

# 1 LIBGDX Creacion dun proxecto base

## UNIDADE 1: Creación dun proxecto base

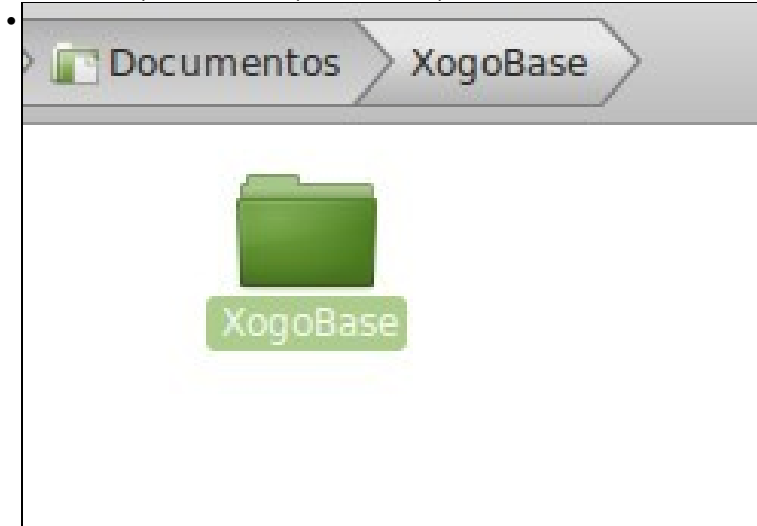
Supoñamos que facemos un proxecto base para os nosos xogos (como imos facer neste curso) no que creamos a estrutura de clases e implementamos o código que imos necesitar en todos os xogos, de tal forma que despois queremos reutilizar todo o feito para facer outro xogo diferente e así non ter que empezar dende 0.

Para facelo:

- Facendo un proxecto BASE



Debemos copiar todos os arquivos xerados pola ferramenta.



Os copiamos a un cartafol de nome XogoBase. Este cartafol non se debe tocar xa que o seu contido será a base de novos xogos.



Agora se queremos crear un novo xogo só temos que copiar o contido do cartafol XogoBase a outro novo. Por exemplo ó cartafol NovoXogo. Unha vez copiado debemos editar os arquivos build.gradle e .project.

```

buildscript {
    repositories {
        mavenCentral()
        mavenLocal()
        maven { url 'https://github.com/steffenschaefer/gwt-gradle-plugin/raw/maven-repo/' }
    }
    dependencies {
        classpath 'de.richsource.gradle.plugins:gwt-gradle-plugin:0.3'
        classpath 'com.android.tools.build:gradle:0.9+'
    }
}

allprojects {
    apply plugin: "eclipse"
    apply plugin: "idea"

    version = '1.0'
    ext {
        appName = 'Xogo2D NOVO'
        gwtVersion = '1.11.1-SNAPSHOT'
        roboVMVersion = '0.0.11'
    }
}

```

No build.gradle dámoslle un novo nome ó xogo. Este será o nome do proxecto onde vai a información que utiliza o gradle.

```

<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
    <name>xogosxerados nuevo</name>
    <comment></comment>
    <projects>
    </projects>
    <buildSpec>
        <buildCommand>
            <name>org.eclipse.jdt.core.javabuilder</name>
            <arguments>
            </arguments>
        </buildCommand>
    </buildSpec>
    <natures>

```

No .project dámoslle un novo nome ó proxecto.



Despois temos que ir a cada unha das plataformas que xera a ferramenta.



Empezamos pola **core**. Neste só temos que editar o arquivo .project.

```

<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
    <name>Xogo2D NUEVO-core</name>
    <comment></comment>
    <projects>
    </projects>
    <buildSpec>
        <buildCommand>
            <name>org.eclipse.jdt.core.javabuilder</name>
            <arguments>
            </arguments>
        </buildCommand>
    </buildSpec>
    <natures>
        <nature>org.springframework.ide.eclipse.gradle.core.nature</nature>
        <nature>org.eclipse.jdt.core.javanature</nature>
    </natures>
</projectDescription>

```

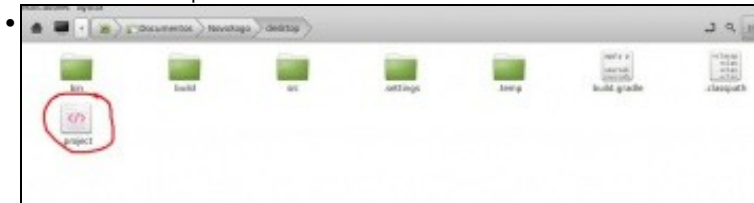
Dámoslle un novo nome ó proxecto core.

```

<?xml version="1.0" encoding="UTF-8"?>
<classpath>
    <classpathentry kind="src" path="src"/>
    <classpathentry kind="src" path="src"/>
    <classpathentry exported="true" kind="src" path="src"/>
    <classpathentry exported="true" kind="com" path="com.android.ide.eclipse.adt.ANDROID_FRAMEWORK"/>
    <classpathentry exported="true" kind="com" path="com.android.ide.eclipse.adt.LIBRARIES"/>
    <classpathentry exported="true" kind="com" path="org.springframework.ide.eclipse.gradle.classpathcontainer"/>
    <classpathentry exported="true" kind="com" path="com.android.ide.eclipse.adt.DEPENDENCIES"/>
    <classpathentry kind="output" path="bin/classes"/>
</classpath>

```

Despois temos cambiar o .classpath da versión Android e html, indicando o novo nome do proxecto core (feito antes). Non se fai no cartafol da versión Desktop.



Debemos editar o arquivo .project das tres plataformas: Android, Html e Desktop.

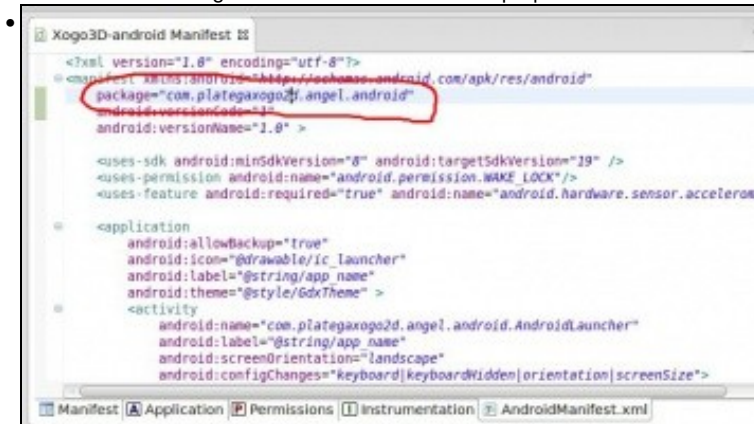


Cambiamos o nome dos proxectos nas tres plataformas: Android, Html e Desktop.

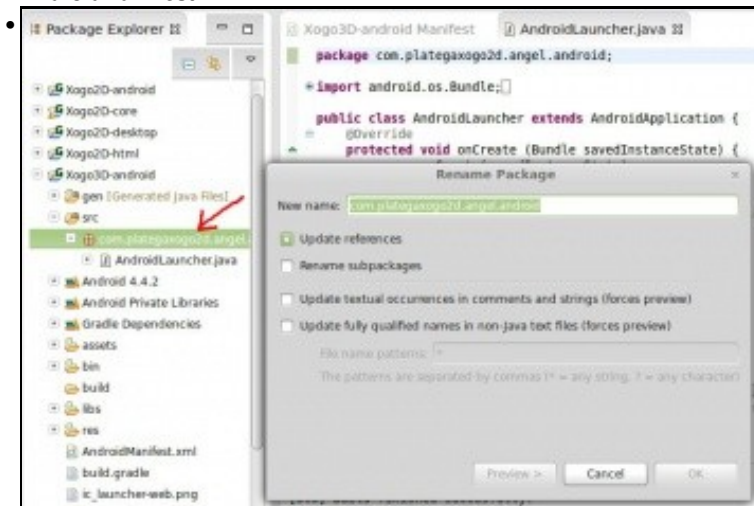
Agora só temos que facer a importación dos proxectos, como [indicamos anteriormente](#).

A maiores, é aconsellable cambiar o nome do paquete do xogo (polo menos na versión Android).

- Creando un novo xogo. Cambiando os nomes dos paquetes



Debemos cambiar o paquete na versión Android xa que cada xogo ten que desenvolverse nun único paquete. Isto se fai no arquivo **AndroidManifest.xml**.



Despois poderemos cambiar todos os paquetes dos demais proxectos (html, desktop, core,...) e darlle o nome que queiramos. Para facelo debemos de seleccionar o paquete, premer o botón dereito sobre o mesmo paquete e escoller a opción **Refactor -> Rename**

- Se cambiades de computador e ides dun Windows a un Linux, por exemplo, pode ser necesario descargar outra vez as librerías de gradle. Para facelo tedes que marcar todos os proxectos, premer o botón dereito sobre un deles, escoller a opción **Gradle** e despois **Refresh all**.
- Se na versión HTML vos aparece unha admiración vermella probade a limpar o proxecto. Tedes que ir ó menú de eclipse **Project=>Clean** e marcar o proxecto HTML.