

**PÓS-GRADUAÇÃO EM
DESENVOLVIMENTO DE
APLICAÇÕES PARA
DISPOSITIVOS MÓVEIS E
CLOUD COMPUTING**

DM107

Desenvolvimento de Web Services com segurança sob plataforma Java e PHP

Profa. Daniela E. C. de Almeida

E-mail: daniela.edvana@inatel.br

Prof. Elton Barbosa

E-mail: elton.barbosa@inatel.br

Agenda

- Introdução ao Gradle;
- Evolução das ferramentas de *build*;
- Características;
- *Plugins* Gradle;
- Instalação do Gradle;
- Comandos Gradle;
- Usando o Gradle no Eclipse IDE;

Introdução ao Gradle

- Baseia-se em lições aprendidas com ferramentas já estabelecidas como Ant e Maven e leva suas melhores práticas.
- Segue uma abordagem de compilação por convenção.
- Permite modelar declarativamente seu domínio de problemas usando uma linguagem específica de domínio (DSL) implementada em Groovy em vez de XML.
- Permite a escrita da lógica personalizada no idioma com o qual você se sente mais confortável, seja Java ou Groovy.

Evolução das ferramentas de *builds*

- Ant e Maven dominaram a construção de projetos Java.
- São altamente populares e padrões da indústria.
- Possuem como ponto fraco: a lógica de compilação precisa ser descrita em XML.

Características

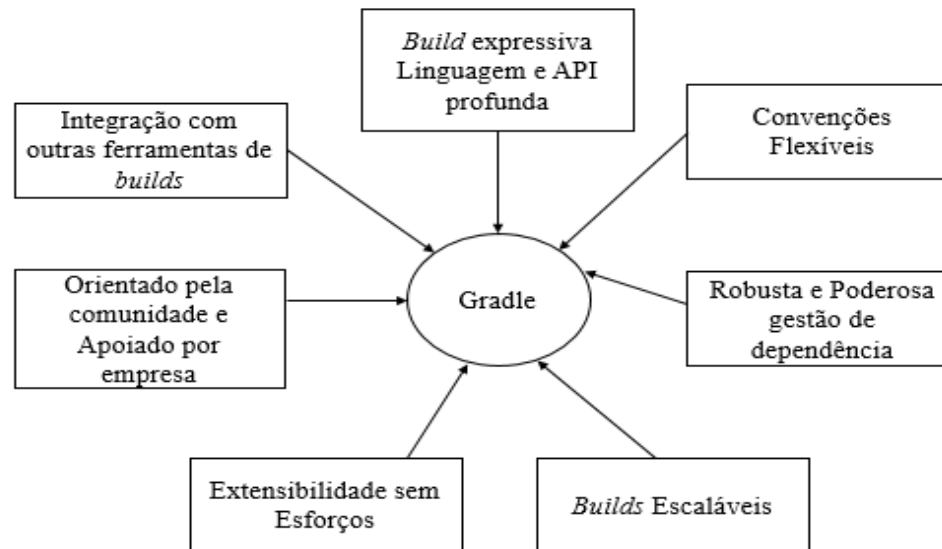


Figura 1- Conjunto de características

Características

- **Convenções Flexíveis**

Todo o projeto Java em Gradle sabe:

- Onde o arquivo de origem e a classe de teste devem estar;
- Como compilar seu código;
- Executar testes de unidade;
- Gerar relatórios Javadoc;
- Criar uma distribuição de seu código.

Se a convenção for cumprida, há apenas um esforço de configuração mínimo.

Características

- **Robusta e Poderosa gestão de dependências**

Fornece uma infra-estrutura para gerenciar a complexidade da resolução, recuperação e armazenamento de dependências.

Uma vez baixadas as bibliotecas e colocados no cache local, elas estão disponíveis para o projeto.

Cada um dos submódulos é considerado um projeto que pode definir dependências para bibliotecas externas ou outros módulos.

Características

- **Robusta e Poderosa gestão de dependências**

Cada subprojeto pode ser executado individualmente.

Características

- ***Builds* Escaláveis**

Suporta compilações incrementais especificando entradas e saídas de tarefas.

Calcula quais tarefas precisam ser ignoradas, construídas ou parcialmente reconstruídas.

Suporta a execução de testes paralelos.

Pode ser executado no modo *daemon*.

Características

- **Extensibilidade sem esforço**

A maneira mais fácil de implementar a lógica personalizada é escrever uma *task*.

Os *plugins* são o mecanismo de extensão mais poderoso do Gradle, eles oferecem acesso completo à API e podem ser escritos, testados e distribuídos como qualquer outro *software*.

Características

- **Integração com outras ferramentas de *build***

Permite que importe-se a lógica de compilação existente e reutilize tarefas Ant padrão.

Compilações do Gradle são 100% compatíveis com os repositórios Maven e Ivy.

Fornece um conversor para construções Maven existentes que podem traduzir a lógica de compilação em um *script* de compilação Gradle.

Características

- **Orientado pela comunidade e apoiado por empresas**

Gradle é gratuito e é fornecido com a Licença Apache 2.0.

Projetos como Groovy e Hibernate mudaram completamente suas construções para Gradle.

Todo projeto Android vem com o Gradle como o sistema de compilação padrão.

Características

- **Orientado pela comunidade e apoiado por empresas**

Companhias como Orbitz, EADS e Software AG também abraçaram Gradle.

Framework Spring e Grails, são construídos literalmente com base em Gradle.

Plugins Gradle

O Gradle no seu núcleo fornece intencionalmente muito pouco para a automação do mundo real.

Todos os recursos úteis são adicionados através de *plugins*.

A aplicação de um *plugin* permite ampliar os recursos do projeto:

- Estender o modelo Gradle;
- Configurar o projeto de acordo com as convenções;
- Aplicar configuração específica.

Plugins Gradle

Ao aplicar *plugins* é possível colher uma série de benefícios:

- Promove a reutilização e reduz a manutenção de lógica semelhante em vários projetos;
- Permite um maior grau de modularização;
- Encapsula a lógica imperativa e permite que os *scripts* de compilação sejam declarativos.

Plugins Gradle

- **Tipos de *plugins*:**
 - ***Plugins de script***

São *scripts* de compilação adicionais que geralmente implementam uma abordagem declarativa para manipular a compilação.

São usados dentro de uma compilação, embora possam ser externalizados e acessados a partir de um local remoto.

Plugins Gradle

- **Tipos de *plugins*:**
 - ***Plugins* binários**

São classes que implementam a interface `Plugin` e adotam uma abordagem programática para manipular a compilação.

Podem residir dentro de um *script* de compilação, dentro da hierarquia do projeto ou externamente em um jar de *plugin*.

Plugins Gradle

- **Uso de *plugins*:**

Para usar a lógica de compilação encapsulada em um *plugin*, o Gradle precisa executar duas etapas.

Primeiro, ele precisa resolver o *plugin* e, em seguida, ele precisa aplicar o *plugin* ao alvo, geralmente um projeto.

Instalação do Gradle

Pré-requisito: instalação do JDK com uma versão 1.7 ou superior.

Posteriormente faça download da versão 4.1 no endereço <https://gradle.org/releases/>

Instalação do Gradle

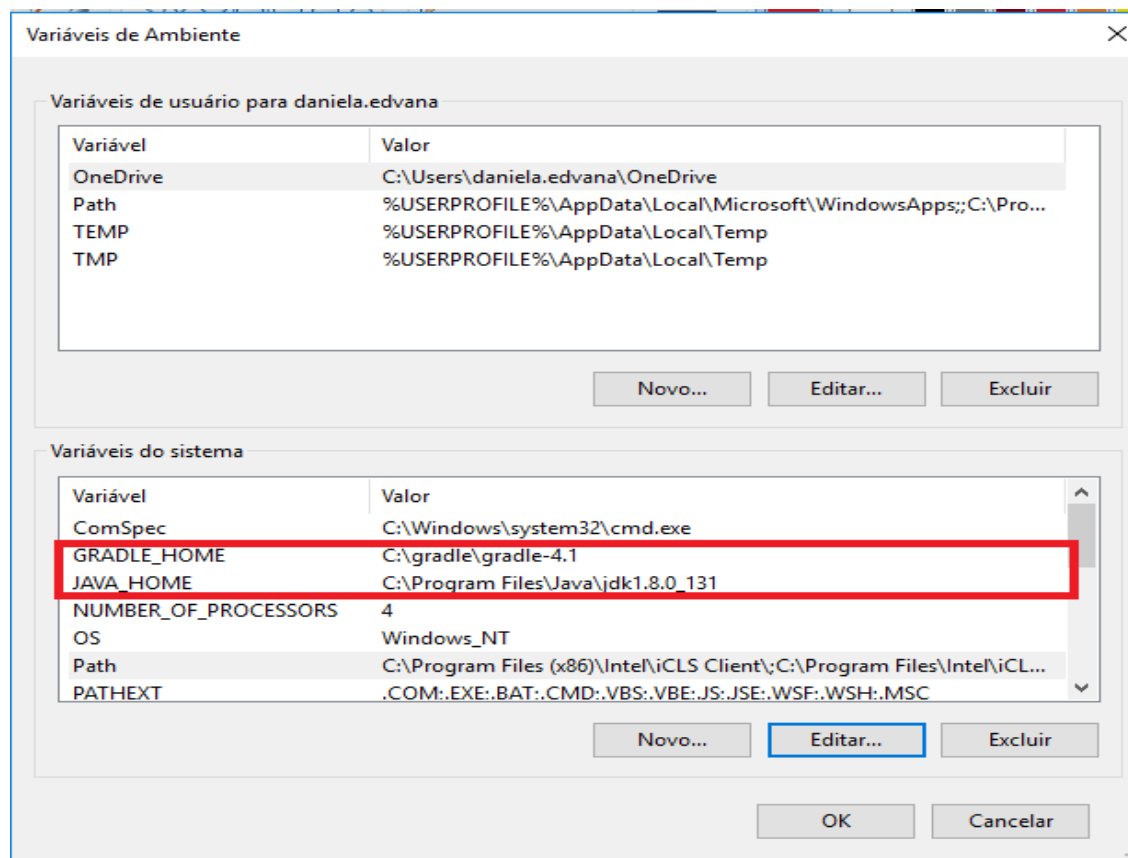


Figura 2 – Criando a variável de ambiente GRADLE_HOME no Windows

Instalação do Gradle

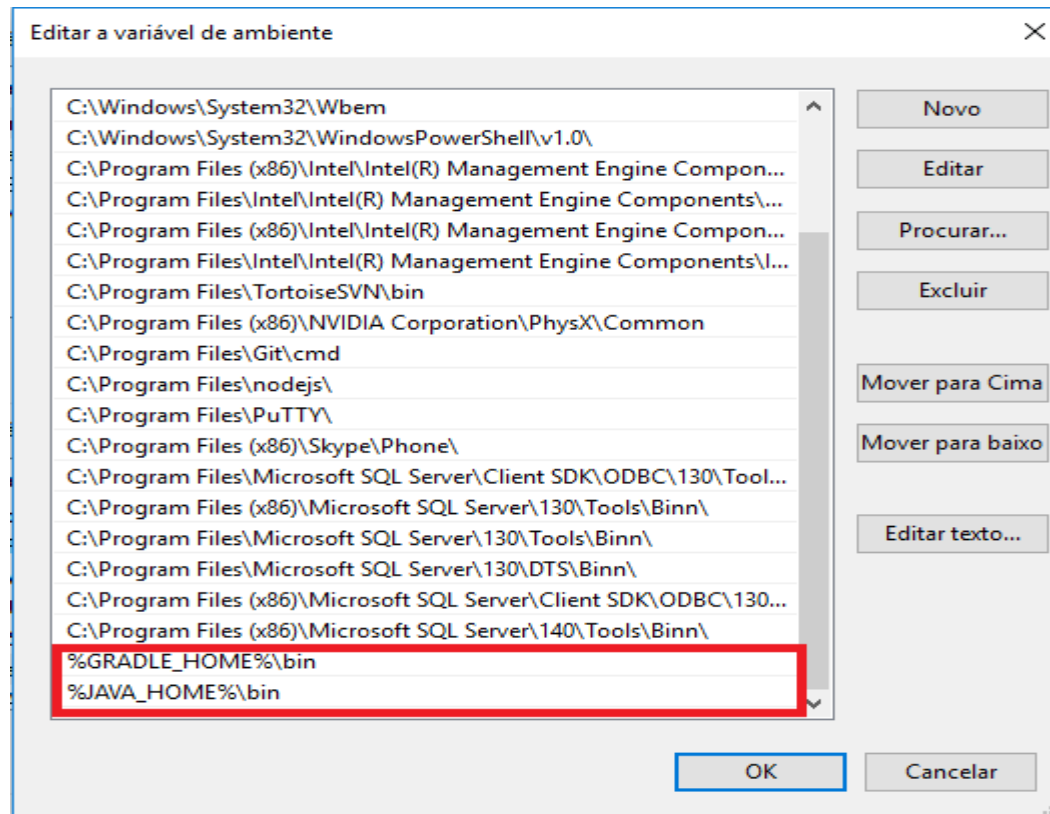
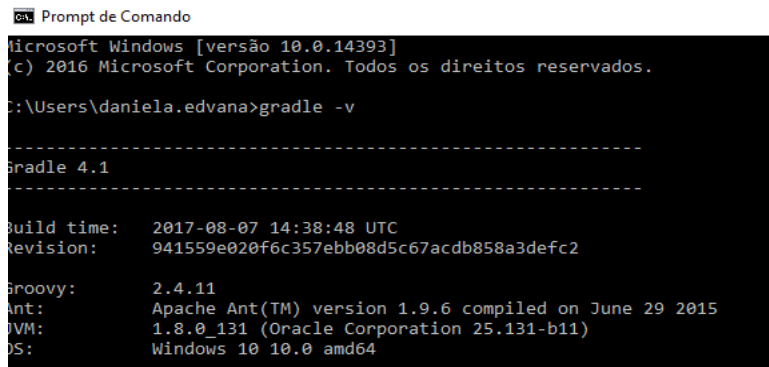


Figura 3 – Atualizando a variável de ambiente PATH no Windows

Instalação do Gradle

Para verificar se o Gradle foi instalado corretamente execute o comando:

```
$ gradle -v
```



```
cmd Prompt de Comando
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\daniela.edvana>gradle -v

-----
Gradle 4.1
-----

Build time:   2017-08-07 14:38:48 UTC
Revision:    941559e020f6c357ebb08d5c67acdb858a3defc2

Groovy:      2.4.11
Ant:         Apache Ant(TM) version 1.9.6 compiled on June 29 2015
JVM:         1.8.0_131 (Oracle Corporation 25.131-b11)
OS:          Windows 10 10.0 amd64
```

Figura 4 - Informação sobre a instalação do Gradle

Comandos Gradle

- `-?`, `-h`, `--help`: imprime todas as opções de linha de comando disponíveis.
- `-b`, `--build-file`: Opção para executar um *script* de compilação com um nome diferente. Ex: `gradle -b test.gradle`.
- *offline*: Esta opção é usada para executar sua compilação no modo *offline*

Comandos Gradle

- Opções de Log

- -i, --info: Esta opção é usada para obter mais mensagens informativas, alterando o *logger* Gradle para o nível de log INFO.
- -s, --stacktrace: A opção -s imprime um rastreamento abreviado da *stack trace* se uma exceção for lançada.
- -q, --quiet: reduz as mensagens de log de uma execução de compilação apenas para mensagens de erro.

Comandos Gradle

- Ajuda
 - *tasks*: exibe todas as tarefas executáveis do seu projeto, incluindo suas descrições.
 - *properties*: emite uma lista de todas as propriedades disponíveis no projeto.

Usando o Gradle no Eclipse IDE

- Instalação do Gradle (Buildship) via Eclipse Marketplace

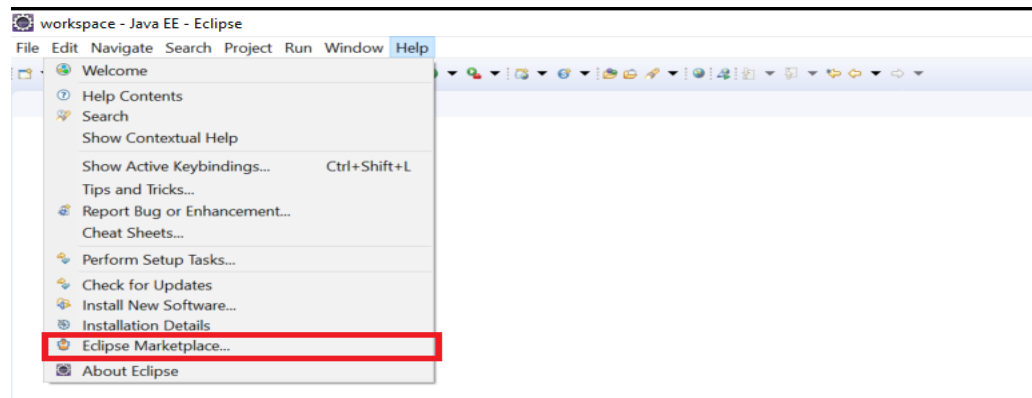


Figura 5- Eclipse Marketplace

Usando o Gradle no Eclipse IDE

- Instalação do Gradle (Buildship) via Eclipse Marketplace

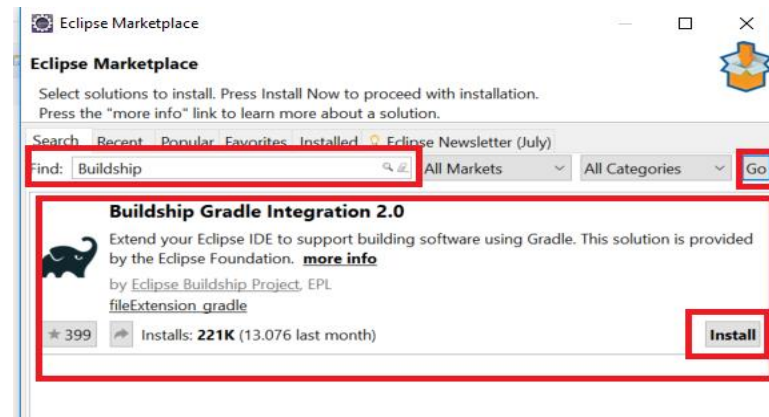


Figura 6- Busca da ferramenta Buildship no Eclipse Marketplace

Usando o Gradle no Eclipse IDE

- Criando projetos Gradle

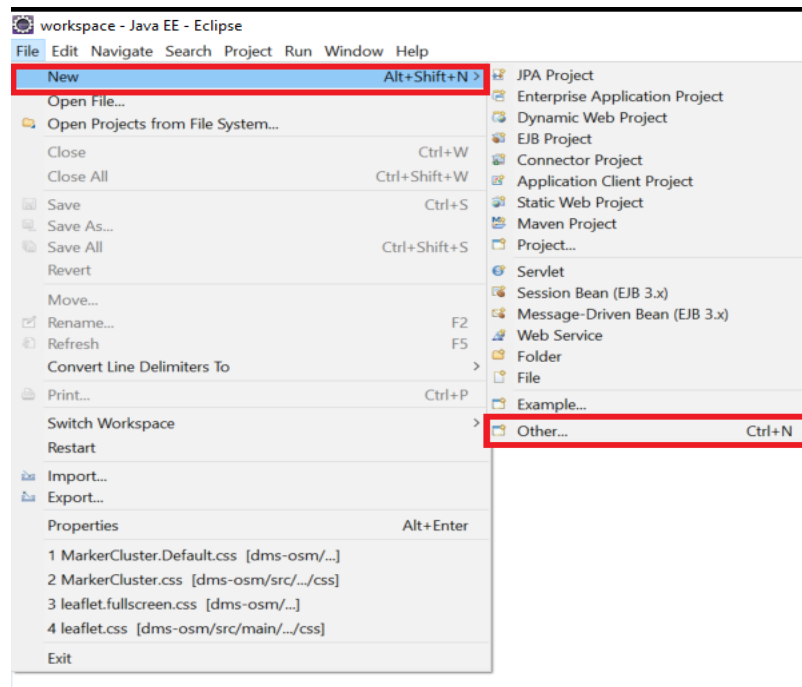


Figura 7 - Criação de um projeto Gradle

Usando o Gradle no Eclipse IDE

- Criando projetos Gradle

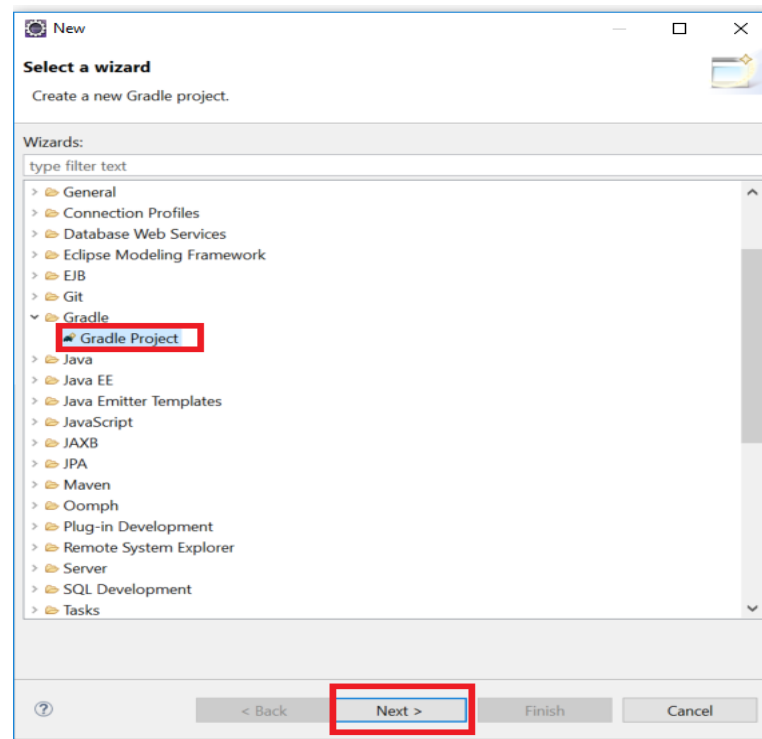


Figura 8 - Criação de um projeto Gradle

Usando o Gradle no Eclipse IDE

- Criando projetos Gradle

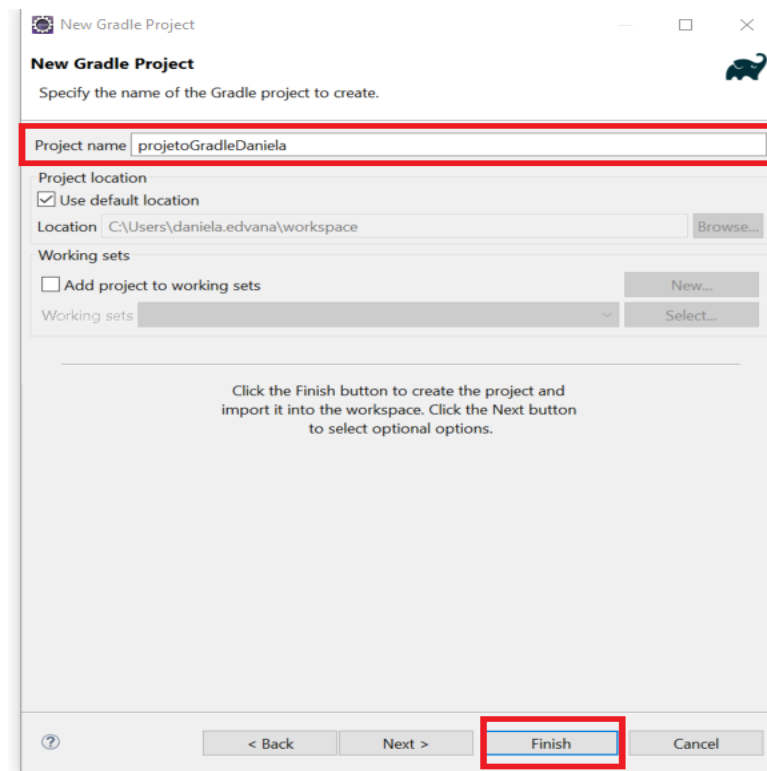


Figura 9 - Criação de um projeto Gradle

Usando o Gradle no Eclipse IDE

- **Prática 1 - Gerenciando dependências**

Usando o Gradle no Eclipse IDE

- **Prática 2 - Criando subprojetos com Gradle**

Usando o Gradle no Eclipse IDE

- **Prática 3 - Criando projeto web com Gradle**

Solução dos exercícios

<https://github.com/eltonbarbosa/java/tree/base-exemplos/base-exemplos>

Referências

- **Create a Gradle Java Web Application and run on Gradle Tomcat Plugin.** Disponível em <<http://o7planning.org/en/11247/create-a-gradle-java-web-application-and-run-on-gradle-tomcat-plugin>> Acesso em 15 Ago. 2017.
- **Gradle Build Tool.** Disponível em <<https://gradle.org/>>. Acesso em 15 Ago. 2017.
- Muschko, Benjamin. **Gradle in Action.** Manning Publications, 2014.
- Scholz, Simon.; Lars, Vogel. Using the Gradle build system in the Eclipse IDE – Tutorial. Disponível em <<http://www.vogella.com/tutorials/EclipseGradle/article.html#eclipse-gradle-support>> Acesso em 15 Ago. 2017.