ÍNDICE

PREFACIO

El desarrollo de aplicaciones multiplataforma (es decir, aplicaciones que puedan funcionar en diferentes sistemas operativos) es algo que hoy en día están buscando muchos desarrolladores y empresas.

Existen muchas opciones en el mercado que nos prometen lograr que nuestra aplicación pueda ser funcional para todos los sistemas operativos y que no tengamos que hacer una versión distinta para cada uno de ellos. La realidad es que todas las opciones en el mercado tienen ventajas y desventajas, unas más que otras. Pero hasta el momento ninguna ha logrado cumplir el desarrollo multiplataforma al 100%.

Flutter es un SDK que está logrando no solamente este 100% de éxito en las aplicaciones, sino que está trabajando en conjunto con la comunidad para ser la opción más viable y por ahora es la apuesta más rentable para cumplir el objetivo de programar una vez y poder hacer funcionar la aplicación en distintos sistemas operativos.

Para poder utilizar de la mejor forma Flutter, necesitamos trabajar con el lenguaje de programación Dart, un lenguaje de programación desarrollado por Google. Tanto Flutter como Dart pertenecen a esta compañía.

Es por esa razón que si desarrollamos con Flutter, encontraremos relacionado el lenguaje de programación Dart, ya que al trabajar con ambos se forma el equipo ideal que nos permite desarrollar aplicaciones tanto para Android como para iOS.

Dart es el primer paso entonces para cualquier desarrollador que esté interesado en el desarrollo multiplataforma con el SDK Flutter; por lo tanto, esta curva de aprendizaje es algo que debemos de superar si es que queremos dominar el desarrollo multiplataforma.

Dart es un lenguaje de los considerados modernos, como Kotlin, Swift y otros. Aunque es mucho más antiguo que estos dos, cuenta con elementos que son clave en el crecimiento de software actual para el desarrollo rápido y ágil de aplicaciones robustas.

El equipo de desarrollo Dart ha tomado elementos de mejora de otros lenguajes de programación y lo ha convertido en la ventaja de este lenguaje con la idea de facilitar la vida del desarrollador que decida utilizar este lenguaje de programación.

El libro pretende guiar al lector durante el proceso de dar los primeros pasos con Dart, para después abrir paso al aprendizaje de Flutter y llegar hasta el entendimiento de la estructura de una aplicación móvil con Flutter y la configuración completa para tener un paquete completo que funcione como aplicación para ambos sistemas operativos, Android e iOS.

Acerca del autor

El autor de este libro es un apasionado de la tecnología y la docencia. Comenzó en el mundo de la programación con el lenguaje Basic a los 13 años de edad. Colaborador habitual de comunidades en español sobre diversos temas: desarrollo de videojuegos, programación orientada a objetos, desarrollo web y dispositivos móviles.

José Dimas Luján Castillo nació en 1986, tiene el grado de Maestría en Tecnologías de Información. En la docencia ha colaborado con más de 14 universidades a nivel presencial en Latinoamérica en los niveles de Licenciatura y Maestría. En la educación en línea es colaborador de las plataformas más importantes a nivel mundial con más

de 250 cursos online en la actualidad; además de ser conferenciante habitual de eventos tecnológicos apoyando siempre la adopción de nuevas tecnologías.

Para que el lector pueda consultar y contactar con el autor, puede localizarlo en redes sociales con el alias josedlujan, twitter, Facebook, entre otras. En su web: www.josedlujan.com o por correo electrónico a josedlujan@gmail.com.

El código que vayamos utilizando lo podrás encontrar en el repositorio:

Https://github.com/josedlujan/libro_dart_flutter_aprende_a_crear_aplicaciones

Agradecimientos

En este libro está plasmado el esfuerzo de los últimos 5 años en los que me he dedicado a conocer el lenguaje de programación Dart y el SDK Flutter. Mi principal tarea siempre fue entenderlo a detalle para mantenerme como un desarrollador de aplicaciones móviles actualizado e interesado en las tecnologías del desarrollo móvil.

Este libro se lo quiero dedicar con mucho cariño y de una manera muy especial a mi esposa, Noemí, y a mis padres, José y Fabiola, que siempre me han apoyado durante todo este largo viaje dentro de la tecnología, y esta es la primera de muchas veces que le dedicaré algo con un gran cariño especial a Viena.

1 INTRODUCCIÓN

Desde la aparición de Dart los desarrolladores por todo el mundo estaban intrigados con la razón de su existencia, muchos se preguntaban para qué se había diseñado este lenguaje de programación, existían muchos rumores sobre cuáles podrían ser los objetivos. Muchas veces se mencionó a Dart como un claro sucesor de JAVA dentro del desarrollo Android, algo que no sonaba tan raro ya que Google y Oracle entablaron una demanda por varios años debido a esta tecnología.

Al pasar los años se dieron cuenta de que este motivo era cada vez más lejano, sobre todo en el momento en el que aparece Kotlin y se le nombra como un lenguaje oficial para el desarrollo móvil. En ese instante Dart pasó a segundo término ya que no era entonces la opción que todos esperaban para el desarrollo de aplicaciones Android.

Pasado un tiempo se presenta un SDK, que lleva el nombre de Flutter y tiene como principal objetivo el desarrollo de aplicaciones para varias plataformas, es entonces cuando se anuncia que el lenguaje de programación Dart es el lenguaje que tenemos que usar para desarrollar aplicaciones con este SDK. En ese momento Dart recobra fuerza y empieza a subir en popularidad, después de estar algunos años esperando a que la comunidad de desarrollo lo volviera a ver.

Historia

Dart fue anunciado en un evento de tecnología en el año 2011 como simplemente una de muchas de las tecnologías actuales, en ese instante se presentó solamente como un lenguaje de programación "moderno", demostrándose que podría funcionar en navegadores como una alternativa indirecta a JavaScript.

Dart, de manera directa, nos ofrece la posibilidad de generar código en JavaScript, puede funcionar sobre una máquina virtual con la que podemos ampliar el alcance de ejecución del lenguaje. La versión 1.0 se presentó oficialmente en el año 2013. Esta fue considerada la versión 1.0 del lenguaje.

A partir de este punto la historia ha sido muy veloz, los cambios que ha sufrido el lenguaje de programación y las nuevas implementaciones con tecnologías van a un paso rápido.

Cuando se lanza un lenguaje de programación, el primer elemento que buscamos es el editor con el que lo vamos a utilizar, en este caso DartEditor fue el editor con el que se nos presentó la idea de poder trabajar con Dart, podemos considerarlo como un editor ligero debido a que no exige gran cantidad de recursos para su funcionamiento. Las operaciones que tiene son las básicas, pero son más que suficiente para el uso del lenguaje.

Dart llamó la atención en un principio por ser de los pocos lenguajes de programación que seguían pasos similares a Java, es decir, que estaba implementando una máquina virtual, esto por supuesto permite que cualquier dispositivo que cuenta con una máquina virtual tendrá la capacidad de ejecutar las instrucciones, esta filosofía de trabajo es algo que se resalta siempre que se habla de este lenguaje de programación.

El rendimiento es algo que el equipo que tiene el control sobre Dart está cuidando mucho, cada vez que se hacen cambios considerables se resaltan los muy buenos números que pueden lograr tomando en cuenta el rendimiento y se hacen pruebas en las que se comparten los números con la comunidad.

En el momento del lanzamiento de Dart había elementos que eran novedosos en el mundo del desarrollo y que algunos lenguajes aún estaban considerando implementar pero que en realidad todavía no estaban disponibles. Algunos de ellos son, por ejemplo, las “promises”, variables de ámbito por bloque.

Todos estos elementos se podían comparar con los alcances de JavaScript y se veía que se estaba quedando corto ya que Dart había dado un paso adelante con estas características.

La sintaxis, ciertamente, podemos decir que ha sufrido cambios, pero la realidad es que tiene que ver mucho más con las mejoras por hacer de Dart un lenguaje más práctico y flexible, esto quiere decir que han logrado omitir algunas palabras y colocar como opcionales a otras para tratar de no ser repetitivo o redundante.

En un principio Dart contaba con un sitio web considerado como el oficial para que los desarrolladores interesados en él pudieran consultar la información, pero en la actualidad el sitio es otro:

www.dart.dev

En resumen, se puede decir con toda seguridad que Dart es una grata sorpresa dentro del desarrollo de software, pero sobre todo, en el desarrollo móvil, ya que esta alternativa, que es parte del SDK Flutter, es una herramienta que está dando un giro al desarrollo móvil.

En el momento que se está haciendo este libro, Dart se encuentra en su versión estable 2.9.0. Seguramente mencionaremos algunas versiones anteriores durante algunos apartados del libro debido a que explicaremos algunos cambios con las versiones anteriores, y sobre algunos detalles técnicos para compartir temas técnicos con el lector.

En este libro mencionaremos varias versiones para explicar algunos cambios. No nos limitaremos a hablar solamente de las versiones 2.18 y 2.12, sino que también consideraremos las versiones 2.0 y 1.0, pero solo con el objetivo de mostrar cómo ha evolucionado el software y los cambios que se han producido.

Si el lector cuenta con experiencia en lenguajes de programación como Java, JavaScript y Go, seguramente existan algunos elementos que le resultarán parecidos y que tendrán un rostro familiar, pero si el lector no tiene conocimientos previos de programación, puede basarse solamente en el libro y logrará cumplir la meta de superar la curva de aprendizaje, ya que el contenido tiene este objetivo.

Dart

Como mencionamos en párrafos anteriores, en el momento de escribir este libro Dart se encuentra en la versión 2.9.0, como su "Stable channel" y en la versión 2.10.0 como "Dev channel".

Un canal es una forma de ordenar la estabilidad de las versiones. Cuando hablamos de un canal dev: quiere decir que es una versión que ya fue completamente probada, de cierta manera podemos decir que es funcional. El canal estable, quiere decir de una manera muy directa que es la versión con la que se debe trabajar, esto es, que podemos confiar en la estabilidad de la misma y que no deberíamos tener mayor problema en el momento de lanzar proyectos utilizando esta versión como la base.

No importa el lenguaje de programación que utilicemos, en el juego de palabras con el que se define el nivel estabilidad de la versión queda claro que siempre es mejor utilizar la versión que es considerada como "estable", eso nos asegura un mínimo de calidad y seguridad en el momento de trabajar.

Para conocer más sobre Dart, tendremos que ver una pregunta habitual que se hace un desarrollador cuando comienza a utilizar un lenguaje de programación:

¿El lenguaje X (En este caso Dart) es compilado o interpretado?

La realidad es que depende, a diferencia de muchos lenguajes en donde la respuesta es algo que se puede decir de memoria y señalarlo como la respuesta, en este caso es más complejo. Si tenemos que dar una respuesta corta podemos decir que es un lenguaje con ambas opciones, puede ser compilado e interpretado.

Esta pregunta nos lleva a hablar de una de las similitudes en la que podemos notar el parecido con Java, pues este es un lenguaje que es compilado e interpretado. Ahora vamos a describir un flujo normal de trabajo. Cuando escribimos código en el lenguaje de programación Dart, lo que sucede es que tenemos un compilador que se da a la tarea de trabajar con el código que generamos, en ese momento se genera un "resultado". Este "resultado" necesita ser pasado a una máquina virtual que va a ser la encargada de leer y entregar el producto final. En este proceso se observa de manera clara que Dart es un lenguaje compilado e interpretado.

Esta respuesta es muy simple, ya que en realidad sí tenemos que extender podemos decir que Dart puede ser compilado, pero no solamente eso, depende si queremos hacerlo como AOT o JIT.

- AOT significa ahead-of-time
- JIT significa Just-in-Time.

Es posible que esto no nos diga mucho por ahora, pero vamos a explicarlo para el lector. Dentro de la compilación podemos decir que tenemos estas dos grandes opciones, existen compiladores JIT y AOT. Cuando hablamos de compiladores JIT, el compilador se ejecuta durante la ejecución del programa, en otras palabras se compila sobre la marcha. Y al hablar de compiladores AOT, existen los compiladores que se ejecutan durante la creación de la programación, esto sería antes del considerado tiempo de ejecución.

Además, sabemos que puede ser interpretado, es por eso que podemos decir que Dart sí es un lenguaje distinto, ya que en realidad todo lo que acabamos de mencionar no es algo habitual, eso hace que se considere como un lenguaje muy flexible. Al final no es lo más importante cuando quieres aprender Dart ya que sin el conocimiento previo te puede confundir, pero en algunos proyectos puede que sea muy buena idea estudiarlo, sobre todo para sacar el máximo provecho a las diversas situaciones que podemos encontrarnos.