENI.

Hohensee, B. (2014). Introducción A Android Studio. Incluye Proyectos Reales Y El Código Fuente. Babelcube Inc.

Petzold, C. (2015). Creating Mobile Apps with Xamarin.Forms Preview Edition 2. Microsoft Press.



UNIVERSIDAD DISTRICTAL  
FRANCISCO JOSÉ DE CALDAS  
Acreditación Institucional de Alta Calidad