

Trabajo de Investigación Formativa

Aplicación de un Lenguaje de programación con el uso de Dart

Autores:

Italo Fabricio Bertrand, Velita Aguilar CUI: 20190052 15%

Daniel Edward, Tapia Saenz CUI: 20213034 30%

André Renzo Añazco Huamanquispe CUI: 20211778 15%

Juan Carlos, Llayqui Huillca CUI:20210560 20%

Josué Mathías Miguel Cahuana Aguilar CUI:20210553 20%

Docente:

Richart Smith Escobedo Quispe



Escuela Profesional de Ingeniería de Sistema

Facultad de Producción y Servicios

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

Arequipa – Perú

2021

INTRODUCCIÓN

Los humanos inventaron las computadoras hace menos de 100 años y las primeras de ellas eran enormes y pesaban más de 1 tonelada y solo era capaz de realizar 4 operaciones, desde ese entonces hemos mejorado y optimizado el aspecto, rendimiento y muchos otros aspectos de las computadoras. A su vez hemos creado muchos lenguajes de programación de diversos tipos, actualmente el número de lenguajes de programación creados rondan por los 9000 y se crearán muchos más. Sin embargo muchos de estos lenguajes existentes no son usados por muchas personas o por el contrario, hay algunos lenguajes de programación los cuales son usados por mucha gente, tal es el caso de Python, Java, JavaScript, C#, etc. En este trabajo de investigación formativa hablaremos del lenguaje de programación Dart el cual fue lanzado el 14 de Noviembre del 2013. Y desde entonces ha ganado popularidad teniendo una audiencia de hasta 800.000 personas y unos 10.000 grupos siguiendo las novedades.

Dart está pensado para resolver los problemas de JavaScript y ofrecer unos mejores resultados, pese a que en su presentación no se hicieron referencias explícitas a JavaScript. Tiene la capacidad de ser una herramienta sencilla para proyectos más grandes y ofrece una mejor seguridad. Los ingenieros de Google desarrollaron desde 2011 un programa desde Internet para el desarrollo de aplicaciones llamado Brightly, que quizás podrá ser la primera aplicación creada con Dart. Google ofrecerá una herramienta para compilar Dart a ECMAScript 3 en el instante, para ser compatible con los navegadores que no utilicen Dart. Google también integrará una máquina virtual nativa en Google Chrome y facilitará a sus competidores lo mismo para sus navegadores.

Una cosa que esperamos al aprender este nuevo idioma de programación y al hacer ejercicios de dificultades variadas es poder aprender algo nuevo y saber usar correctamente este nuevo lenguaje de programación. Y nuestras motivaciones es aprender este nuevo lenguaje de programación que algunos ni conocíamos antes y tener un nivel medio al usar este lenguaje

CARACTERÍSTICAS DEL LENGUAJE ESCOGIDO

DART:

Dart es un lenguaje de programación creado por Google que tiene como objetivo el desarrollo de aplicaciones web complejas de forma sencilla y eficiente. Inició como una alternativa para Javascript en 2011. Con Dart diseñar aplicaciones móviles y para escritorio son posibles, aún con un nivel de entendimiento básico sobre cómo programar. Además de scripts y realizar programación Backend (programación de la lógica de un sitio web).

Dart se diseñó de forma muy similar a Javascript, C, C + +, por lo que adopta una sintaxis semejante a estos lenguajes, pero de forma simplificada. También se asemeja por paradigma de **programación orientada a objetos (POO)**, cosa que lo familiariza aún más con los lenguajes antes mencionados. Aunque para diferenciarse incluye un conjunto de herramientas pre integrado, como su propio gestor de paquetes, varios compiladores/transpiladores, un analizador y formateador. Dart también cuenta con una característica que facilita el desarrollo de aplicaciones con conexión a internet, pues su máquina virtual *Just in Time*, permite que los cambios realizados al código se vean reflejados de forma casi instantánea.

Una vez en producción, el código se puede compilar en lenguaje nativo, por lo que no es necesario un entorno especial para ejecutarlo. En caso de que se haga desarrollo web, Dart

se transpila a JavaScript. Además, Dart consta de un gran apoyo para la asincronía, y trabajar con generadores e iterables es extremadamente sencillo.

UTILIDAD:

Dart es un lenguaje de propósito general, y lo puedes utilizar casi para cualquier cosa:

¿Por qué comenzar a usar dart?

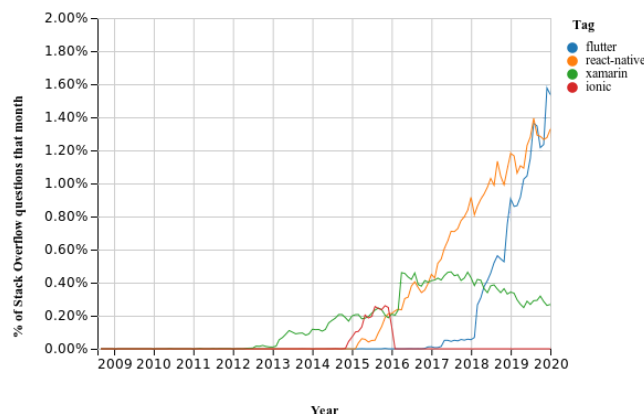
- Es compatible con javascript.
- Es usado por flutter.
- Es muy eficiente (puesto que usa AOT y JIT).
- Tiene muchos IDEs y editores compatibles.
- Se pueden crear aplicaciones para cualquier plataforma.
- Es Open Source.
- Tiene tipado opcional.
- Está orientado a objetos.
- Es fácil de aprender.

Como Inconvenientes

- Dart es un lenguaje de programación relativamente nuevo, por lo que su comunidad no es de gran extensión.
- Sus librerías son limitadas a comparación de otros lenguajes por el mismo inconveniente que su comunidad.
- Su enfoque es mayor a aplicativos móviles, pero tiene capacidad para desarrollar aplicativos de escritorio.

¿Qué es flutter?

Flutter es un *framework* de Dart para crear aplicaciones multiplataforma con un único código. A diferencia de otros *frameworks* multiplataforma como por ejemplo Ionic, el código de una aplicación de Flutter se compila a código nativo, por lo que el rendimiento alcanzado es superior a aplicaciones basadas en *web-views*. Además, a diferencia de React Native, Flutter no utiliza componentes nativos, sino que viene con sus propios componentes, llamados *widgets*, por lo que la misma aplicación se verá igual en cualquier dispositivo, independientemente de su sistema operativo o la versión. Gracias a ello, el desarrollador no tiene que preocuparse por que el diseño de su aplicación se vea mal en dispositivos antiguos.



Hola mundo

Para iniciar cada programa en Dart debe tener una función llamada ("main"), ya que de esta forma le dice cuál es la entrada para la ejecución de dicho programa, escribimos ("print") dentro del mismo por medio de comillas simples o dobles escribimos ("hola mundo").



```
1 void main()
2 {
3   print('Hola mundo');
4 }
```

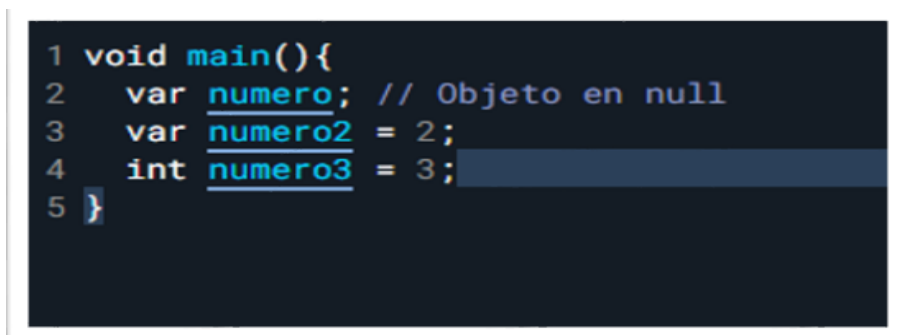
Run

Console

Hola mundo

Variables

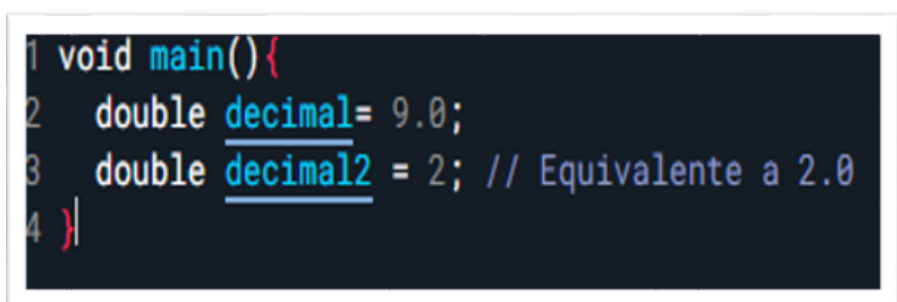
Como vemos en el ejemplo de abajo basta con utilizar la palabra var y el nombre de la variable, o si lo prefieres, indicar el nombre del tipo seguido del nombre de la variable.



```
1 void main(){
2   var numero; // Objeto en null
3   var numero2 = 2;
4   int numero3 = 3;
5 }
```

-De punto flotante

No podemos utilizar explícitamente el tipo float, en su lugar utilizamos el tipo double. A los objetos double se les puede asignar un valor entero, es decir, no hay necesidad de agregar el punto y los valores decimales forzosamente.



```
1 void main(){
2   double decimal= 9.0;
3   double decimal2 = 2; // Equivalente a 2.0
4 }
```

-Booleanos

Solo se admiten los valores de true y false como valores booleanos.

```
1 void main(){
2     bool verdadero = true;
3     bool falso = false;
4 }
```

-Cadenas de texto

Podemos utilizar comillas dobles “ o comillas simples ‘ para asignar una cadena de texto.

```
1 void main(){
2     String saludo = "Hola mundo";
3     String saludo2 = 'Hola mundo';
4 }
```

-Dynamic

Estas variables pueden tomar cualquier tipo de dato y cambiarlo a cualquier otro en todo momento.

```
1 void main(){
2     dynamic valor = 12;
3     valor = "Cadena de texto";
4     valor = 3.1416;
5     valor = true;
6 }
```

Estructuras de control

La estructura If la podemos definir de la misma manera que en otros lenguajes de programación.

```
1 void main(){
2     var edad = 18;
3     if(edad <= 18){
4         print('adulto');
5     }else{
6         print('niño');
7     }
8 }
```

Ciclo For

En Dart el ciclo for tiene la siguiente forma.

Si queremos recorrer una lista definida podemos usar el siguiente código.

```
1 void main(){
2     for(int i = 0; i<=10; i++){
3         //Si i es par
4         if(i.isEven){
5             print(i);
6         }
7     }
8 }
```

Ciclo While

En Dart al igual que en la mayoría de lenguajes de programación podemos hacer uso de ciclos while y do while. En ambos casos el ciclo continua hasta que la condición indicada sea falsa.

```
1 void main(){
2     // while
3     int i = 0;
4     while(i<=10){
5         print('Valor actual $i');
6         i++;
7     }
8     // do while
9     int i = 10;
10    do{
11        print('Valor actual es $i');
12        i--;
13    }while(i>=0);
14 }
```

<https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart>

Ejercicios

Crear un programa que halle si un número entero positivo es compuesto o primo - básico

```
main.dart ×
1  import 'dart:io';
2  import 'dart:math';
3  void main() {
4      double divisores = 0, a = 1;
5      print('Ingese el número entero positivo a evaluar:');
6      String numero = stdin.readLineSync();
7
8      if(int.parse(numero)==1||int.parse(numero)==0){
9          print("su número no pertenece a los primos o compuestos ") ;
10     }
11     else{
12         for (a = 1; a <= sqrt(int.parse(numero)); a++) {
13             if (int.parse(numero) % a == 0) {
14                 divisores++;
15             }
16         }
17         if (divisores == 1) {
18             print("Su número $numero es primo");
19         }
20         else {
21             print("Su número $numero es compuesto");
22         }
23     }
24 }
25
```

```
❖ run-project
Ingese el número entero positivo a evaluar:
653
Su número 653 es primo
❖
```

Crea un programa que halle el factorial de un entero positivo usando un Método

main.dart

```
1  import 'dart:io';
2
3  int Factorial(int x){
4      int factorial=1;
5      int a=1;
6      for(a=2; a<=x;a++){
7          factorial=factorial*a;
8      }
9      return factorial;
10 }
11
12 void main() {
13     print('Ingese el entero positivo a evaluar:');
14     String numero = stdin.readLineSync();int num=int.parse(numero);
15     int factorial=1;
16     if(num==1||num==0){
17         print("factorial es 1") ;
18     }
19     else{
20         print("El factorial de su número $numero es ${Factorial(num)}");
21     }
22 }
23
```

> run-project

Ingese el entero positivo a evaluar:

11

El factorial de su número 11 es 39916800

>

Crear un programa que halle la distancia entre dos puntos en el plano cartesiano - **Básico**
Código:

```
1  import 'dart:io';
2  import 'dart:math';
3
4  void main() {
5      //Variables
6      print('Programa que calcula la distancia de 2 puntos');
7      print('Ingrese abcisa del primer punto: '); //x1
8      String x1s= stdin.readLineSync(); int x1 = int.parse(x1s);
9      print('Ingrese ordenada del primer punto: '); //y1
10     String y1s= stdin.readLineSync(); int y1 = int.parse(y1s);
11     print('Ingrese abcisa del segundo punto: '); //x2
12     String x2s= stdin.readLineSync(); int x2 = int.parse(x2s);
13     print('Ingrese ordenada del segundo punto: '); //y2
14     String y2s= stdin.readLineSync(); int y2 = int.parse(y2s);
15     //Calculando distancia
16     double distancia = sqrt(pow(x2-x1, 2)+pow(y2-y1, 2));
17     //Imprimiendo resultado
18     print('La distancia entre ambos puntos es '); print(distancia);
19 }
```

Consola:

```
❏ run-project
Programa que calcula la distancia de 2 puntos
Ingrese abcisa del primer punto:
0
Ingrese ordenada del primer punto:
0
Ingrese abcisa del segundo punto:
3
Ingrese ordenada del segundo punto:
4
La distancia entre ambos puntos es
5.0
```

Crear un programa que imprima una tabla de multiplicar donde se ingrese un valor entero para el multiplicando y un rango de “m “a “n “ que serán los multiplicadores, imprimir la tabla con todos los valores (incluyendo el signo de multiplicación y el de igual) - **Intermedio**

Código:

```
1  import 'dart:io';
2
3  void main() {
4      print('Tabla de multiplicar personalizada para un valor');
5      //Variables
6      print('Ingrese el multiplicando');
7      String num = stdin.readLineSync(); int num1= int.parse(num);
8      print('Ingresar rango de multiplicadores');
9      print('mínimo'); //Mínimo m
10     String m = stdin.readLineSync(); int m1= int.parse(m);
11     print('máximo'); //Máximo n
12     String n = stdin.readLineSync(); int n1=int.parse(n);
13     //Imprimir Tabla
14     print('Tabla ');
15     for(; m1<=n1; m1++){
16         print('${num1} X ${m1} = ${num1*m1} ');
17     }
18 }
```

Consola:

```
❏ run-project
Tabla de multiplicar personalizada para un valor
Ingrese el multiplicando
12
Ingresar rango de multiplicadores
mínimo
30
máximo
40
Tabla
12 X 30 = 360
12 X 31 = 372
12 X 32 = 384
12 X 33 = 396
12 X 34 = 408
12 X 35 = 420
12 X 36 = 432
12 X 37 = 444
12 X 38 = 456
12 X 39 = 468
12 X 40 = 480
```

DESCRIPCIÓN DE LA APLICACIÓN

Nuestro programa busca simular un Sistema Estudiantil basado en listados generales de Alumnos, aulas (en este caso son el año cursado) y las notas de cada estudiante. Para la aplicación generamos un map donde colocaremos la lista general de todos los estudiantes que queramos insertar separándolos por año, incluyendo dentro un array con las notas de cada estudiante (evaluamos para este programa 3 notas, Matemáticas - Fundamentos de Programación - Introducción a la Computación). El lenguaje nos permite asignar dentro del map, keys (llaves) que pueden ser tanto String (que sería el año cursado) como enteros o doubles (el orden de cada alumno dentro de las aulas) hay que tener en cuenta que en dart al igual que la mayoría de lenguajes tienen como primer orden el 0 por lo tanto el último elemento estaría en la posición n-1.

Luego de establecer nuestro map con los datos de las aulas y estudiantes, requerimos crear un objeto que pueda enlazar las funciones principales con el map, a nuestro objeto lo llamaremos "opciones" y aquí crearemos los métodos de búsqueda para consultar aulas, notas y estudiantes, otra ventaja que tiene dart es que el compilador prioriza el método main sobre las demás funciones así que podemos incluir estas funciones de manera simultánea al método principal.

Para poner a prueba nuestro programa debemos crear un método principal que pueda llamar a nuestro objeto y poder lanzarlo, primero creamos un objeto de la Clase "opciones" al que llamaremos "nuevo", luego en nuestro caso asignaremos una leyenda donde indicaremos las opciones disponibles de nuestro objeto, con enteros para un mejor entendimiento: (1) para consultar notas, (2) para listado general, (3) para ver aulas y (4) para poder terminar el programa.

Generamos un bucle donde nos pida ingresar los enteros correspondientes a la opción requerida y cuando ingresemos el valor, mediante estructuras de control de condicionales (elegimos Switch-case debido a que solo queremos analizar un entero) dependiendo del valor que tome nuestro objeto "nuevo" llamará un método que asignamos en la Clase "opciones". Si nuestra selección en algún momento del bucle llega a ser (4) una variable booleana que asignamos como verdadera para mantener activo nuestro bucle pasará a tomar un valor falso y de esa forma terminará nuestro programa.

El mantenimiento que requiere nuestro programa se basaría básicamente una modificación en nuestro map agregando o quitando aulas o alumnos y si queremos mas opciones solo modificamos la Clase "opciones" y agregamos un nuevo valor para verificar dicha opción en el main, a pesar de ser sencillo nuestro programa tiene un potencial para poder complejizarse si nos tomamos el tiempo para poder profundizar en este lenguaje.

CONCLUSIÓN:

Dart parece ser el primer lenguaje versátil, en el sentido de que se puede usar en todos los grandes sistemas operativos de escritorio, en todas las plataformas móviles y web modernas, y en realidad, puede ofrecer buenas interfaces de usuario, Dart es completamente de código abierto y todas sus herramientas también.

Para muchos de nosotros, Dart se convertirá en el próximo estándar de programación y revolucionará el desarrollo web, marcando un antes y un después en el modo de estructurar y desarrollar aplicaciones web, este podría ser un buen momento para voltear a ver este lenguaje que poco a poco está tomando fuerza.

Dart es el lenguaje de programación preferido por Flutter, por lo que podemos usar todo lo que sabemos para empezar a programar aplicaciones nativas de iOS o Android, pero recordemos que Dart no es solo para aplicaciones móviles, podemos crear aplicaciones de servidor e incluso de escritorio con el, por lo tanto este lenguaje nos permitirá construir proyectos e ideas de una manera satisfactoria.