

Gradle Build Tool Fundamentals:

1. Definición: Gradle es una herramienta flexible que permite definir tareas personalizadas. Por ejemplo:

```
task hello {  
    doLast {  
        println 'Hello, Gradle!'  
    }  
}
```

Este script define una tarea simple que se puede ejecutar con `gradle hello`.

2. Lenguaje: Gradle utiliza Groovy o Kotlin DSL. Aquí hay un ejemplo en Kotlin DSL:

```
plugins {  
    java  
}
```

```
repositories {  
    mavenCentral()  
}
```

```
dependencies {  
    implementation("org.springframework.boot:spring-boot-  
starter-web:2.5.0")  
}
```

3. Modelo de ejecución: Gradle construye un grafo de tareas. Por ejemplo:

```
task taskA {  
    doLast {  
        println 'Task A'  
    }  
}
```

```
task taskB(dependsOn: taskA) {
```

```
doLast {
    println 'Task B'
}
```

Aquí, `taskB` depende de `taskA`, por lo que Gradle ejecutará `taskA` antes de `taskB`.

4. Gestión de dependencias: Gradle permite definir configuraciones personalizadas:

```
configurations {
    integrationTestCompile.extendsFrom testCompile
}
```

```
dependencies {
    integrationTestCompile 'org.mockito:mockito-core:3.+'
}
```

5. Plugins: Los plugins extienden la funcionalidad de Gradle. Por ejemplo, el plugin Java:

```
apply plugin: 'java'
```

Comparación con Maven:

1. Flexibilidad:

Gradle: Permite tareas personalizadas fácilmente:

```
task deployToProduction(type: Ssh) {
    // configuración de despliegue
}
```

Maven: Requiere plugins o scripts externos para tareas personalizadas complejas.

2. Lenguaje de configuración:

Gradle (Groovy):

```
dependencies {
```

```
        implementation 'org.springframework.boot:spring-  
boot-starter-web:2.5.0'  
    }
```

Maven (XML):

```
<dependencies>  
    <dependency>  
        <groupId>org.springframework.boot</groupId>  
        <artifactId>spring-boot-starter-web</artifactId>  
        <version>2.5.0</version>  
    </dependency>  
</dependencies>
```

3. Rendimiento: Gradle utiliza construcción incremental y paralelización:

```
org.gradle.parallel=true  
org.gradle.caching=true
```

4. Curva de aprendizaje:

Gradle: Requiere entender conceptos como closures en Groovy:

```
task printVersion {  
    doLast {  
        println "Version: $version"  
    }  
}
```

Maven: Estructura más predecible y basada en convenciones.

5. Ecosistema:

Gradle: Popular en Android:

```
android {  
    compileSdkVersion 30  
    buildToolsVersion "30.0.3"  
}
```

Maven: Ampliamente usado en proyectos Java empresariales.

Casos de uso de Gradle:

Desarrollo de Android:

```
android {  
    defaultConfig {  
        applicationId "com.example.myapp"  
        minSdkVersion 21  
        targetSdkVersion 30  
    }  
}
```

Proyectos multi-módulo:

```
include ':app', ':library'
```

Personalización de procesos de construcción:

```
task createDocs {  
    doLast {  
        // Generar documentación personalizada  
    }  
}
```

Proyectos poliglota:

```
plugins {  
    id 'java'  
    id 'scala'  
}
```

Integración continua y entrega continua (CI/CD):

```
task deployToStaging(type: SshTask) {  
    // Configuración para desplegar en staging  
}
```

Migraciones graduales:

```
apply plugin: 'java'  
apply plugin: 'maven'
```

```
// Configuración de Maven y Gradle juntas
```