



GDG Sucre

Lo que tienes que saber
de Dart para Backend,
Frontend y Mobile.

Quien soy yo



Sergio Antonio Ochoa Martinez

Developer OpenKM

Developer Freelancer

gnu.java.sergio@gmail.com

sochoa@openkm.com





Dart

Historia

- 2011: Se dio a conocer antes que sea estable
- Alternativa: Es una alternativa a Javascript
- Google: Es el desarrollador principal
- Maquina virtual: Google Chrome tiene una maquina virtual incluida para trabajar con dart
- 2013: Salio la versión estable la version 1.0

Actualidad

- 2.x: Utilizar la versión 2.0 de Dart es la mejor opción para tener las nuevas actualizaciones que tendrá
- SDK: Kit de desarrollo
- Chromium: Implementa la VM de Dart

Porque Dart?

- Los desarrolladores de Google y otros sitios utilizan Dart para crear aplicaciones de alta calidad y misión crítica para iOS, Android y la web. Con características dirigidas al desarrollo del lado del cliente.
- Dart es ideal para aplicaciones móviles y web.

Características

- Dart fue creada no para reemplazar javascript sino para tener una opción más moderna en la web
- Dart es free y opensource <https://www.dart.dev/>
- Dart tiene alguna similitud o fue influenciado por los siguientes lenguajes de programación
 - Java
 - Go
 - Javascript
 - C

Características(cont.)

- Es un lenguaje Orientado a Objetos
- JIT Y AOT
 - AOT (Ahead Of Time): Compilado a un rápido y predecible código nativo. Totalmente personalizable
 - Puede se JIT (Just In Time): Compilado para una velocidad excepcional de desarrollo. (Esto incluye el popular ****Hot Reload****)
- Hace fácil la creación de animaciones y transiciones que corren a 60fps(frames por segundo).

Características(cont.)

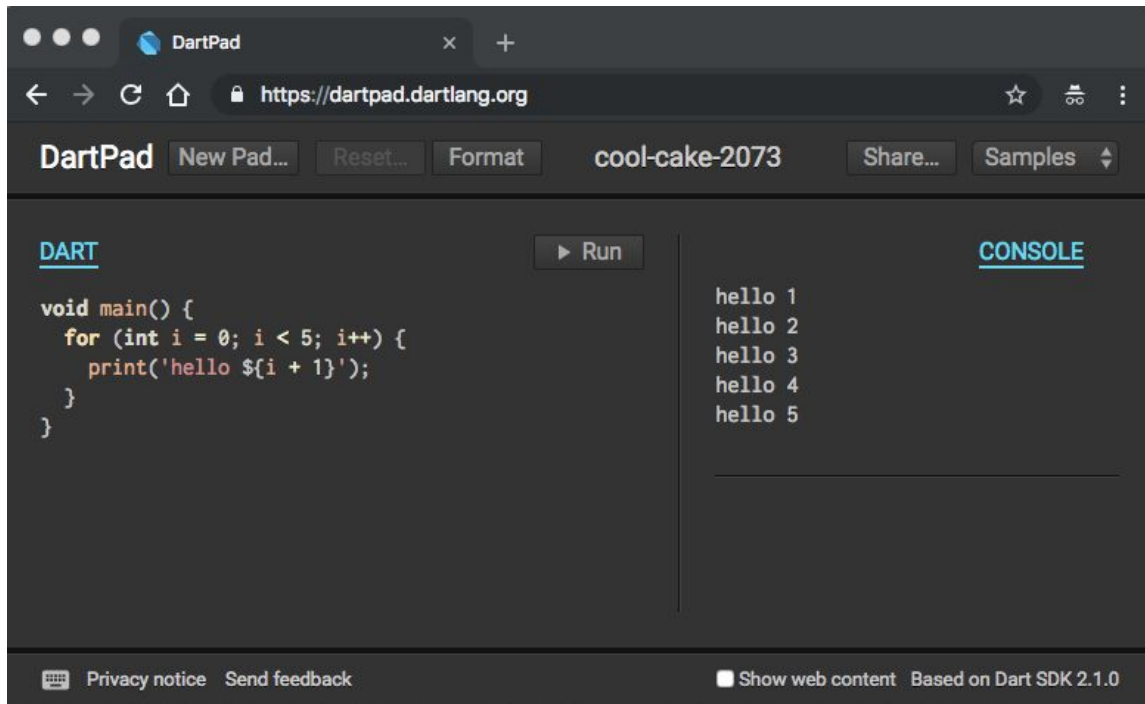
- Al ser compilado a código nativo, no hay puentes innecesarios para correr el código.
- Dart le permite a Flutter evitar el desarrollo de diseños en archivos independientes como JSX, XML o bien interfaces separadas.
- Dart es relativamente fácil de aprender.

Herramientas

(La estándar y las de terceros)

Dartpad

- Editor online



<https://dart.dev/tools/dartpad>

IDEs y Editores



Android Studio



IntelliJ IDEA

(and other JetBrains IDEs)



Emacs



Vim



Visual Studio Code



Atom



Eclipse

Gestión de projects

(building)

Project lifecycle

Java



Declarative XML Logic



Gradle

Declarative and imperative Groovy

Node



Scripts

Bash scripts & JavaScript tools



Imperative JavaScript

Pub <https://dart.dev/tools/pub/cmd>

- Administrar dependencias de paquetes
- Ejecutar aplicaciones de línea de comandos
- Implementar paquetes y aplicaciones

Pub comandos

- `pub cache`
- `pub deps`
- `pub downgrade`
- `pub get`
- `pub global`
- `pub publish`
- `pub run`
- `pub upgrade`
- `pub uploader`

Lenguaje

(La estándar y las de terceros)

Variables

Variable: Espacios en memoria para almacenar un valor

- Variables sin tipo
 - var
- Variable con tipo
 - int
 - double
 - String
 - bool
- Variables dinamicas
 - dynamic

Tipos de datos

- Los tipos de datos en Dart son Objetos
- Valor inicial de los tipos de datos sera null.
 - num
 - int
 - double
 - String
 - bool
 - Lists(Array) => Colecciones
 - Maps(Dictionary) => Colecciones

Operadores

Operadores Aritmeticos

- Suma `+`
- Resta `-`
- Multiplicación `*`
- División `/`
- Division `~/` , este devuelve un entero
- Modulo `%` es el restante de una división

Operadores de igualdad y relacionales

- `==` igualdad
- `!=` No igual o diferente
- `<` menor que
- `>` mayor que
- `<=` menor igual que
- `>=` mayor igual que

Operadores Logicos

- `!expr` Realiza una negación en la expresión
- `&&` AND
- `||` OR

Operadores

Operadores de prueba de tipo

- Útiles para verificar los tipos en tiempo de ejecución
- `as` Typecast (También utilizado para especificar prefijos de biblioteca)
- `is` Verdadero si el objeto es el tipo especificado
- `is!` Falso si el objeto tiene el tipo especificado

Operadores de Asignación

- `??=` Para asignar sólo si la variable asignada es nula
- `=` Asignar valor
- `+=`
- `-=`
- `*=`
- `/=`
- `%=`
- `~/=`

Operadores

Operadores de incremento y decremento

- `++i` Pre incremento
- `i++` Post incremento
- `--i` Pre decremento
- `i--` Post decremento

› Operador Ternario

- El operador ternario si lo utilizo en java, php, javascript ya podra utilizarlo en dart ya que se lo utiliza de la misma forma.
- `condition ? (statement if true) : (statement if false);`

Estructuras

if-else

- Esta estructura de control es igual a java, c++, c#, JavaScript.
- Las sintaxis basico para utilizar if-else:

```
if(condicion){  
    //Codigo al cumplir condicion  
}else{  
    //Codigo al No cumplirse  
}
```

- If-else en escalera

```
if(condicion1){  
    // codigo  
} else if(condicion2){  
    //codigo  
} else if(condicion3){  
    //codigo  
} else {  
    // codigo  
}
```

Estructuras

Expresiones condicionales

- `??` Operador para evaluar si la expresion es null
 - `expr1 ?? expr2`
- Operador ternario
 - `condicion ? value1 : value2`

Switch/Case

```
switch(valor) {  
    case expresion1: {  
        // acciones;  
    }  
    break;  
    case expresion2: {  
        //acciones;  
    }  
    break;  
  
    default: {  
        //acciones si no aplican las demás;  
    }  
    break;  
}
```


Estructuras Repetitivas

For/For...in/foreach

```
for(inicializador, condicion, incrementar/decrementar){  
    //codigo  
}
```

- Estructura del ciclo **for...in** es:

```
List lista = ['a','b','c','d'];  
for(Tipo nombreVariable in lista){  
    print(nombreVariable);  
}
```

Estructuras repetitivas

while/do while

```
while(condicion) {  
    //codigo  
}
```

- La estructura del ciclo **do while** es:

```
do {  
  
} while(condicion);
```

Funciones

- Funciones main
- Funciones
- Funciones con parametros opcionales
- Funciones con parametros nombrados
- Funciones con parentros por defecto
- Funciones Arrow => Fat Arrow
- Funciones anonimas(Lambda)

Programación Orientada a Objetos

- Classes

```
class Portatil {  
    // Características o Atributos  
    String modelo;  
    String marca;  
    double peso;  
    double velocidad;  
    double almacenamiento;  
  
    // Funcionalidades o metodos  
    void programar() {  
  
    }  
  
    void jugar() {  
  
    }  
}
```

Objetos

```
Portatil hpPortatil = new Portatil();  
Portatil lenovoPortatil = Prtatil();
```

POO

- Métodos en una clase
 - Metodos con parametros obligatorios.
 - Metodos con parametros opcionales posicionales.
 - Metodos con parametros opcionales nombrados.
 - Métodos con valores por defecto.
 - Métodos con funciones arrow.
 - Metodos con parametros como función.

POO

- Constructores
- Set y Get
- Herencia
- Sobreescritura de métodos
- Clases Abstractas y métodos abstractos
- Interfaces implícitas
- Metodos y atributos estáticos

Conceptos avanzados

- Colecciones
 - List y Set
 - Map
- Importar librerías propias y de terceros
- Excepciones
- Mixis
- Futures
- Async Await

Stagehand

stagehand

- Stagehand es un scaffolding para generar proyectos en dart, herramienta inspirada en yeoman y start kit.

Stagehand template

- console-full - A command-line application sample.
- package-simple - A starting point for Dart libraries or applications.
- server-shelf - A web server built using the shelf package.
- web-angular - A web app with material design components.
- web-simple - A web app that uses only core Dart libraries.
- web-stagexl - A starting point for 2D animation and games.

Comandos stagehand

- Instalar y actualizar
 - `pub global activate stagehand`

Usar Stagehand

```
$ mkdir fancy_project  
$ cd fancy_project  
$ stagehand package-simple
```

O tambien un proyecto de consola

```
$ mkdir fancy_project  
$ cd fancy_project  
$ stagehand console-full
```

Backend

Backend

- **Aqueduct**
- **Jaguar**
 - MVC
 - ORM
 - Auth
 - Security
- **Angel**
 - Framework Completo (Testing, Rest, DI, MVC, Command Line)
 - Documentación bastante explicada
- **Redstone**
 - Micro-Framework



AQUEDUCT

Aqueduct

- Multi-threaded
- Routing
- ORM
- Database Migration
- OAuth 2.0 Server
- Test Library
- TravisCI - Integración Continua
- OpenAPI 3
 - Generar documentacion de tu API REST
- Logging

Frontend

Frontend

- AngularDart - <https://angulardart.dev/>
- MDL/Dart - <http://www.material-design-lite.pub/>
- OverReact -
https://workiva.github.io/over_react/
- VueDart - <https://refi64.com/vuedart/>



AngularDart - <https://angulardart.dev/guide/setup>

- Modules
- Components
- Templates
- Metadata
- Data binding
- Directives
- Services
- Dependency injection
- Pipes
- Router
- Testing

Flutter

Desarrollo Nativo



Native

Android Studio

- Java
- Kotlin

XCode

- Objective C
- Swift

Las aplicaciones nativas son de Confianza, Rápidas, pero se creara y mantendra dos Aplicaciones lo cual equivale doble esfuerzo

Desarrollo Hybrid/Bridge



Hybrid

Xamarin

Cordova / Ionic

Reactive Native

NativeScript

Ventajas

- Un código Base
- Menor inversión
 - Costo
 - Tiempo
- Mantenimiento Sencillo

Desventajas

- Rendimiento
- Tamaño



Flutter

Compilación Nativa

- Compilación en Tiempo Real
- Compilación Anticipada


Desarrollo Rápido

Gran Experiencia de Usuario

Unico codigo base para IOS y
Android

No Bridges

Flutter es un Framework muy Joven



A horizontal timeline with two circular markers. The first marker is on the left, and the second is on the right. Above each marker is a light gray rectangular box with a pointer to the marker. The first box contains the text 'May 2017', 'Google I/O', and 'Alpha Kit'. The second box contains the text 'May 2018', 'Google I/O', 'Beta 3 - Listo para', and 'Producción'.

May 2017
Google I/O
Alpha Kit

May 2018
Google I/O
Beta 3 - Listo para
Producción

Flutter Soporte en los Editores

- Visual Studio Code
- Android Studio
- IntelliJ IDEA
- O cualquier otro editor de texto
 - flutter-cli

Nota: Es mejor hacerlo en Visual Studio Code o Android Studio porque te ayuda en la compilación

Recursos

Recursos

- <https://dart.dev/guides/language/specifications/DartLangSpec-v2.2.pdf>
- <https://pub.dev/>
- <https://github.com/gnujavasergio/dart-experiments>
- <https://codelabs.developers.google.com/codelabs/from-java-to-dart/index.html#0>
- <https://medium.com/flutter-community/code-sharing-with-bloc-b867302c18ef>
-

Muchas Gracias

Preguntas?

Sergio Antonio Ochoa Martinez

Developer OpenKM

Developer Freelancer

<https://github.com/gnujavaserio>

gnu.java.sergio@gmail.com

sochoa@openkm.com

