

# Kotlin en Fandango Latam



Bruno Aybar

 @brunoaybarg

 @bruno.aybar

 Bruno125







+ Legible

+ Conciso

+ Idiomático

- Lineas de código

# val / var

```
public class Usuario{  
    private String nombre;  
    private final String apellido;  
}
```

# val / var

```
public class Usuario{  
    private String nombre;  
    private final String apellido;  
}
```

```
class Usuario{  
    var nombre: String  
    val apellido: String  
}
```

# val / var

```
public class Usuario{  
    private String nombre;  
    private final String apellido;  
}
```

```
class Usuario{  
    var nombre: String  
    val apellido: String  
}
```



# Null Safety

```
public class Usuario{  
    private String nombre; ← null  
    private String apellido; ← null  
}
```

```
nombre.equals("hola") ← NullPointerException!!!
```

# Null Safety

```
class Usuario{  
    var nombre: String  
    val apellido: String?  
}
```

# Null Safety

```
class Usuario{  
    var nombre: String = "Bruno"  
    val apellido: String? = null  
}
```

apellido?.equals("Aybar") ✓

# Data classes

```
public class Usuario{
    private String nombre;
    private String apellido;
    private int edad;

    public Usuario(String nombre, String apellido, int edad) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
    }
    public Usuario(int edad) { this.edad = edad; }
    public Usuario() { }

    @Override
    public boolean equals(Object obj) {
        if(obj instanceof Usuario) {
            Usuario otro = (Usuario) obj;
            return nombre.equals(otro.getNombre()) &&
                apellido.equals(otro.getApellido())
                && edad == otro.getEdad();
        }else
            return false;
    }

    @Override
    public boolean hash(Object obj) {
        ...
    }

    @Override
    public boolean toString() {
        ...
    }

    public String getNombre() { return nombre; }

    public void setNombre(String nombre) {this.nombre = nombre; }

    public String getApellido() { return apellido; }

    public void setApellido(String apellido) {this.apellido = apellido; }

    public int getEdad() { return edad; }

    public void setEdad(int edad) { this.edad = edad; }

}
```

# Data classes

```
public class Usuario{
    private String nombre;
    private String apellido;
    private int edad;

    public Usuario(String nombre, String apellido, int edad) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
    }
    public Usuario(int edad) { this.edad = edad; }
    public Usuario() { }

    @Override
    public boolean equals(Object obj) {
        if(obj instanceof Usuario) {
            Usuario otro = (Usuario) obj;
            return nombre.equals(otro.getNombre()) &&
                apellido.equals(otro.getApellido())
                && edad == otro.getEdad();
        }else
            return false;
    }

    @Override
    public boolean hash(Object obj) {
        ...
    }

    @Override
    public boolean toString() {
        ...
    }

    public String getNombre() { return nombre; }

    public void setNombre(String nombre) {this.nombre = nombre; }

    public String getApellido() { return apellido; }

    public void setApellido(String apellido) {this.apellido = apellido; }

    public int getEdad() { return edad; }

    public void setEdad(int edad) { this.edad = edad; }

}
```

```
data class User(nombre: String,
    apellido: String,
    edad: Int)
```

# Funciones de Extensión

```
fun Context.isTablet(): Boolean {  
    val xlarge = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
                == Configuration.SCREENLAYOUT_SIZE_XLARGE  
    val large = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
                == Configuration.SCREENLAYOUT_SIZE_LARGE  
    return xlarge || large  
}
```

# Funciones de Extensión

```
fun Context.isTablet(): Boolean {  
    val xlarge = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
                == Configuration.SCREENLAYOUT_SIZE_XLARGE  
    val large = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
               == Configuration.SCREENLAYOUT_SIZE_LARGE  
    return xlarge || large  
}
```

# Funciones de Extensión

```
fun Context.isTablet(): Boolean {  
    val xlarge = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
                == Configuration.SCREENLAYOUT_SIZE_XLARGE  
    val large = (resources.configuration.screenLayout and Configuration.SCREENLAYOUT_SIZE_MASK)  
                == Configuration.SCREENLAYOUT_SIZE_LARGE  
    return xlarge || large  
}
```

## MainFragment.kt

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    if (getContext.isTablet()) {  
        ...  
    }  
}
```



# Kotlin Android Extensions

activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
    xmlns:android="http://schemas.android.com/apk/res/android">

    <TextView
        android:id="@+id/myTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</android.support.v4.widget.NestedScrollView>
```

# Kotlin Android Extensions

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
    xmlns:android="http://schemas.android.com/apk/res/android">

    <TextView
        android:id="@+id/myTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</android.support.v4.widget.NestedScrollView>
```

## MainActivity.xml

```
import kotlinx.android.synthetic.main.activity_main.*

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    myTextView.text = "Hola mundo!"
}
```





[inserta logo de tu  
empresa aqui]



# Experimenta con el lenguaje

# Kotlin Koans

The screenshot displays the Kotlin Koans IDE interface. The breadcrumb navigation at the top reads "Kotlin Koans > Introduction > Hello, world! > Task.kt". The left sidebar shows a tree view of the koans, with "Introduction" (0/42) expanded to show "Hello, world!" and "Task.kt" (the current task). Below "Introduction" are various other koan categories like "Java to Kotlin conversion", "Named arguments", "Default arguments", "Lambdas", "Strings", "Data classes", "Nullable types", "Smart casts", "Extension functions", "Object expressions", "SAM conversions", and "Extensions on collections".

The main editor area is titled "Simple Functions" and contains the following text:  
Take a look at [function syntax](#) and make the function `start` return the string "OK".  
In the tasks the function `TODO()` is used that throws an exception. Your job during the koans will be to replace this function invocation with a meaningful code according to the problem.

Below the text are three buttons: "Check", "Revert", and "Show answer".

The code editor shows the following Kotlin code:

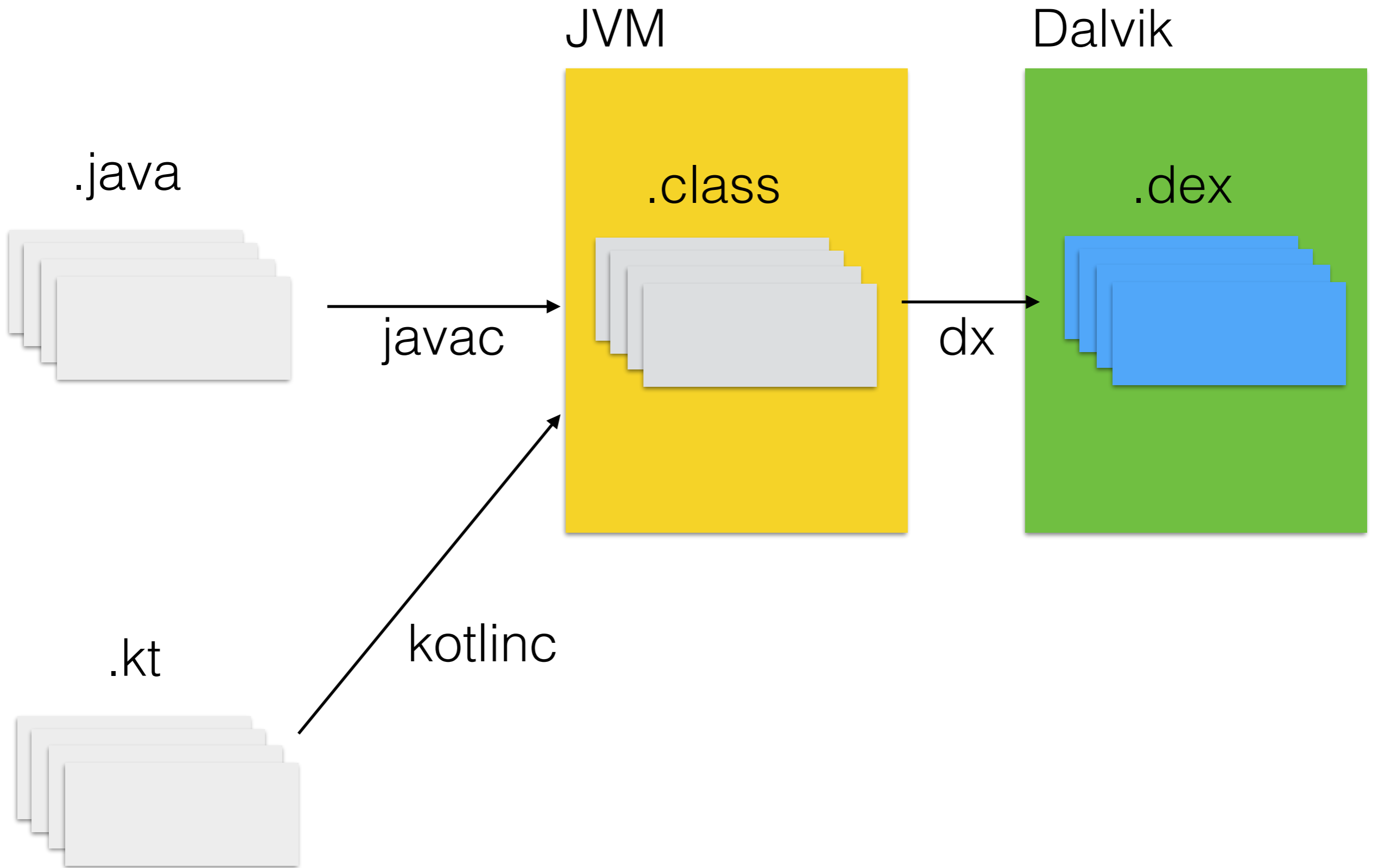
```
1 fun start(): String {  
2  
3     print("abc")  
4     return "OK"  
5 }
```

At the bottom of the IDE, there is a status bar with "Projects headers were loaded" on the left, "On-the-fly type checking" (unchecked) on the right, and "This demo is running on Kotlin v. 1.1.3" at the bottom right. The bottom left corner has tabs for "Problems view", "Console", and "Generated classfiles".

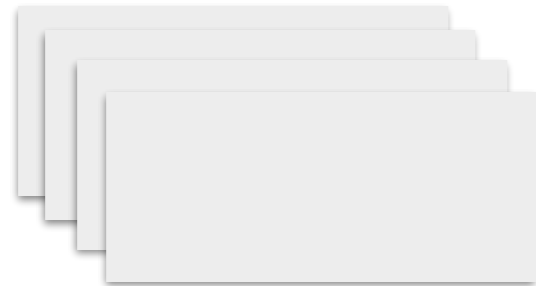
# Proyectos personales



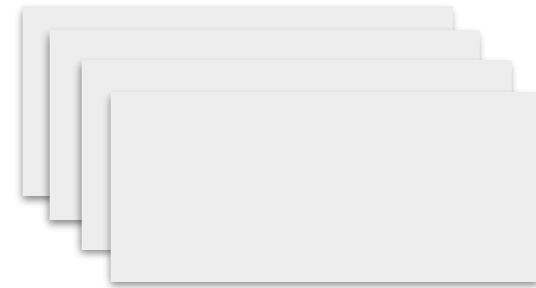




.java



.kt





Usuario.java

```
public class Usuario{  
    //...  
}
```



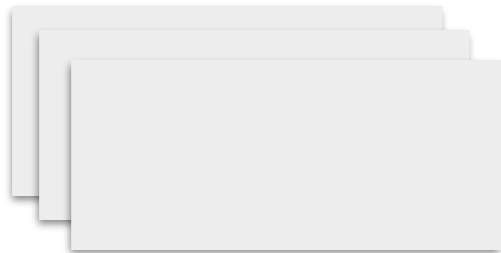
Usuario.java

```
public class Usuario{  
    //...  
}
```

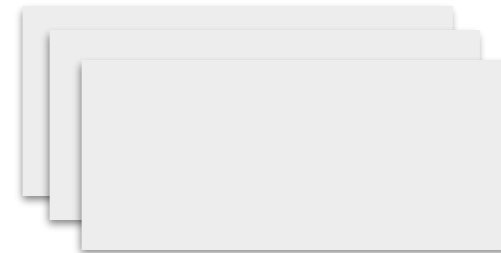
Android.kt

```
class Android{  
    fun getUser() = Usuario()  
}
```

.java



.kt

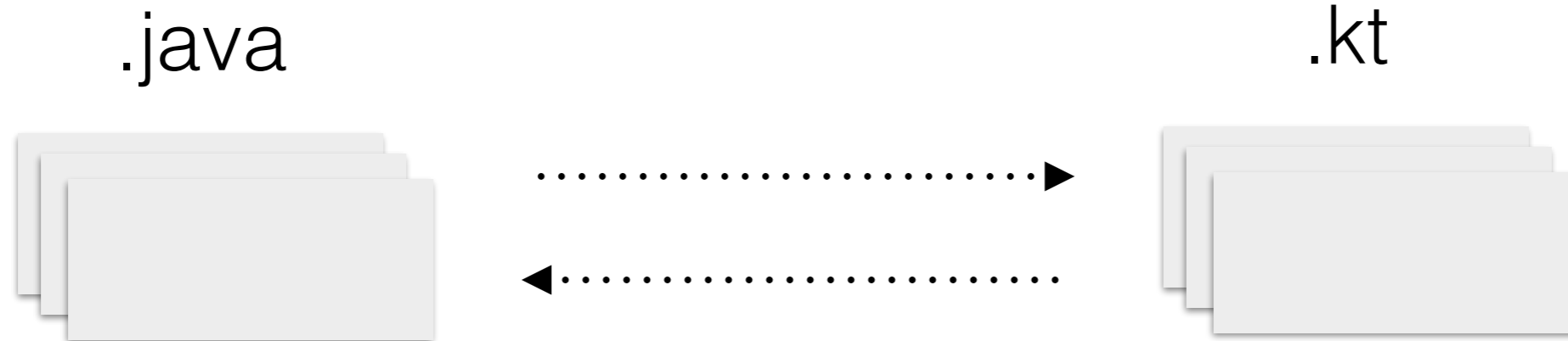


Usuario.java

```
public class Usuario{  
    //...  
}
```

Android.kt

```
class Android{  
    fun getUser() = Usuario()  
}
```



Usuario.java

```
public class Usuario {  
    void foo() {  
        new Android();  
    }  
}
```

Android.kt

```
class Android{  
    fun getUser() = Usuario()  
}
```

Java & Kotlin

100%

interoperables

# No incrementa considerablemente el tamaño de tu APK

File	Raw File Size	Download Size	% of Total Download size
classes.dex	8,6 MB	3,3 MB	26,8%
res	6,3 MB	5,6 MB	45,1%
classes2.dex	4,2 MB	1,6 MB	12,6%
lib	3,8 MB	1,3 MB	10,4%
resources.arsc	1,1 MB	204,7 KB	1,6%
assets	570,6 KB	305,9 KB	2,4%
META-INF	382,1 KB	124 KB	1%
AndroidManifest.xml	29,8 KB	5,5 KB	0%
<b>kotlin</b>	<b>26,1 KB</b>	<b>9,3 KB</b>	<b>0,1%</b>
fabric	1,1 KB	760 B	0%
build-data.properties	1 KB	583 B	0%
jsr305_annotations	133 B	104 B	0%



# 2011

Jetbrains anuncia el proyecto Kotlin



# 2012

Kotlin se vuelve Open Source

# 2016

Kotlin v1.0

# 2017

Google le da soporte a Kotlin



WHAT DO 17 GOOGLE  
DEVELOPERS EXPERTS  
FOR ANDROID THINK  
ABOUT KOTLIN?

ANTONIO LEIVA

---

antonioleiva.com



Google Developers  
**Experts**



**Kotlin**

<https://antonioleiva.com/google-kotlin/>

WHY?

+ Rápido  
+ Mantenable  
+ Estable

# NullPointerException

The Billion Dollar Mistake



Reducimos la brecha  
entre Android y iOS



<http://nilhcem.com/swift-is-like-kotlin/>

Swift is like Kotlin

## BASICS

Hello World

Swift

```
print("Hello, world!")
```

Kotlin

```
println("Hello, world!")
```

Variables And Constants

Swift

```
var myVariable = 42  
myVariable = 50  
let myConstant = 42
```

Kotlin

```
var myVariable = 42  
myVariable = 50  
val myConstant = 42
```

Explicit Types

Swift

```
let explicitDouble: Double = 70
```

Kotlin

```
val explicitDouble: Double = 70.0
```





“It's a breath of fresh air in the AndroidDev world”  
- Donn Felker

# Recomendaciones

Empezar haciendo pruebas

# Recomendaciones

Empezar haciendo pruebas

¡No es necesario que todo esté en Kotlin!

# Recomendaciones

Empezar haciendo pruebas

¡No es necesario que todo esté en Kotlin!

¡No es necesario **transformar** todo Kotlin!

# Recomendaciones

Empezar haciendo pruebas

¡No es necesario que todo esté en Kotlin!

¡No es necesario **transformar** todo Kotlin!

**Enter action or option name:**

🔍 convert



**Convert** Java File to Kotlin File ↵⌘K

Code



# Recomendaciones

Empezar haciendo pruebas

¡No es necesario que todo esté en Kotlin!

¡No es necesario transformar todo Kotlin!

Feedback en el equip

# Recomendaciones

Empezar haciendo pruebas

¡No es necesario que todo esté en Kotlin!

¡No es necesario transformar todo Kotlin!

Feedback en el equipo

Establecer guías



# Gracias!

 @brunoaybarg

 @bruno.aybar

 Bruno125