



EBook Gratis

APRENDIZAJE android-ndk

Free unaffiliated eBook created from
Stack Overflow contributors.

#android-
ndk

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con android-ndk.....	2
Observaciones.....	2
¿Qué es el NDK de Android?.....	2
Versiones.....	2
Examples.....	2
Comenzando con Android NDK con un simple ejemplo.....	2
Creditos.....	13

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [android-ndk](#)

It is an unofficial and free android-ndk ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official android-ndk.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con android-ndk

Observaciones

¿Qué es el NDK de Android?

El Kit de desarrollo nativo de Android (NDK) es una herramienta complementaria del SDK de Android que permite que partes de aplicaciones se incorporen en C / C ++. Esto es útil para:

- Compartir componentes de aplicaciones a través de plataformas (Android, iOS, Linux, etc.)
- Mejora del rendimiento para porciones críticas.
- Reutilizando bibliotecas C / C ++ existentes

El NDK proporciona encabezados y bibliotecas que permiten al desarrollador crear actividades, manejar las opiniones de los usuarios, usar sensores de hardware, acceder a los recursos de la aplicación y más, todo al mismo tiempo programar en C / C ++.

Versiones

Versión	Fecha de lanzamiento
r12	2016-06-09
r11	2016-03-09
r10	2014-07-01

Examples

Comenzando con Android NDK con un simple ejemplo

Usando Android Studio 2.2 y superior Native Development Kit (NDK), puede usarlo para compilar códigos C y C ++.

Puedes usar NDK descargando NDK manualmente y compilándolo o a través de CMake.

Aquí daré un flujo de proceso para instalar NDK manualmente y un código de ejemplo,

Según el sistema operativo de su sistema, puede descargar NDK desde esta ubicación <https://developer.android.com/ndk/downloads/index.html> .

Después de la descarga, **indique** la ruta en la Variable del entorno del sistema como nombre de variable " **NDK_PROJECT_PATH** " y el valor de la variable "ubicación de la ruta almacenada NDK".

- A continuación, para integrar NDK con Android Studio, después de crear un nuevo proyecto de Android,

En gradle - propiedades locales agregar ubicación de ruta sdk como

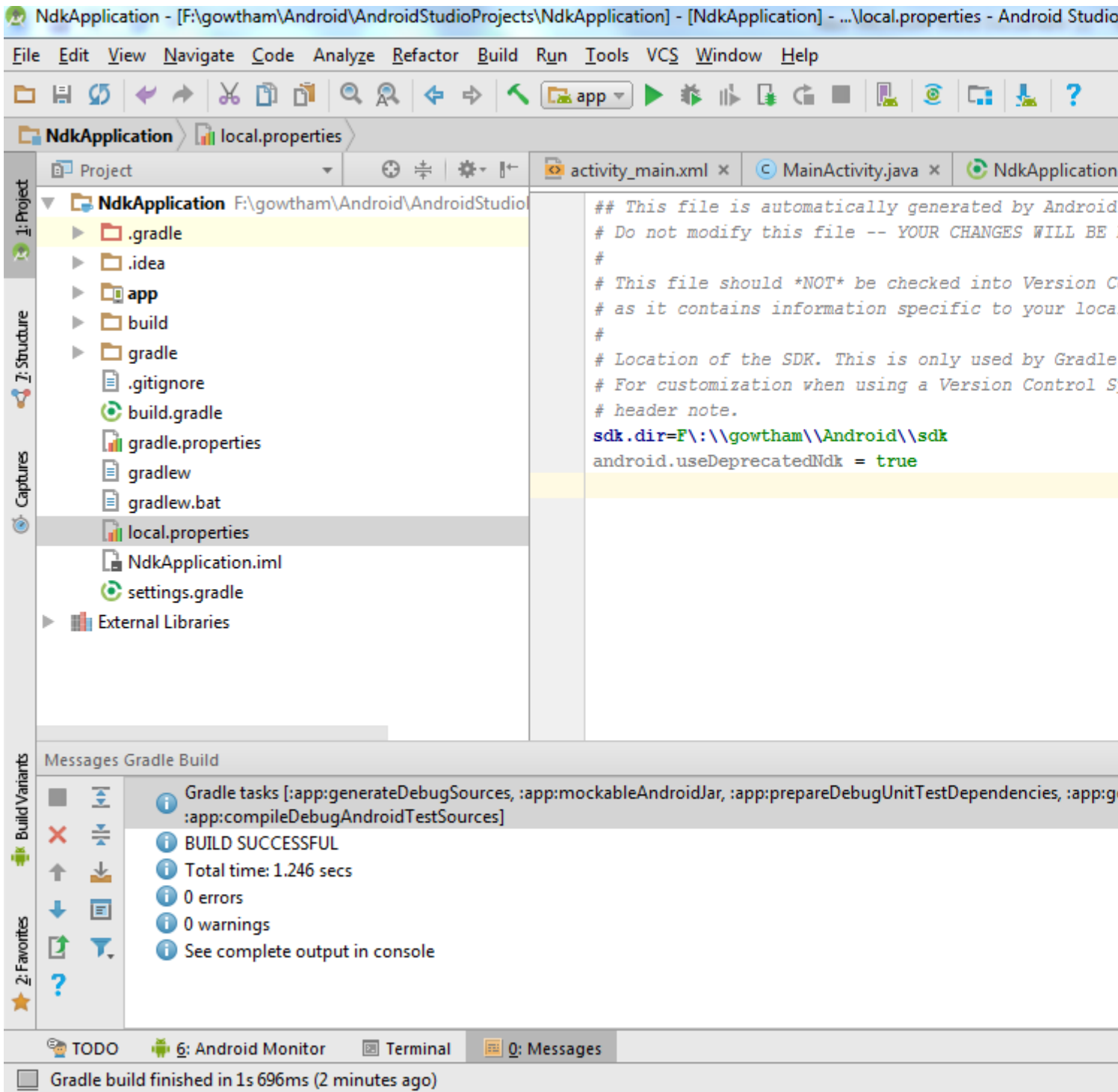
```
sdk.dir=F:\:\gowtham\Android\sdk
```

y

```
android.useDeprecatedNdk = true
```

- Luego, presione Construir - Crear proyecto (Ctrl + f9).

Su proyecto se construirá con éxito y obtendrá BUILD SUCCESSFUL en el mensaje gradle build, como se muestra a continuación



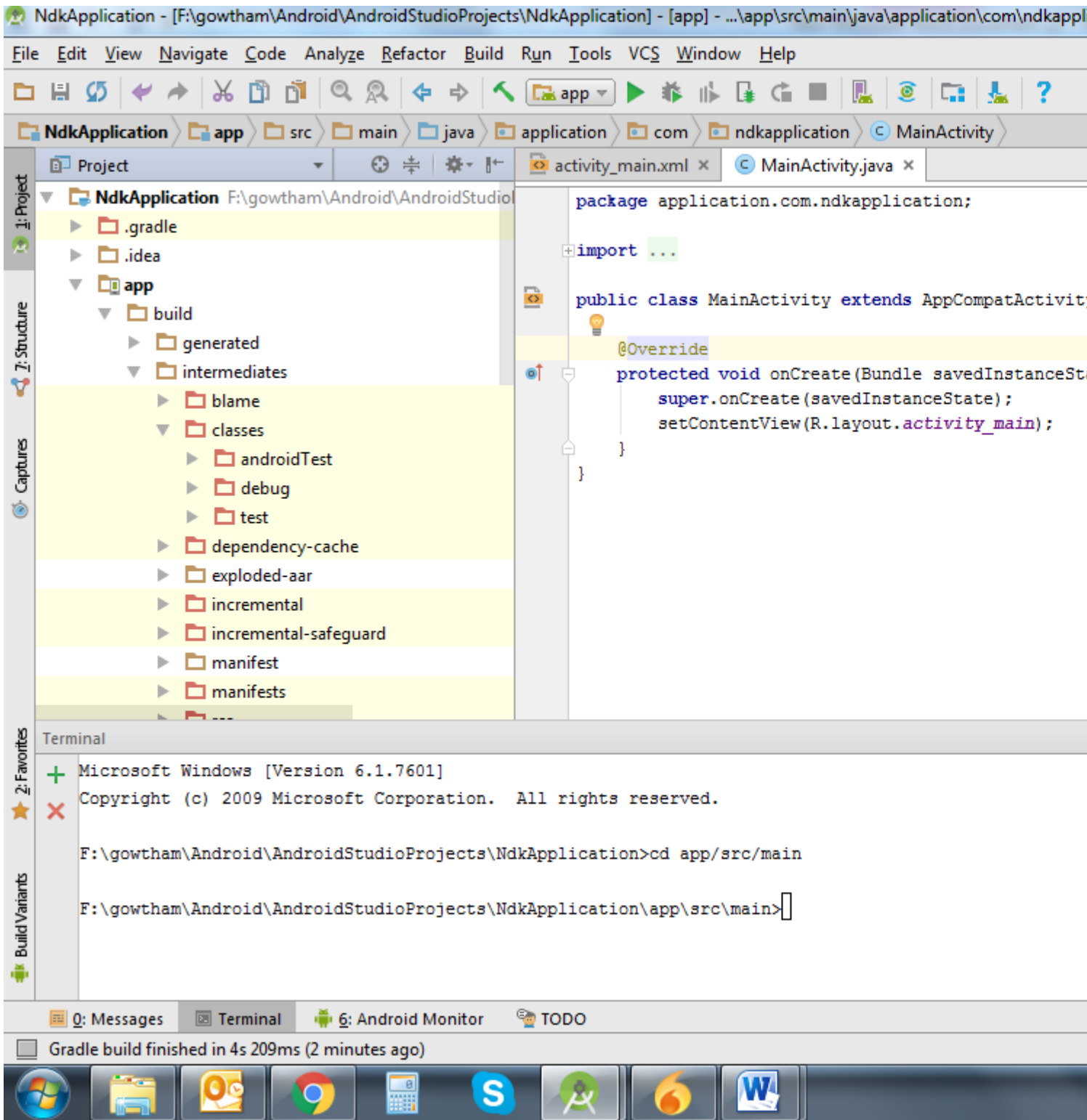
Luego en la Terminal, inicialmente contendrá la ruta del proyecto.

Añada la `cd app/src/main`

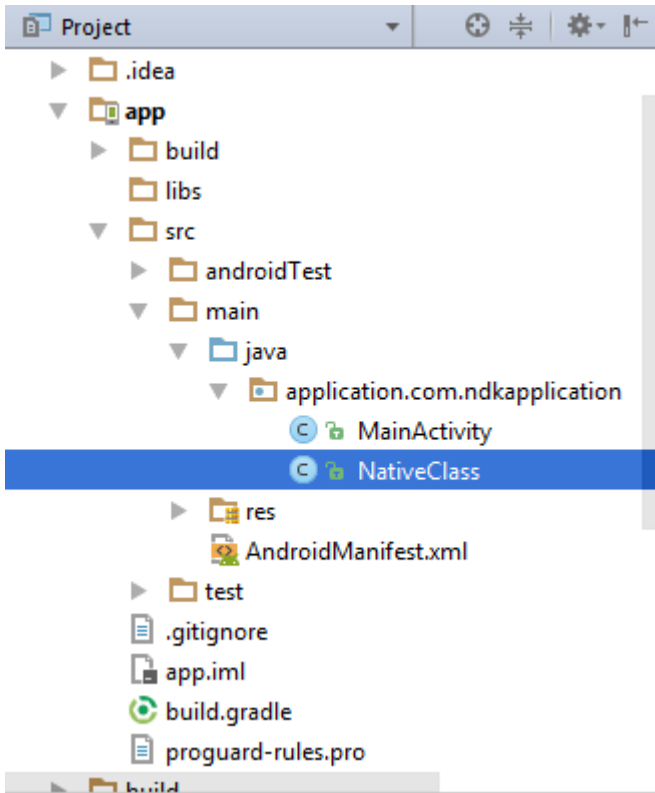
La ruta se extenderá desde la ruta del proyecto hasta la principal.

A continuación, nuevamente **Build** - Make Project (Ctrl + f9).

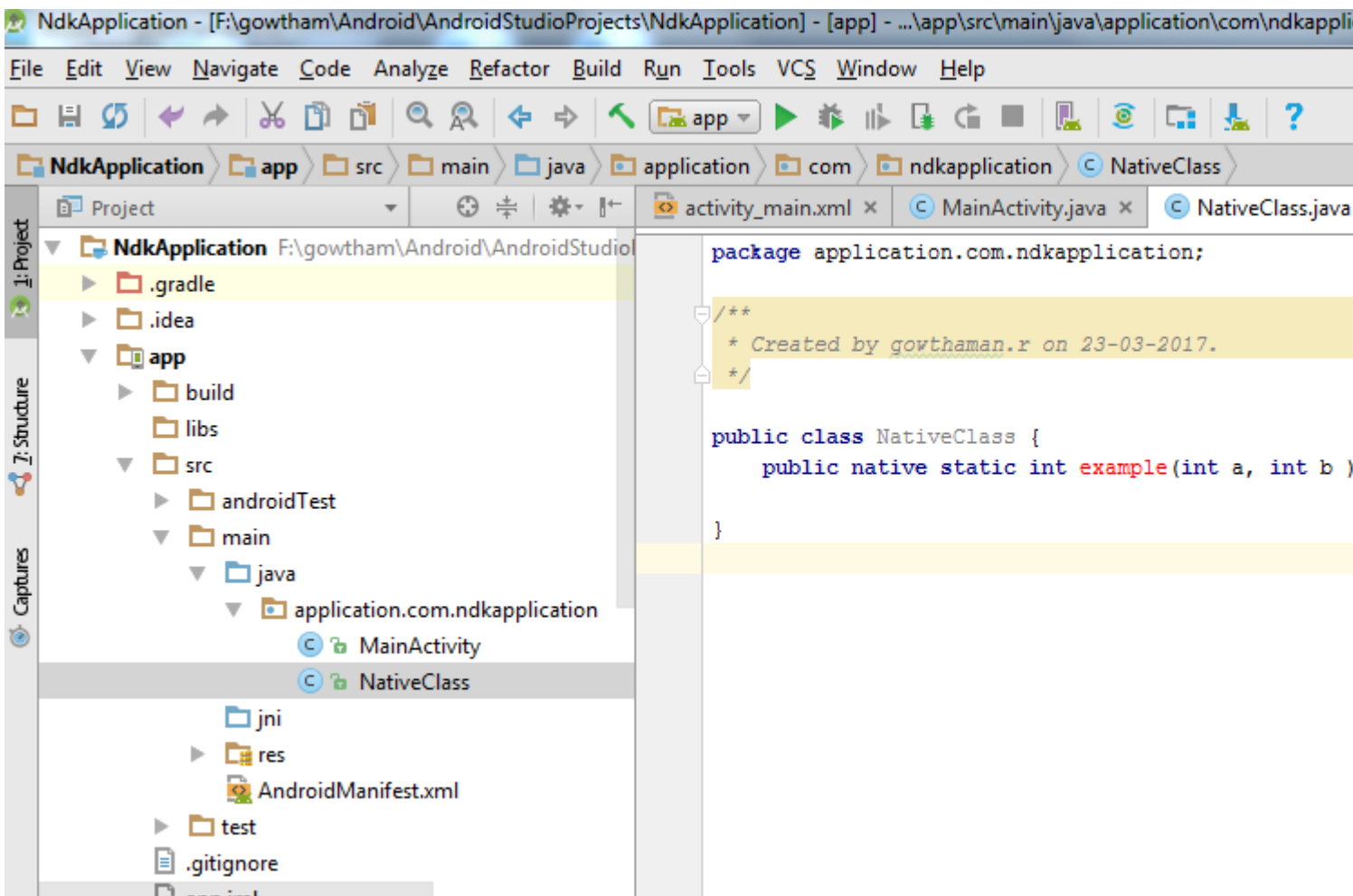
Ahora lo encontrarás en **app-build – intermediates – classes – debug** , como se muestra a continuación.



Ahora, cree un nuevo archivo de clase Java en una aplicación / src / main / java, aquí creé un archivo java llamado NativeClass

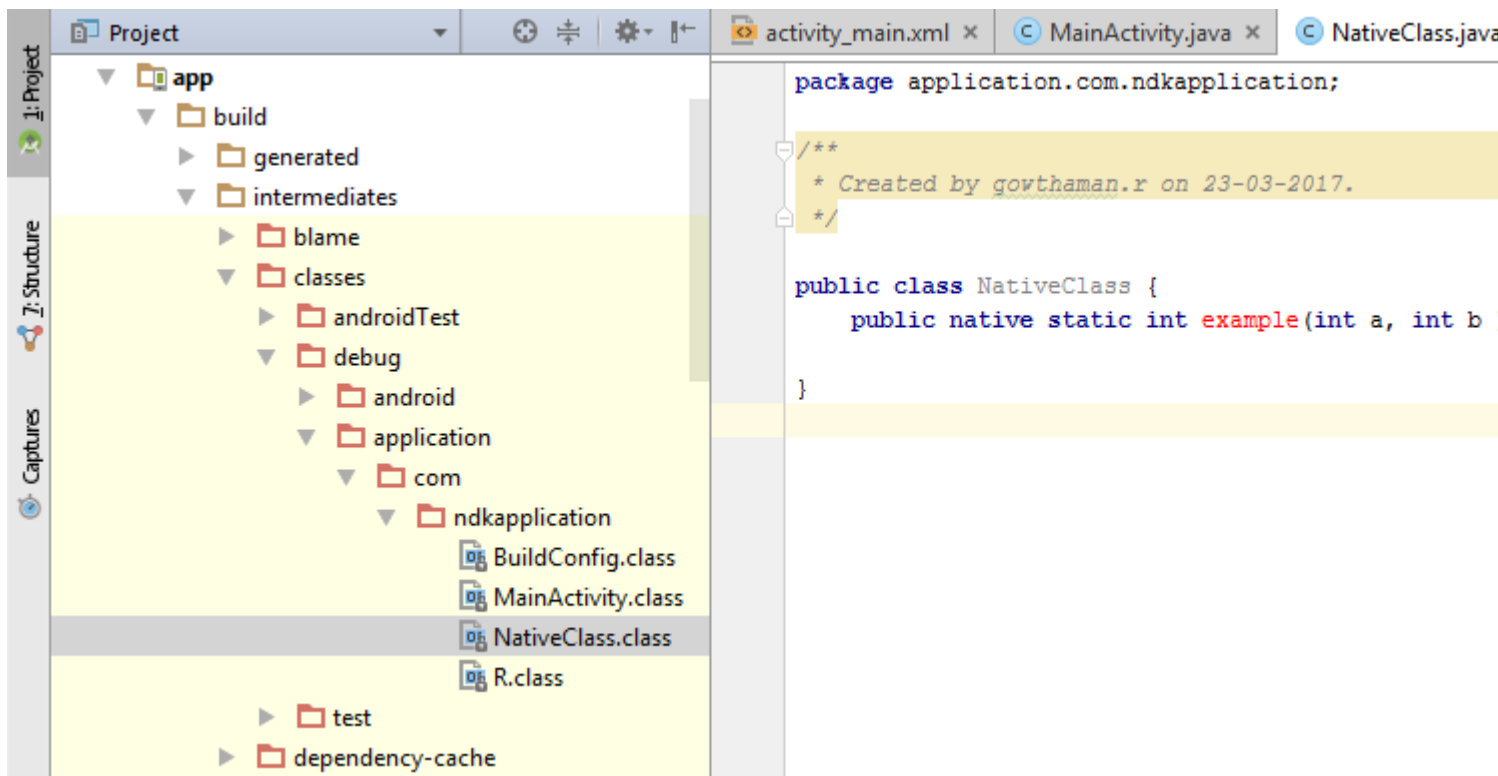


Escriba una función de llamada simple con el nombre de la función y la entrada para la función. Aquí escribí la función como ejemplo y le di dos entradas enteras,



De nuevo construye el proyecto (**Ctrl + f9**) ,

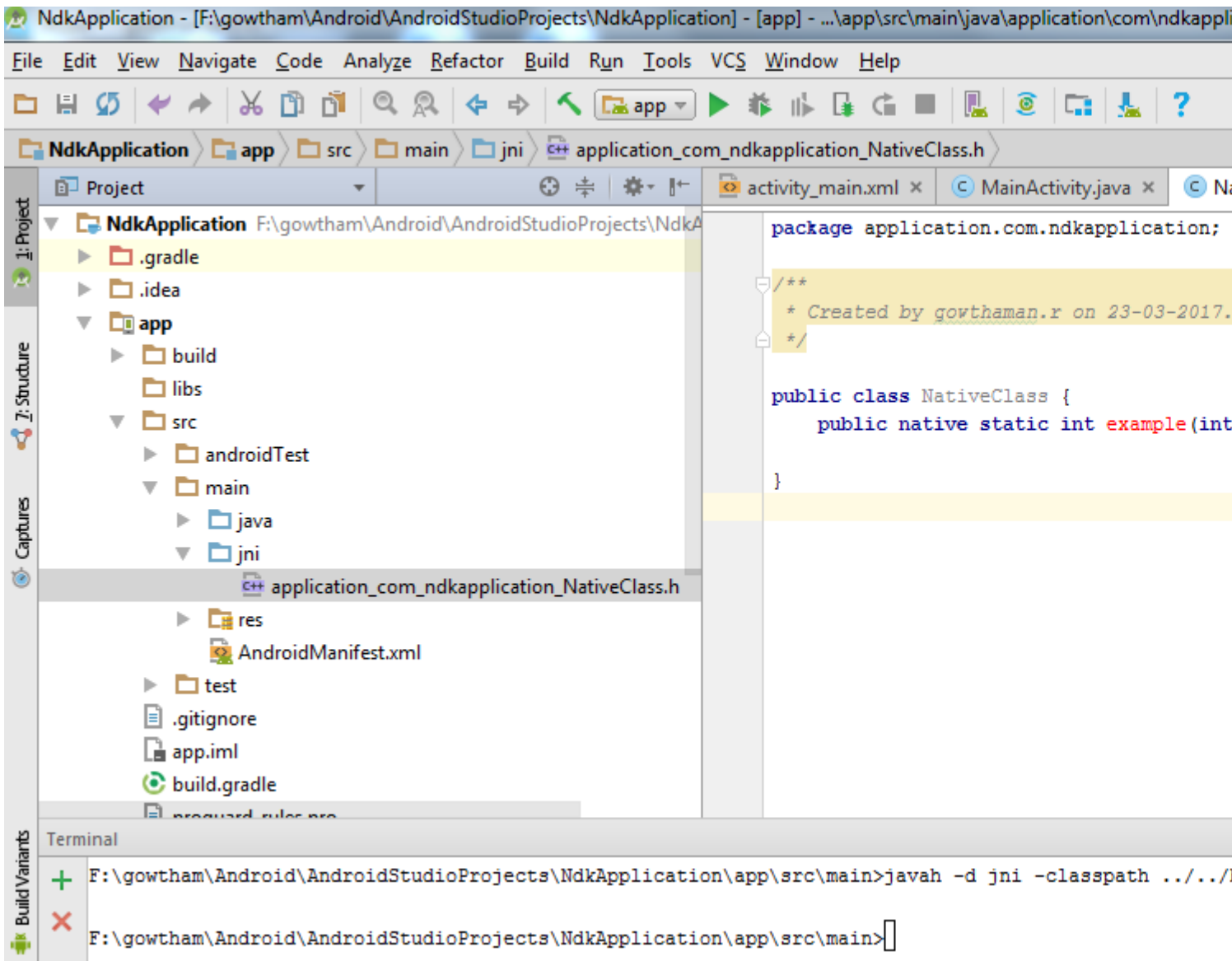
Cuando construyas, encontrarás el archivo de clase creado en esta compilación,



```
Then, in terminal
[] -Javah -d jni -classpath
.../build/intermediates/classes/debug`application.com.ndkapplication.NativeClass
```

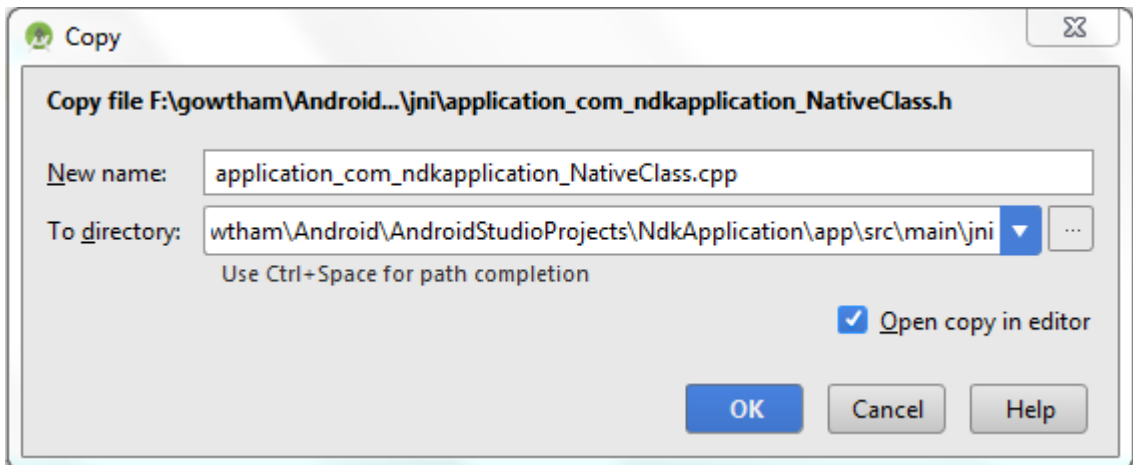
donde, - **d** - para el directorio de salida - **jni -Generar** archivo de encabezado de estilo JNI (predeterminado) - **classpath** - para el cual cargar clases

Ahora cree el proyecto (Ctrl + f9), encontrará la carpeta **jni** creada y se creará un **archivo de encabezado** con el nombre que especificó en el terminal antes, como se muestra a continuación.



Ahora copie el archivo de encabezado y guárdelo como archivo **.cpp o .c** con el mismo nombre del archivo de encabezado en la carpeta jni.

He creado un archivo cpp como se muestra a continuación



Delete everything before ****JNIEXPORT** line** in this file and ****add header file name**** alone. Here I am for simple example just adding two numbers and returning value to android java.

Application_com_ndkapplication_NativeClass.cpp

```
#include <application_com_ndkapplication_NativeClass.h>

JNIEXPORT jint JNICALL Java_application_com_ndkapplication_NativeClass_example
    (JNIEnv *, jclass , jint as , jint bs){

    return (as + bs);

}
```

Y para este ejemplo no es necesario agregar ninguna función en su archivo de encabezado.

Cree un nuevo archivo llamado **Android.mk** y **Application.mk** en la carpeta **Jni**

El archivo **Android.mk** es para describir sus fuentes al sistema de compilación.

Android.mk

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_SRC_FILES := application_com_ndkapplication_NativeClass.cpp

LOCAL_LDLIBS += -llog

LOCAL_MODULE := exp-jni

include $(BUILD_SHARED_LIBRARY)
```

para conocer detalles sobre este archivo, lea este enlace

https://developer.android.com/ndk/guides/android_mk.html

```
Android.mk x
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_SRC_FILES := application_com_ndkapplication_NativeClass.cpp

LOCAL_LDLIBS += -llog

LOCAL_MODULE := exp-jni

include $(BUILD_SHARED_LIBRARY)
```

Application.mk que describe los módulos nativos que requiere su aplicación.

Aplicación.mk

```
APP_STL := gnu STL_static

APP_CPPFLAGS := -frtti -fexceptions

APP_ABI := armeabi-v7a armeabi arm64-v8a mips mips64 x86 x86_64

APP_PLATFORM := android-16
```

para conocer detalles sobre este archivo, lea este enlace

https://developer.android.com/ndk/guides/application_mk.html

```
Application.mk x
APP_STL := gnuSTL_static

APP_CPPFLAGS := -frtti -fexceptions

APP_ABI := armeabi-v7a armeabi arm64-v8a mips mips64 x86 x86_64

APP_PLATFORM := android-16
```

Now build the project again **(Ctrl + f9)**, you will find the **armeabi-v7a, armeabi, arm64-v8a, mips, mips64, x86 and x86_64** folder created inside **jniLibs**.

Luego, en la actividad principal, pase la entrada y obtenga la salida para el archivo de clase nativa.

```
int a = 5, b = 5, res ;

res = NativeClass.example(((int) a), ((int) b));

TextView textView = (TextView) this.findViewById(R.id.tv);

textView.setText(new Integer(res).toString());
```

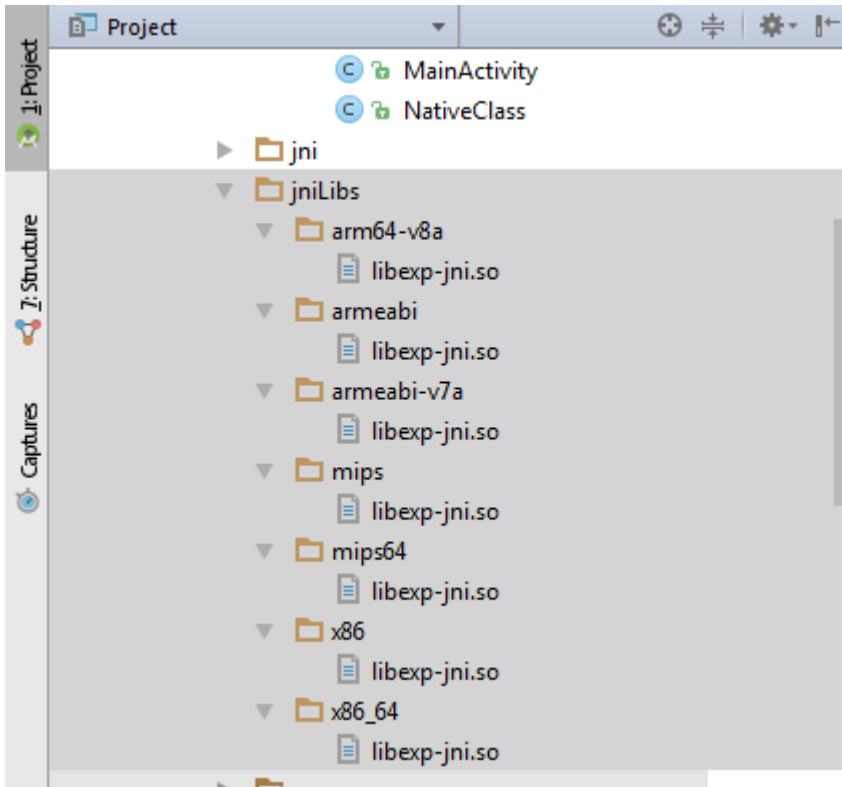
Aquí di dos entradas de enteros a través de a, b y obtengo la salida de la variable res.

Y la salida obtenida se muestra en la pantalla al pasar a TextView.

Y no olvide agregar la **biblioteca** que especificó en el archivo **Android.mk** como **LOCAL_MODULE** como este,

```
static {
    System.loadLibrary("exp-jni");
}
```

Finalmente, vuelva a compilar el proyecto (Ctrl + f9), encontrará los archivos .so creados en cada carpeta armeabi-v7a, armeabi, arm64-v8a, mips, mips64, x86 y x86_64.



Entonces ahora ejecute la aplicación, obtendrá un resultado para este ejemplo como 10.



Este es un programa básico para principiantes del NDK, la biblioteca OpenCV se puede importar aquí y también puede hacer aplicaciones de procesamiento de imágenes.

Lea [Empezando con android-ndk en línea](https://riptutorial.com/es/android-ndk/topic/6314/empezando-con-android-ndk): <https://riptutorial.com/es/android-ndk/topic/6314/empezando-con-android-ndk>

Creditos

S. No	Capítulos	Contributors
1	Empezando con android-ndk	4444 , Alex , Community , Dan Albert , Gowthaman , Jomo Fisher