ÍNDICE

PREFACIO

En el mundo del desarrollo todo avanza muy rápido, esto no es solamente una frase es una realidad y cada día vemos avances que pueden ser muy importantes para el futuro aunque en un principio no los notemos. La programación es una asignatura que exige siempre un constante esfuerzo por actualizar los conocimientos de una persona, pero sobre todo mucho amor y pasión por lo que se hace día a día.

En el año 2017 se anunció una noticia que pocos esperaban o que por lo menos no parecía que fuera a suceder a corto plazo, Android el sistema operativo más usado en dispositivos móviles ahora cuenta con otro lenguaje como opción en el desarrollo nativo, Kotlin.

Hay una gran cantidad de lenguajes de programación que aparecen en plataformas, navegadores, consolas de videojuegos, etc., pero podemos decir que Kotlin es el último lenguaje de los que han surgido que ha despertado mayor interés. Solamente Swift de Apple ha logrado algo similar al irrumpir hace algunos años.

Es un lenguaje que se coloca en los primeros lugares como opción para aprender a programar y para iniciarse en este mundo, debido a que es sencillo en interfaz y en lógica en comparación con Java y que es la otra opción que tenemos en el entorno del desarrollo móvil.

Y por ello, es una buena posibilidad para aprender a programar pero sobre todo pensando en el desarrollo móvil de una manera sencilla. Ahora mismo Kotlin no cuenta con una gran cantidad de comunidades o fuentes de información debido a su juventud, pero esto es algo normal y que con el paso del tiempo se irá superando.

Cualquier desarrollador que esté pensando en aprender a programar debe de considerar Kotlin como una de sus primeras opciones por el gran alcance que cuenta, puede servirle tanto para web como para móviles, además es sencillo y funciona en cualquier dispositivo que cuente con una máquina virtual de Java, es decir, en casi cualquier dispositivo programable.

Kotlin es la última gran puerta que se abre con una invitación amena, divertida y sencilla para iniciarse en el mundo de la programación.

Acerca del autor

José Dimas Luján Castillo nació en 1986 en la ciudad de México. En referencia a sus estudios el último grado que obtuvo es una Maestría en Tecnologías de Información en el ITESM.

En su experiencia como desarrollador tiene ya más de 200 aplicaciones realizadas para clientes, plataformas en línea, cursos y ejemplos compartidos por internet. Cuenta con la impartición de clases en más de 18 universidades a nivel presencial en Latinoamérica en los niveles de Licenciatura y Maestría.

Actualmente es colaborador de las plataformas más importantes de educación en línea: LinkedIn Learning, Video2Brain, lynda.com, CodigoFacilito, Escuela Digita, EscuelaIT.

El autor proporciona su sitio web, redes sociales, twitter y correo a continuación:

Sitio web: www.josedlujan.com

Facebook: www.facebook.com/josedlujan

Twitter: www.twitter.com/josedlujan

Correo: josedlujan@gmail.com

Github: https://github.com/josedlujan

Agradecimientos

Se lo dedico con cariño a mis padres, Fabiola y José, y a mi esposa Noemí.

1 INTRODUCCIÓN

Kotlin es un lenguaje de programación que funciona sobre la máquina virtual de Java. Esto quiere decir que si queremos ejecutarlo necesitamos tener en ese dispositivo la máquina virtual para hacer funcionar nuestro programa.

La característica que se describió en el párrafo anterior es la interoperabilidad, en otras palabras podemos decir que Kotlin es un lenguaje 100% compatible con Java. Así que podremos tener proyectos que tengan 100% Kotlin o por ejemplo: 40% Kotlin y 60% de su código en Java, funcionarían en conjunto sin ningún problema.

Kotlin tiene la posibilidad de también poder compilar el código fuente de JavaScript, así que esto abre las posibilidades de la implementación de Kotlin y es una razón más para aprender a usar este lenguaje.

Historia

Kotlin es un lenguaje que fue generado dentro de una empresa, JetBrains, conocida en el mundo del desarrollo por las herramientas que ha creado y que desempeñan un papel crucial en el mundo del desarrollo. Estas son solo algunas de las que están en el catálogo de la compañía:

- Intellij Idea
- PhpStorm
- PyCharm
- WebStorm

Durante aproximadamente un año la compañía dedicó un equipo de desarrollo comandado por Andrey Breslav para crear un lenguaje de programación con las características positivas de los lenguajes Java, C#, Scala y Groovy. La empresa con

base en San Petersburgo sorprendió en el año 2012 cuando presentó oficialmente este lenguaje con el objetivo de que cualquier desarrollador lo pudiera implementar bajo la licencia Apache 2.

El nombre de Kotlin proviene de una isla que se encuentra cerca de San Petersburgo, la ciudad base de la empresa.

Características

Veremos algunas de las características del lenguaje pero en este libro vamos a analizar cada una de ellas y mencionaremos otras que en este apartado no se citen.

Kotlin es un lenguaje de programación que trabaja en principio bajo el paradigma orientado a objetos, pero también existe la posibilidad de utilizarlo como uno orientado por procedimientos. Otra característica que ya mencionamos entre líneas pero que tenemos que dejar claro es que es un lenguaje de programación Open Source, así que se puede aplicar en cualquier tipo de proyecto.

Cuando se habla con algunos de los creadores de Kotlin, estos mencionan que una de las características que ellos buscaban al trabajar en un lenguaje de programación es que sea conciso, es decir, con poco código y claro, o sea, solo lo necesario. Cualquier persona que ha trabajado con Java sabe que esto no suele suceder con este lenguaje, hay ocasiones en las que se coloca código que se necesita poner por la lógica de Java y muchos desarrolladores más de una vez nos hemos preguntado sobre la posibilidad de evitar ese problema, en Kotlin esto se lleva al extremo, probablemente nunca se lo pregunta un desarrollador en ese lenguaje.

Todos los elementos se pueden manejar como objetos, esto quiere decir que mientras en otros lenguajes tenemos el objeto entero y el dato primitivo entero, aquí no hay distinciones.

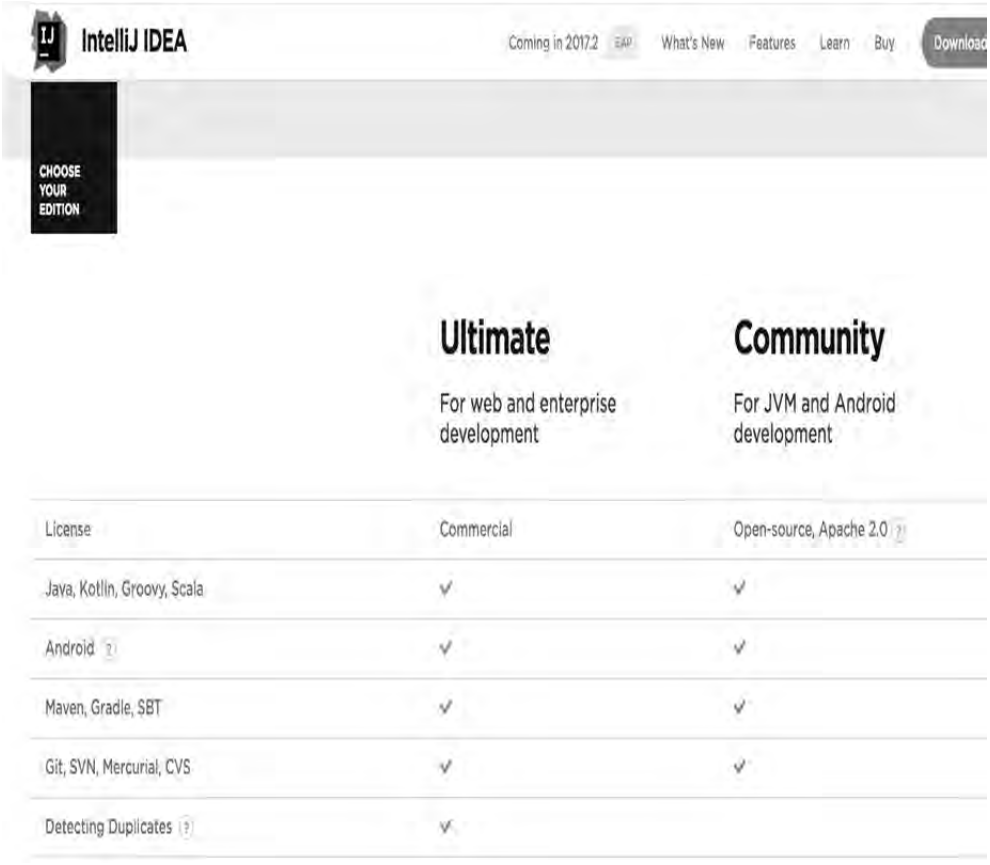
Estas son solo algunas de las características, hay otras como el tipado estático, distinciones de los nulos y otras que mencionaremos en su capítulo correspondiente más adelante.

Ambiente de trabajo

Ahora vamos a preparar todo lo necesario para comenzar a programar nuestros ejercicios, lo primero que vamos a hacer es abrir un navegador e ir a la página de JetBrains y específicamente a la sección del IDE Intellj IDEA:

www.jetbrains.com/idea/

En este sitio vamos a encontrar 2 opciones. La opción Ultimate y la opción Community, como se ve en la siguiente imagen:



Como podemos comprobar en el sitio web, la opción Ultimate es la de pago y tiene mucho más funciones y herramientas para el desarrollo, la opción Community es un poco más austera y que contiene solamente lo elemental para trabajar.

Nosotros utilizaremos para lo que hagamos en este libro la opción Community para que el lector pueda realizar todos los ejercicios sin ningún problema, la versión Ultimate tiene una versión de pruebas que puede ser también implementada pero transcurridos algunos días ya no puede usarse hasta realizar el pago.

Debajo de la tabla de comparaciones en donde se ven las características de las dos versiones del IDE podemos observar los botones de descargas que están en inglés, en cada caso nos detecta el sistema operativo de cada uno de ellos, en mi caso sabe que estoy trabajando con una computadora Mac Pro y es por eso que sugiere que descargue la versión para Apple, pero si el lector tiene Windows le hará la sugerencia de que descargue la versión propia de ese sistema operativo, simplemente le damos a descargar y listo, comienza la descarga.

En el momento de descargar el paquete lo único que tenemos que hacer es instalarlo como cualquier aplicación simple en nuestro sistema operativo, damos un doble clic y listo.

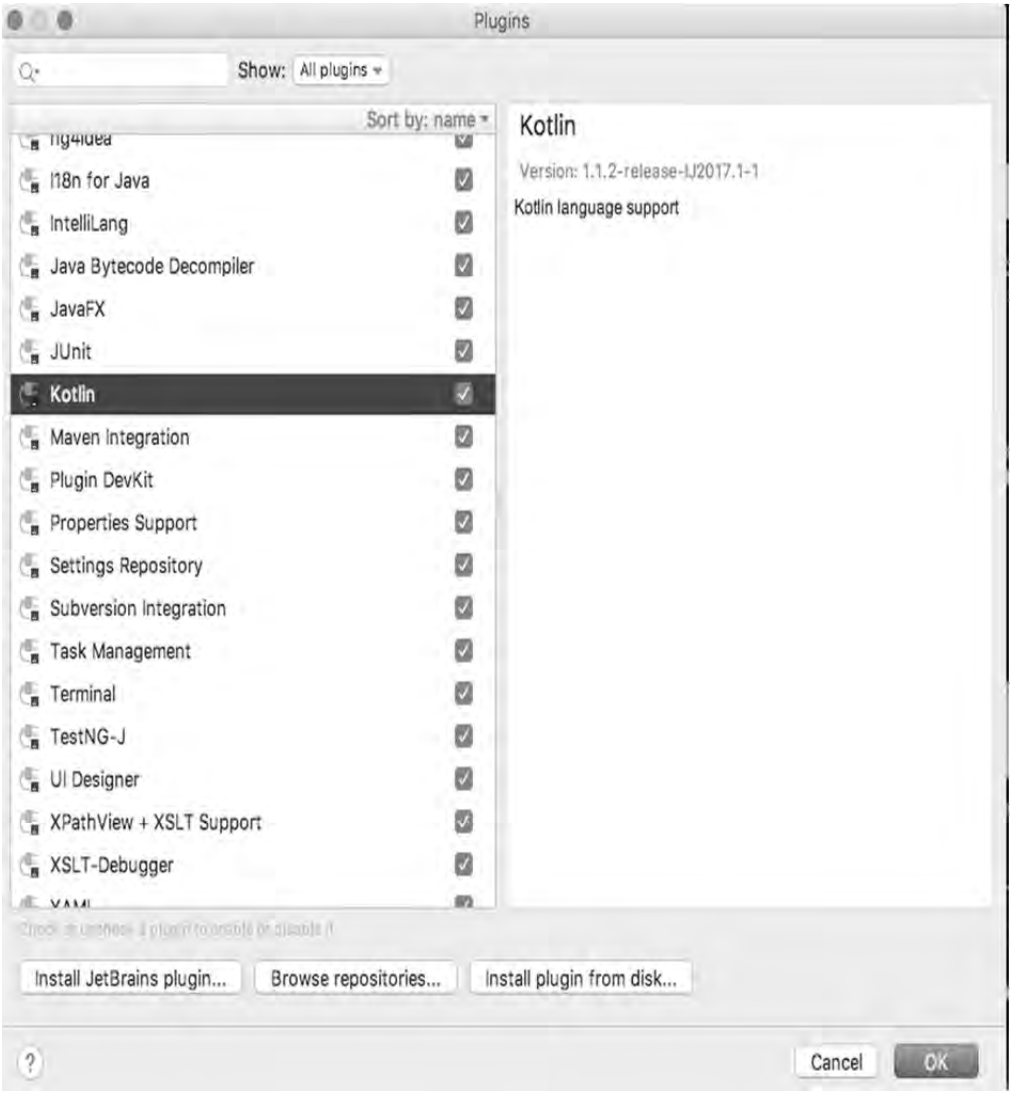
Ahora ya que tenemos descargado e instalado el IDE IntelliJ IDEA vamos a abrirlo y veremos la siguiente ventana:

Como podemos ver en la imagen anterior tenemos la ventana principal del IDE en donde podemos crear nuestro primer proyecto. La ventana se divide en dos, el lado derecho con el color gris de fondo y el lado blanco del lado izquierdo.

En el lado derecho vemos el menú principal y en el lado izquierdo los proyectos previos que se han creado, en este caso ya se cuenta con dos proyectos pero si nunca se ha utilizado un proyecto lo normal es que este apartado se encuentre vacío.

Ahora vamos a asegurarnos de que podemos programar utilizando Kotlin en este IDE. Iremos a la opción de configuración que se encuentra en el menú del lado inferior de la parte derecha de la pantalla y seleccionaremos “Plugins”, como se ve en la siguiente imagen:

En la ventana nueva que se acaba de abrir vemos una gran cantidad de plugins que podemos utilizar para desarrollar en este IDE, entre ellos se debe de encontrar Kotlin, habilitado como en la siguiente imagen:



En caso de que la opción no esté disponible o habilitada damos clic en el botón que dice “Install JetBrains plugin” para que el mismo IDE se encargue de instalar el plugin que es necesario para comenzar a utilizar el lenguaje de programación Kotlin.

Después de esto último finalmente podemos decir que ya tenemos el ambiente de desarrollo listo para comenzar a programar.

Crear un proyecto

Partimos de la ventana de inicio del IDE:

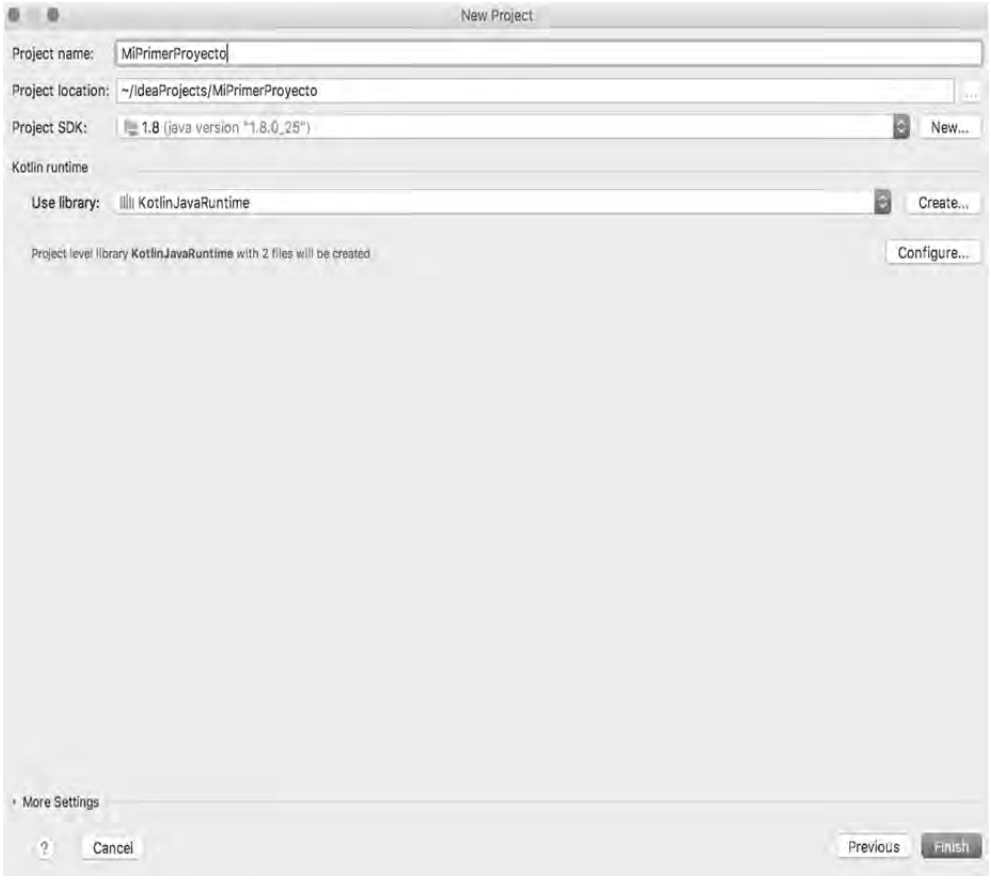
En esa primera ventana vamos a dar clic a la opción que dice “Create new Project” que en español dice “Crear proyecto nuevo”.

Ahora veremos una ventana como la siguiente:

La ventana se divide en dos partes, una columna en el lado izquierdo y un lado derecho que depende de la selección que tengamos en la columna.

En la columna izquierda vamos a seleccionar Kotlin como hemos hecho en la imagen anterior y después seleccionaremos la opción Kotlin (JVM) ya que esa opción es la que nos permite generar nuestro proyecto interoperable con Java; como podemos ver existe una segunda opción entre paréntesis (JavaScript) que genera la compatibilidad con el lenguaje JavaScript que ya hemos descrito pero que no utilizaremos.

Al seleccionar Kotlin (JVM) ahora veremos una ventana como la siguiente:



Tenemos las siguientes opciones:

Project name: en esta opción vamos a colocar el nombre del proyecto, puede ser el que lector quiera, por cuestiones prácticas se recomienda como en muchos de

los nombres de proyectos de software evitar caracteres especiales, números y ese tipo de reglas generales que se usan en informática.

Project location: en esta opción hay que seleccionar en qué ubicación se colocarán los archivos del proyecto, por defecto lo crea en el directorio de proyectos del IDE, pero siempre existe la posibilidad de que se lo coloque en cualquier directorio, por recomendación es buena idea no situarlo en un directorio de fácil acceso como escritorio o mis documentos ya que si el proyecto se elimina se puede perder todo el trabajo.

Project SDK: en este apartado vamos a ver la versión de Java que estamos utilizando, la versión 8 en este caso.

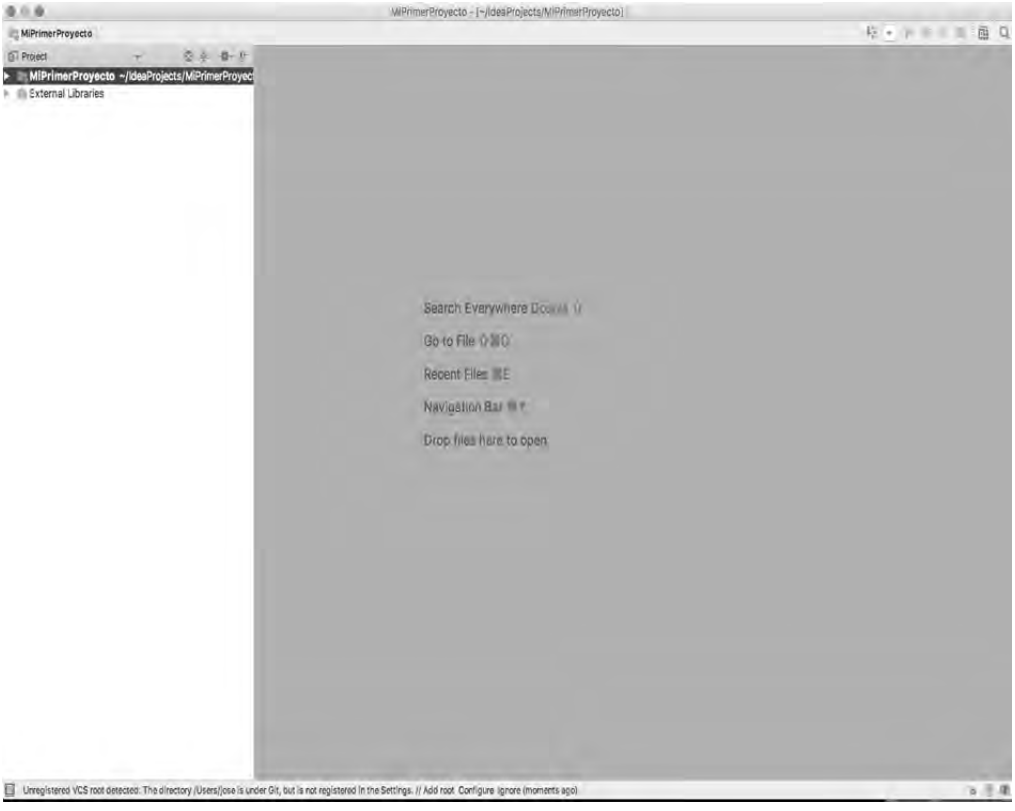
Ahora que ya sabemos qué significan esas opciones y en qué nos pueden ayudar podemos dar clic en Finalizar y listo, solamente tenemos que esperar unos segundos y ya hemos generado nuestro primer proyecto.

Tendremos una ventana como la siguiente que nos indicará que tenemos todo de forma correcta:

Hola mundo con Kotlin

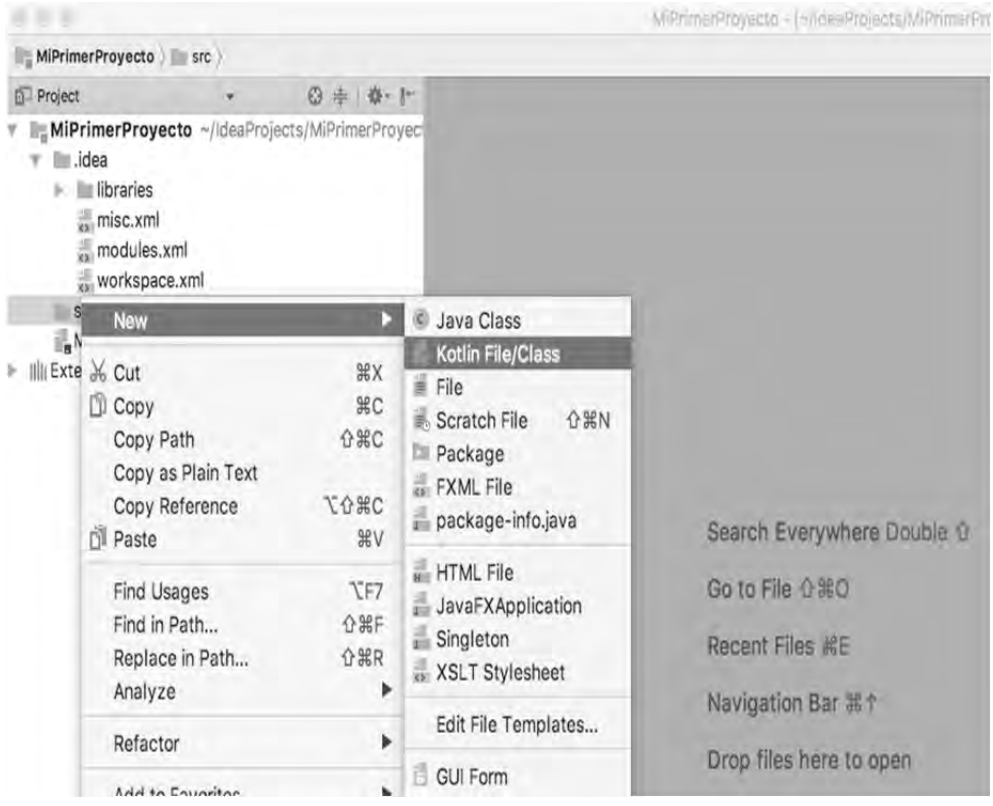
En este apartado vamos a crear un archivo que nos permita realizar nuestro primer ejemplo como en cualquier lenguaje de programación, el famoso “Hola mundo!”.

Comenzaremos colocándonos en la parte de la izquierda de la herramienta y damos clic al primer elemento que se llama MiPrimerProyecto, a continuación vemos cómo se despliegan unos directorios que se han creado en conjunto con el proyecto, se debe de ver como la siguiente imagen:



Ahora nos centraremos en el directorio que se llama “src”, en ese directorio vamos a tener todos los archivos correspondientes a la programación, en otras palabras estamos diciendo que los archivos que se escriban en Kotlin los vamos a colocar ahí. Cabe mencionar que en Java existe también este directorio “src”.

Damos clic derecho y vamos a crear un nuevo archivo, después seleccionamos la opción Kotlin, los pasos podemos verlos en la siguiente imagen:

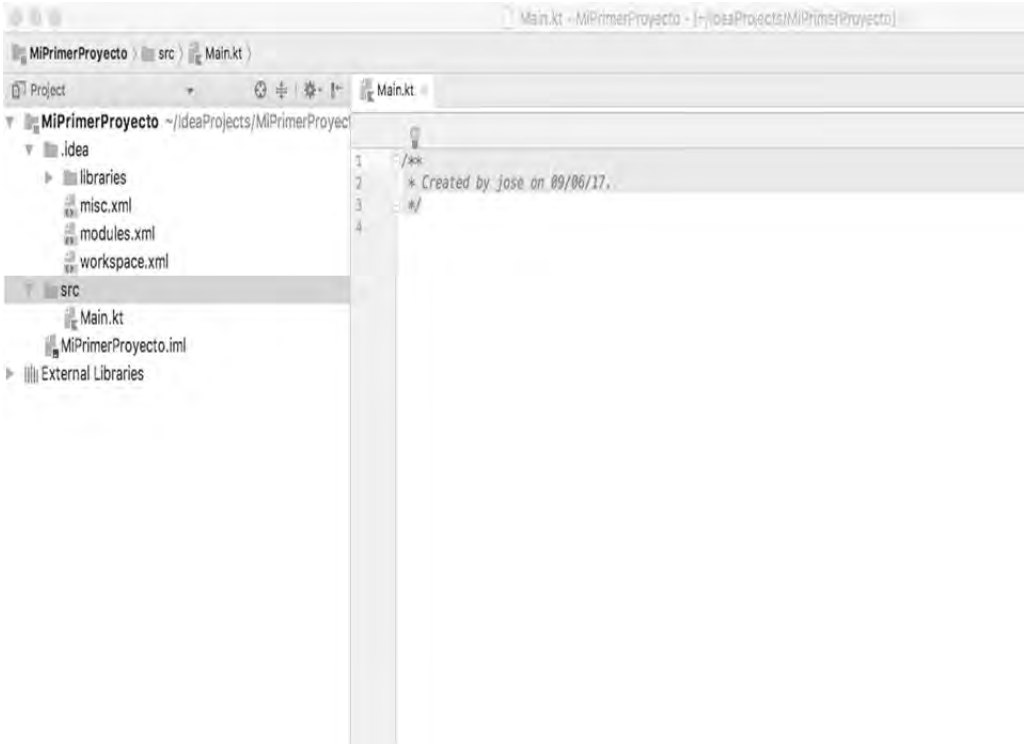


Con la opción de la imagen anterior estamos diciendo que en el directorio src vamos a crear un archivo que contendrá código en lenguaje Kotlin. Podemos ver una pequeña ventana que nos pide colocarle un nombre al archivo, el nombre que le podemos colocar por convención es “MAIN” que significa principal en español, esto se debe a que en la mayoría de los lenguajes de programación como Java, C, C++ siempre contamos con una clase principal o función que se encarga de ser la que dicta la lógica del programa, en este caso podemos seguir esta convención para no salir mucho del estándar.

Después de colocar el nombre podemos dar clic al botón OK. Si analizamos el cambio que acabamos de ver de la interfaz podemos notar lo siguiente: el directorio SRC ahora contiene un archivo que se llama Main.kt, la extensión kt es la que indica que este archivo contiene lenguaje Kotlin, y la parte derecha del IDE que antes

estaba vacía ahora contiene el archivo pero se muestra vacío, solo podemos ver en la parte superior un comentario que presenta el nombre de la cuenta que creó el archivo y la fecha.

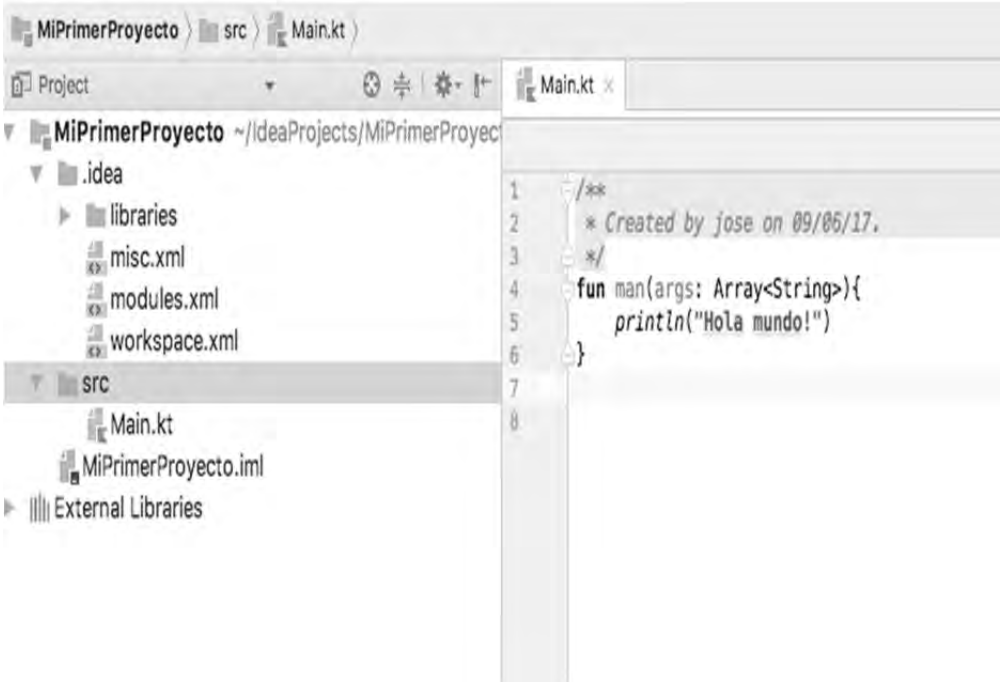
Podemos verlo en la siguiente imagen:



Por ahora este archivo no contiene código, nos corresponde a nosotros colocar la programación, comenzaremos poniendo un código que por ahora no vamos a explicar con detalle ya que tendremos un capítulo en donde explicaremos a fondo en qué consiste, el código es el siguiente:

```
fun man(args: Array<String>){
println(“Hola mundo!”)
}
```

Así se vería el código en el IDE:



La parte de código que no vamos a explicar es esta:

```
fun man(args: Array<String>){
}
```

Las líneas anteriores las vamos a tener que colocar ahora por defecto, es decir, que todos los programas y ejemplos que hagamos van a tener que llevarlo sí o sí, pero como ya se mencionó vamos a dedicarnos a estudiar a fondo qué es lo que sucede.

Por otro lado tenemos una línea, la línea 5, como aparece en la imagen que es donde mandamos a imprimir el “Hola mundo”:

```
println(“Hola mundo!”)
```

Esta línea lo que hace es mandar a imprimir la cadena de texto “Hola mundo!”.

Con esto podemos decir que hemos logrado nuestro objetivo de crear el famoso “Hola mundo!”, que es la instrucción que se ejecuta por defecto cuando se está conociendo un lenguaje de programación, ¡bienvenido a Kotlin!

Ejecutar un proyecto

Ya tenemos nuestras primeras instrucciones en el lenguaje de programación Kotlin, ahora nos falta ejecutar y ver si todo lo que hemos puesto está correcto, lo que vamos a hacer es dar clic en “Run” para poder ejecutar, lo podemos ver en la siguiente imagen: