

# Comparação de desenvolvimento de aplicações web com ASP.NET MVC 5 e Spring MVC 4

José Rafael Vasconcelos Cavalcante

January 5, 2011

## 1 Introdução

## 2 Configuração de ambiente de desenvolvimento

Neste capítulo será demonstrado o preparo do ambiente de desenvolvimento em um computador rodando o sistema operacional *Windows* 8.1 de 64 *bits* . Primeiramente, instala-se um sistema gerenciador de banco de dados para trabalhar tanto com a plataforma *Java \ Spring MVC* quanto com a plataforma *ASP.NET MVC 5* . O banco de dados usado será o *MySQL Community Server* versão 5.6.22 (a versão mais atual até o momento de criação desta monografia). A *Integrated Development Environment* ( *IDE* ) utilizada para escrever código em *Java* , será o *Eclipse Luna* . O *Gradle* será usado como *build tool* e o *Apache Tomcat* como *container Java Enterprise Edition* ( *JEE* ). Para desenvolver em *C#* , será usado o *Visual Studio 2013 Community* . Considerando que o leitor já possui entendimentos sobre informática necessários para instalar programas no *Windows* , as instruções de instalação serão sucintas.

### 2.1 Instalação do MySQL

O download do instalador do MySQL Community foi feito no seguinte endereço <https://dev.mysql.com/downloads/windows/installer/5.6.html>. O instalador está disponível duas versões, web installer e off-line installer. O web installer é um arquivo pequeno que quando executado irá baixar os arquivos do MySQL para a máquina, o off-line installer é maior e vem com todos os arquivos necessários para a instalação do MySQL. Qualquer que seja o método de instalação escolhido, eles terão as mesmas opções.

Executando o instalador, é escolhida a opção Custom e na árvore de opções que aparecerá na tela a seguir, são escolhidos o MySQL Server, o MySQL Workbench, o

Connector/J (para Java) e o Connector/NET (para .NET), como mostrado na figura 1. Pode acontecer do instalador pedir para instalar o Microsoft Visual C++ 2013 como dependência do MySQL Workbench, se isso acontecer, o próprio instalador proverá um botão para instalar essa dependência.

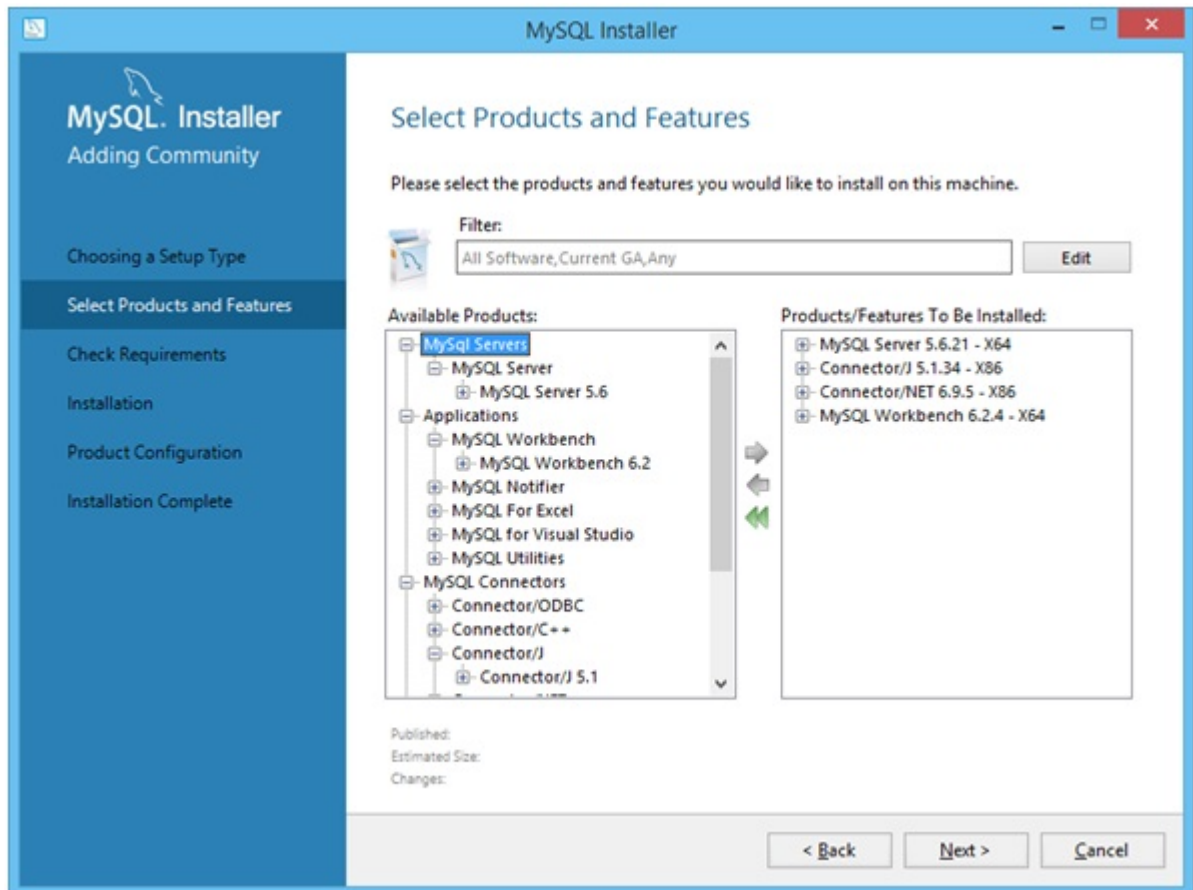


Figura 1: Opções de instalação do *MySQL*

Concluída a instalação, é hora de configurar o serviço do MySQL. Deixa-se selecionado o tipo de configuração como Development Machine e as configurações de rede padrão (protocolo TCP/IP, porta 3306). Quando for necessária a senha do usuário Root, será usada “1234”, é uma senha fraca que não se recomenda usar em ambiente de produção, mas serve para propósito de exemplo. Finaliza-se a configuração deixando marcados os restantes das opções de configuração como padrão do instalador.

Com o objetivo de testar o sucesso da instalação, o desenvolvedor pode executar o MySQL Workbench, como ilustrado na figura 2, e tentar se conectar à instância do MySQL.

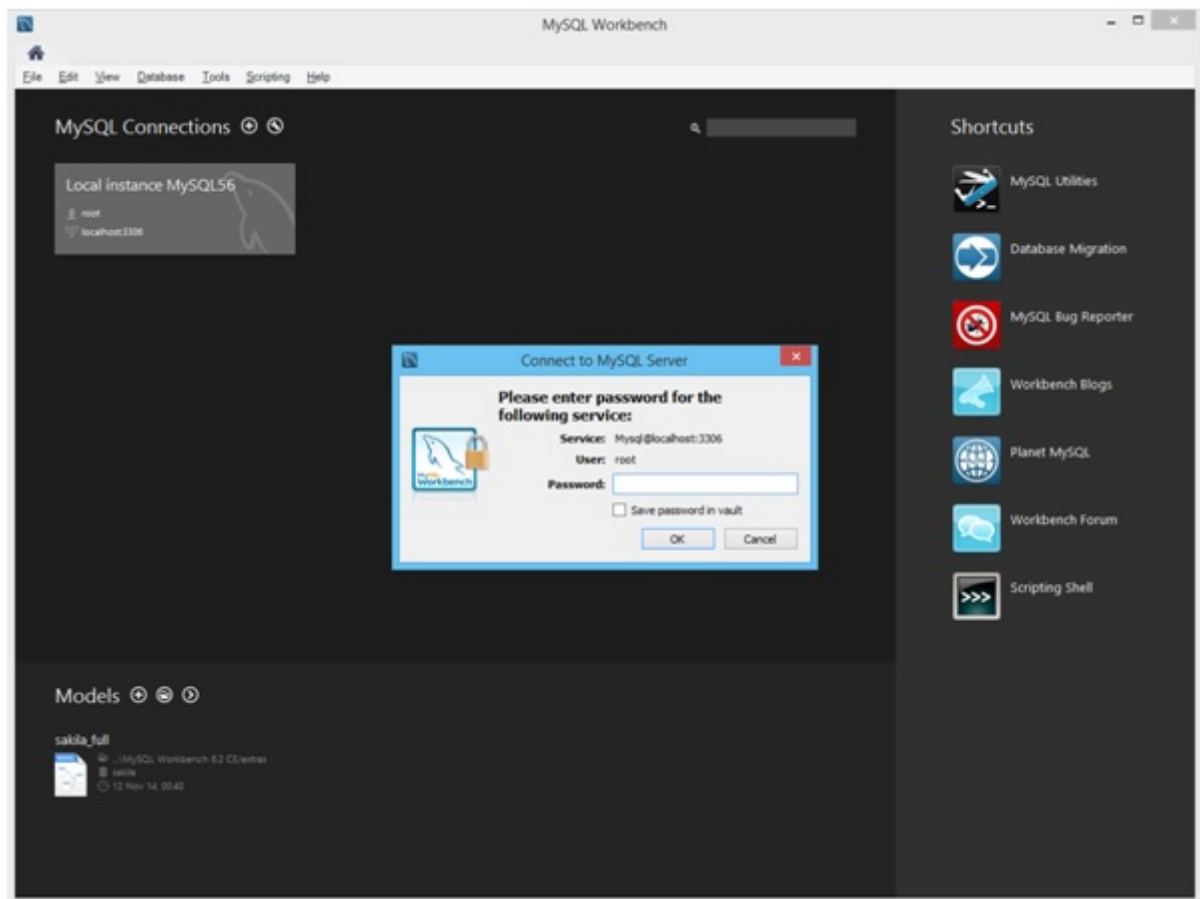


Figura 2: O *MySQL Workbench*

Para mais informações sobre o MySQL, visite a página oficial do projeto, <https://www.mysql.com/>.

## 2.2 Preparando o ambiente Java

Para desenvolver em Java, será utilizado o Eclipse Luna e o Java Development Kit 8 (JDK 8). Será usado o Gradle como build tool através de um plugin do Eclipse e o servidor web utilizado será o Apache Tomcat.

### 2.2.1 Instalando JDK 8

O instalador do JDK 8 pode ser adquirido no endereço <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Existem diversas versões para diversos sistemas operacionais, será usado nesse trabalho a versão para Windows de 64 bits.

Para fazer a instalação do JDK, foi executado o arquivo de instalação seguindo as

suas instruções. A única configuração possível durante a instalação é a mudança da sua pasta de destino, mas será mantido o diretório padrão como mostrado na figura 3.

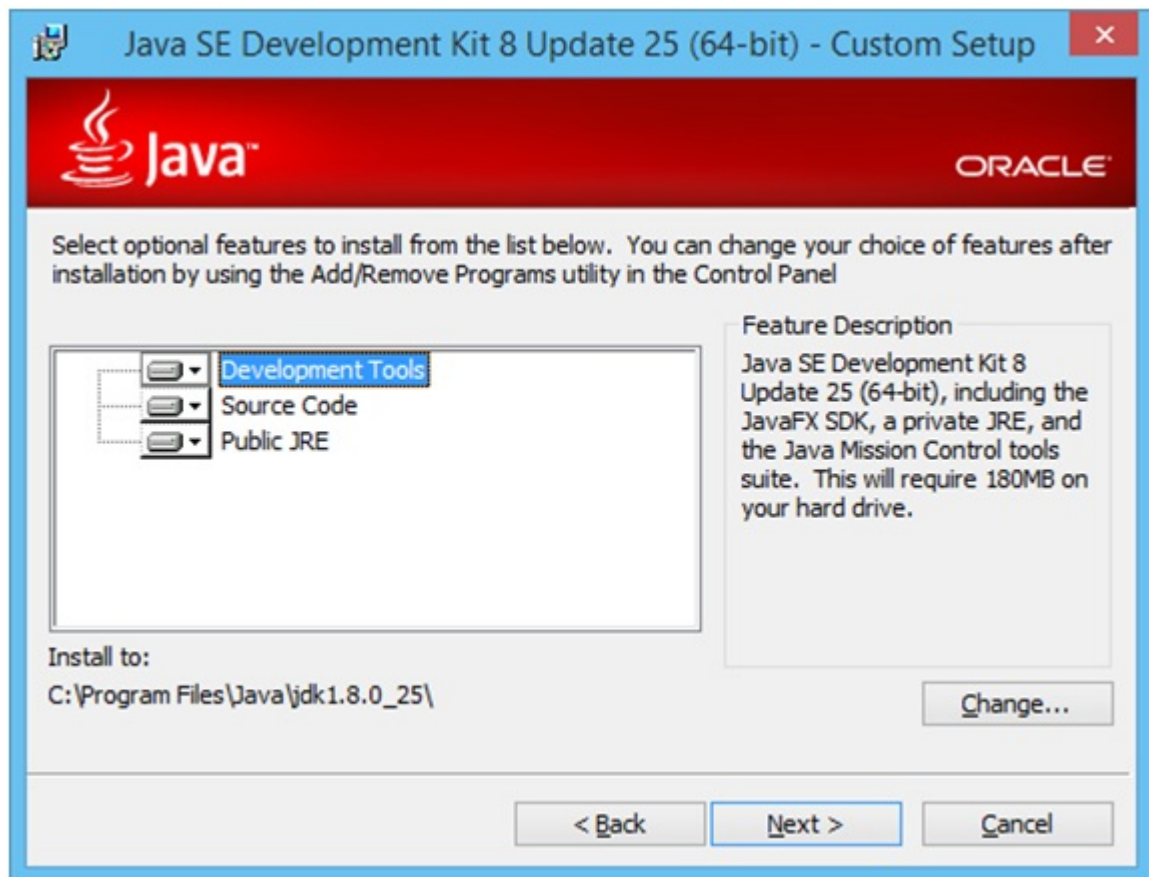


Figura 3: O instalador do JDK 8

Terminada a instalação, é necessário configurar a variável PATH para que o sistema encontre os arquivos do Java. Essas opções de configuração estão no painel de controle do Windows, no caminho Sistema/Configurações avançadas do sistema/-Variáveis de ambiente. Na janela de variáveis do sistema, é editada a variável PATH. Se adiciona o caminho onde o JDK foi instalado acrescido da pasta bin (C:\Program Files\Java\jdk1.8.0\_25\bin) como mostrado na figura 4.

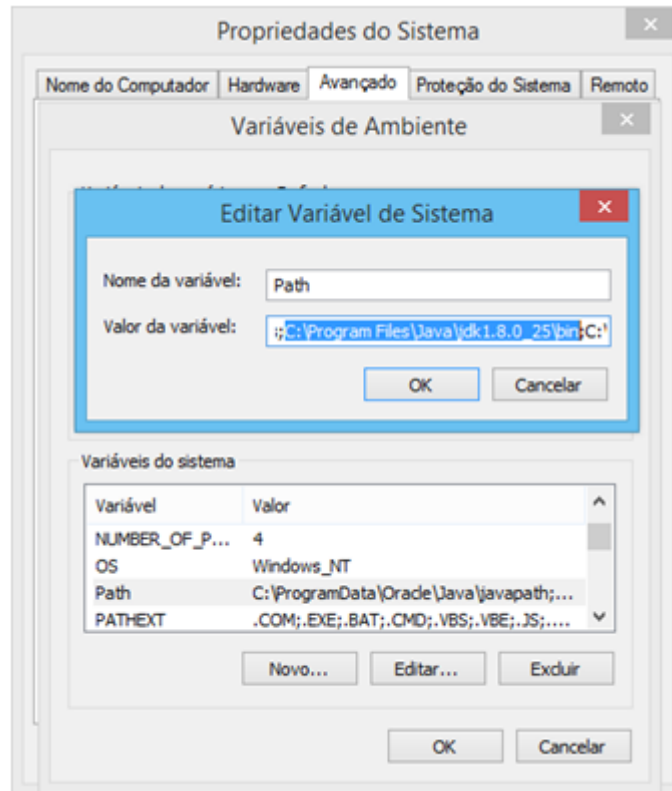


Figura 4: Configurando a variável PATH

Para testar se tudo foi instalado corretamente, abre-se uma janela do prompt de comando e digita-se o comando “java –version” (sem aspas). Se não existir problemas, será exibida na tela o número da versão do JDK instalado. Se isso não acontecer, é aconselhável desinstalar o JDK e repetir o processo de instalação.

### 2.2.2 Instalando o Eclipse Luna

O Eclipse Luna para Desenvolvedores Java EE é encontrado no endereço <https://www.eclipse.org/downloads/>. Terminando o download, a pasta “eclipse” pode ser descompactada para qualquer diretório do computador. Coloca-se um atalho na sua área de trabalho para o executável do Eclipse (eclipse.exe) para facilitar o acesso.

Na primeira vez que o Eclipse é executado será exibida uma janela para configurar o Workspace padrão (uma pasta onde serão guardados projetos e configurações).

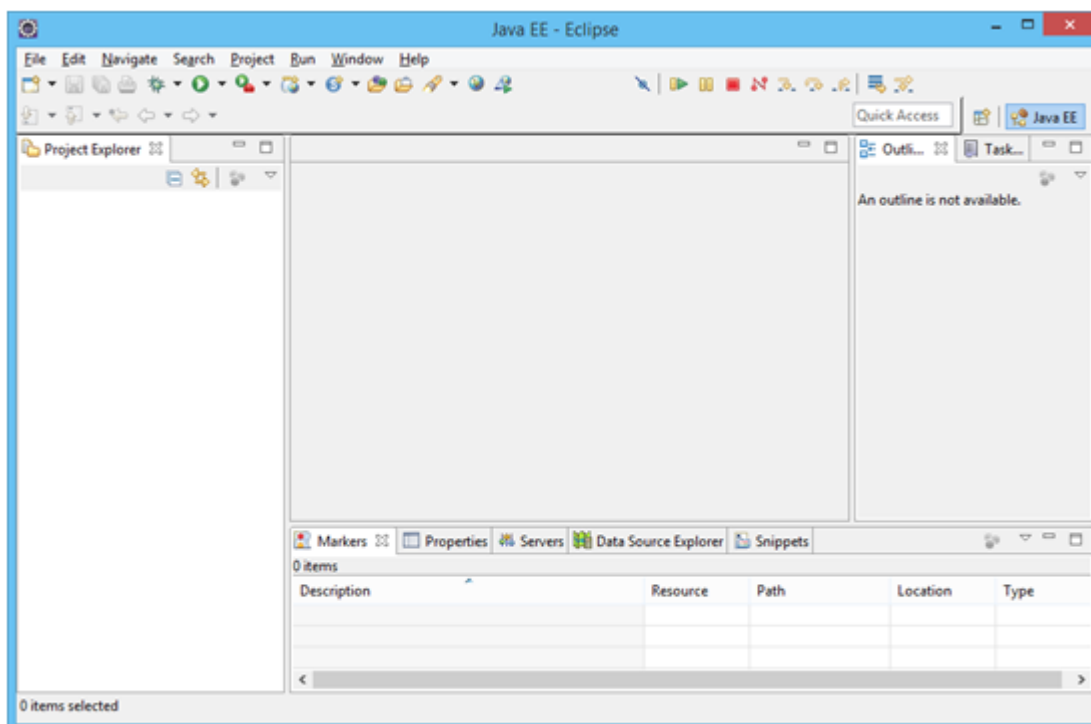


Figura 5: Eclipse recém instalado

### 2.2.3 Instalando o plugin do Gradle

O Gradle é a build tool que será utilizada nos exemplos do projeto Java. Ele faz o mesmo trabalho que o ANT junto com o Ivy ou o Maven fazem, mas ele é considerado por alguns autores como o que existe de mais moderno em se tratando de build tools, pois ele usa a linguagem Groovy em vez de XML para os scripts de builds, e permite configuração que o Maven não permite. O Gradle está presente em todo ciclo de vida do software (ele gera artefatos, roda teste unitários, resolve dependências e executa integração contínua), mas será usada apenas uma pequena parte do que ele pode oferecer. Para mais informações sobre o Gradle acesse <https://www.gradle.org/>.

No Eclipse Marketplace (repositório de plugins do Eclipse), faz-se uma pesquisa por “Gradle” na barra de buscas. Entre os resultados está o Gradle IDE Pack, esse plugin será usado nos exemplos desse trabalho. O Eclipse pede confirmação de todos os pacotes necessários para a instalação, todos são selecionados e instalados. Aceita-se os termos de uso do Gradle e quando aparecer uma janela de alerta confirmando a instalação, clica-se em OK. Quando a instalação terminar, o Eclipse é reiniciado.

Para verificar se o plugin foi instalado com sucesso, a pasta Gradle deve aparecer na árvore de tipos de projetos, no menu de novos projetos.

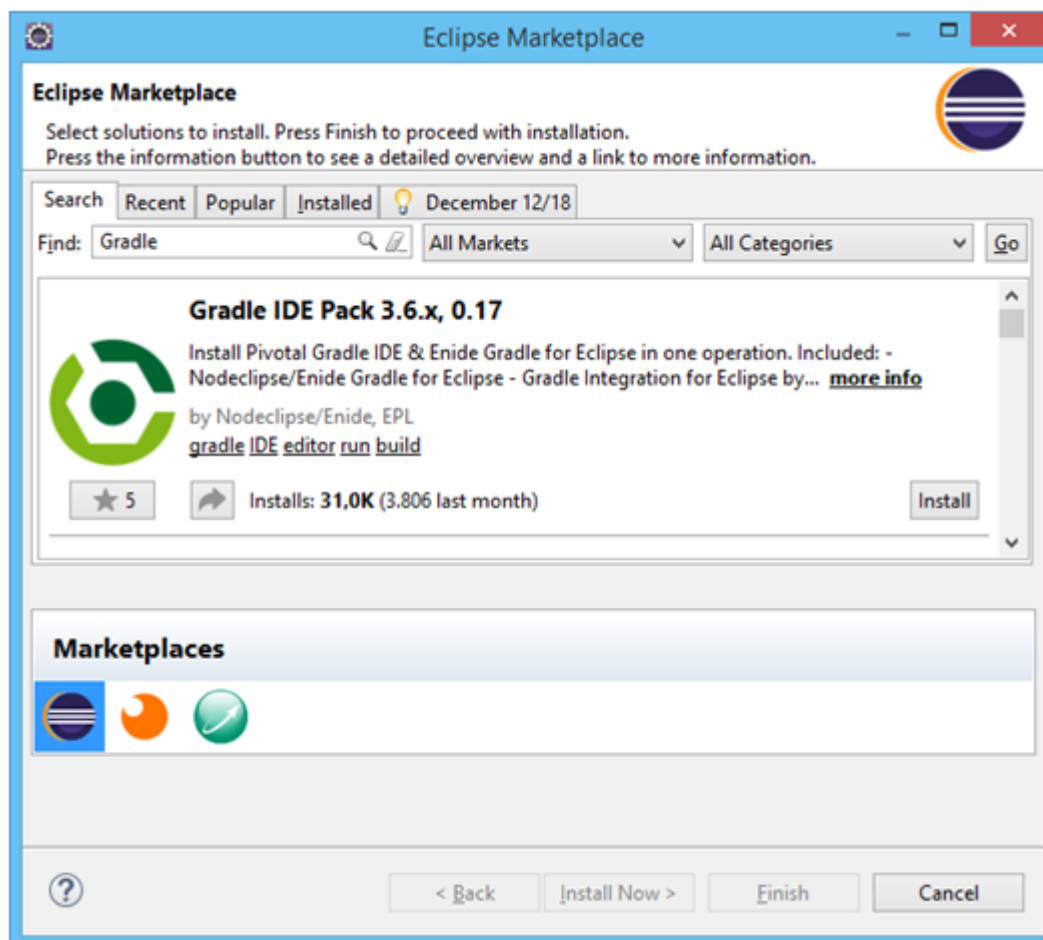


Figura 6: Instalando o plugin do Gradle

#### 2.2.4 Instalando o Apache Tomcat 8 como servidor de desenvolvimento do Eclipse

O Java EE é um conjunto de especificações que precisam ser implementadas por um container (um servidor de aplicação ou servidor web) que irá efetivamente executar a aplicação. Existem diversos containers disponíveis no mercado, sendo o GlassFish o próprio container da Oracle. Por motivos de praticidade, será usado o Apache Tomcat. O Tomcat pode ser gerenciado pela aba de servidores do Eclipse.

O instalador do Tomcat 8 pode ser encontrado no endereço <https://tomcat.apache.org/download-80.cgi>. Para os exemplos, usa-se a distribuição para Windows de 64 bits no formato zip. A pasta `apache-tomcat-8.0.15` pode ser descompactada para qualquer diretório no disco rígido (como exemplo será usada a raiz do disco C:).

A na aba “Servers” do Eclipse possui um link auto descritivo para adicionar um novo servidor. Clicando no link e expandindo pasta “Apache”, é escolhido o Tomcat 8 na árvore de opções. Na janela seguinte, em “Tomcat installation directory”, o botão “browse...” é usado para escolher o caminho de instalação do Tomcat (`C:\apache-tomcat-8.0.15` no nosso exemplo). Clicando no botão “Finish”, o Tomcat está pronto para uso com o Eclipse.

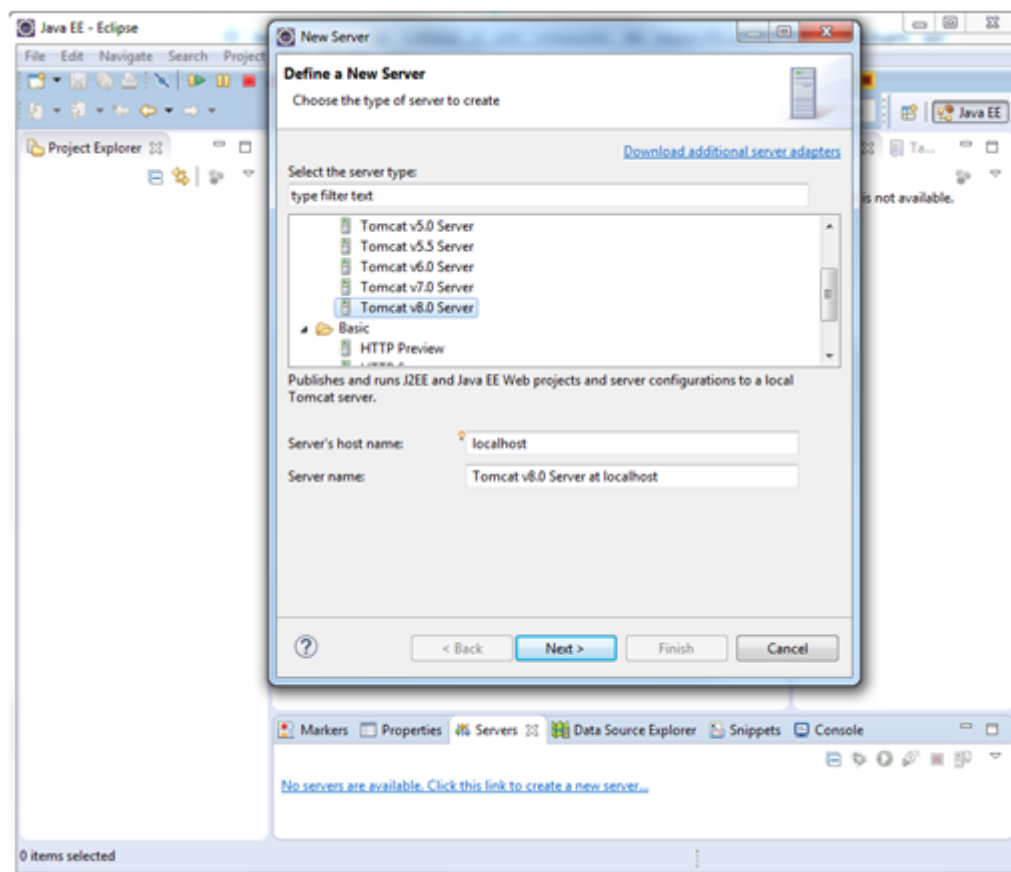


Figura 7: Janela pra adicionar servidores no Eclipse

## 2.3 Instalando o Visual Studio Community 2013

O instalador do Visual Studio Community 2013 pode ser encontrado no endereço <http://www.visualstudio.com/en-us/news/vs2013-community-vs.aspx>. Esse é um instalador online, ele irá baixar os arquivos do Visual Studio e opcionais selecionados à medida que a instalação for progredindo. Uma imagem do DVD de instalação off-line também está disponível na sessão de downloads do site <http://www.visualstudio.com>.



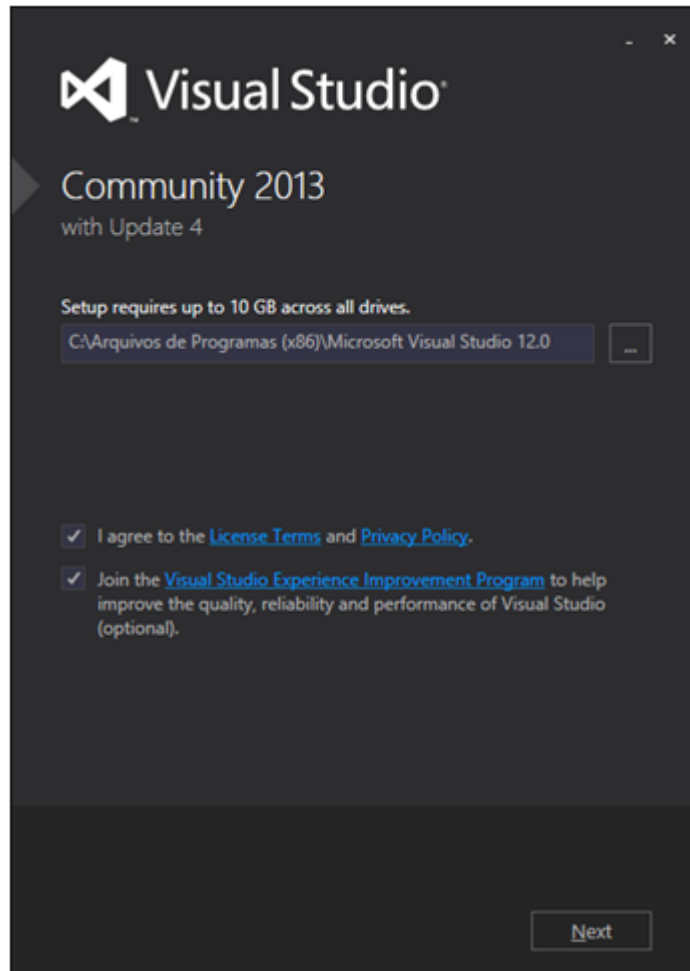


Figura 8: Instalador do Visual Studio Community 2013

Clicando no botão “Next”, a próxima tela o instalador irá exibir uma lista de componentes opcionais como o SDK do Windows Phone 8 e o Silverlight. Desses componentes opcionais, é aconselhável instalar pelo menos o Microsoft Web Developer Tools para facilitar o desenvolvimento de aplicações web.

Após a instalação, o desenvolvedor pode utilizar sua conta da Microsoft como perfil no Visual Studio e publicar suas aplicações no Microsoft Azure, mas isso é opcional. Quando se executa o Visual Studio pela primeira vez, o desenvolvedor pode escolher as opções de desenvolvimento e um esquema de cores que irá usar. Como exemplo, é escolhida a opção de desenvolvimento “Web Development”.

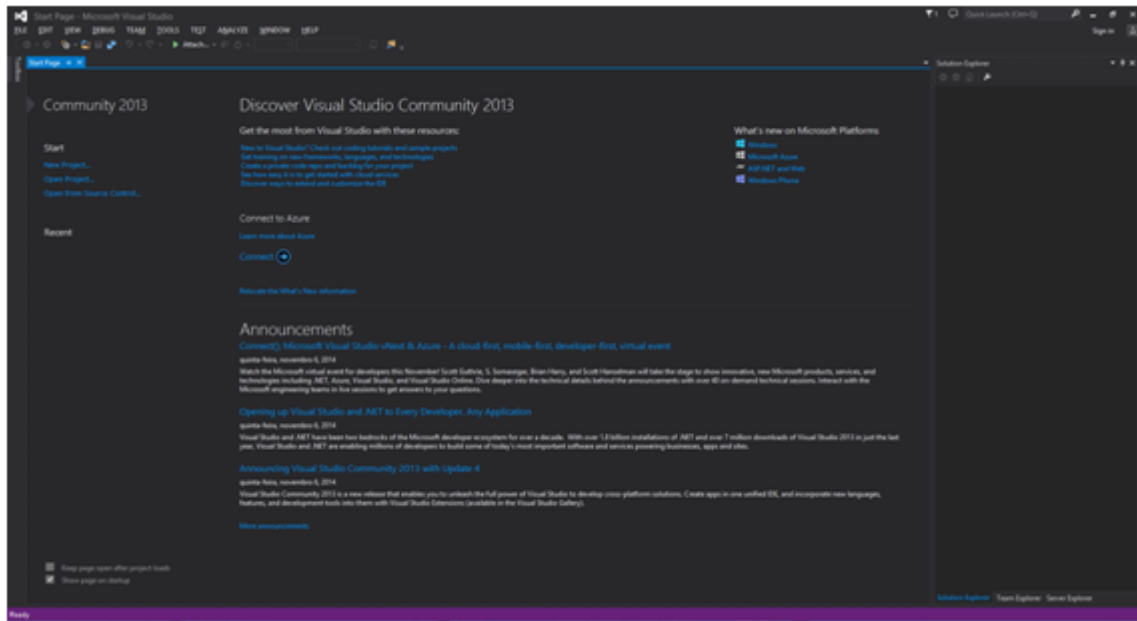


Figura 9: Tela inicial do Visual Studio Community 2013

O Visual Studio possui sua própria ferramenta de geração de builds (MsBuild), gerenciador de pacotes para obter bibliotecas de terceiros (Nuget) e um servidor de desenvolvimento minimalista para executar e depurar aplicações web.

## 2.4 Conclusão

A preparação de um ambiente de desenvolvimento Java requer mais passos, instalar o kit de desenvolvimento do Java, uma IDE (Eclipse), um servidor Java EE (Tomcat) e uma build tool (Gradle), enquanto o todo o software necessário para se desenvolver com .NET é adquirido em um único instalador.

As vantagens do ambiente Java é que ele dá ao desenvolvedor mais opções de como configurar seu ambiente, e o tamanho em megabytes do software necessário é consideravelmente menor do que o ambiente .NET. O desenvolvedor tem a liberdade de escolher outras build tools disponíveis no mercado (Ex: Ant, Maven), outras IDEs (Ex: Netbeans) e outros servidores Java EE (Ex: Jetty, Glassfish). O lado “negativo” dessa liberdade é que com tantas opções, o desenvolvedor tem que pesquisar mais sobre cada solução até decidir como vai montar seu ambiente de desenvolvimento, e depois de escolher quais produtos irá utilizar, pode ser que precise mais algum tempo estudando como eles interagem.

A vantagem do ambiente .NET é a facilidade de se ter todo o software necessário para o desenvolvimento em um único pacote, o .NET Framework, Visual Studio Community 2013 e demais ferramentas. A desvantagem é que esse pacote pode conter coisas que o desenvolvedor não precisa ou deseja, baixando arquivos desnecessários e tomando espaço em disco.

No próximo capítulo será mostrado como criar um projeto de uma aplicação web nas duas plataformas.

### 3 Criando projetos web no padrão MVC

Nesse capítulo será mostrado como se cria um novo projeto para uma aplicação web que utiliza o padrão MVC nas plataformas Java e .NET. Para Java será usado o Spring Framework, que cuidará da arquitetura MVC e injeção de dependências, e o Hibernate, que cuidará da persistência e acesso a dados. Na plataforma .NET será utilizado o ASP.NET MVC 5, que cuida de toda estrutura de uma aplicação web MVC, o Entity Framework 6 para acesso a dados e o Ninject para injeção de dependências.

#### 3.1 Criando um projeto do Gradle no Eclipse

O modelo de projeto usado nos exemplos desse trabalho será o Gradle Project. Esse modelo de projeto está localizado na pasta Gradle na árvore de novos projetos do Eclipse.

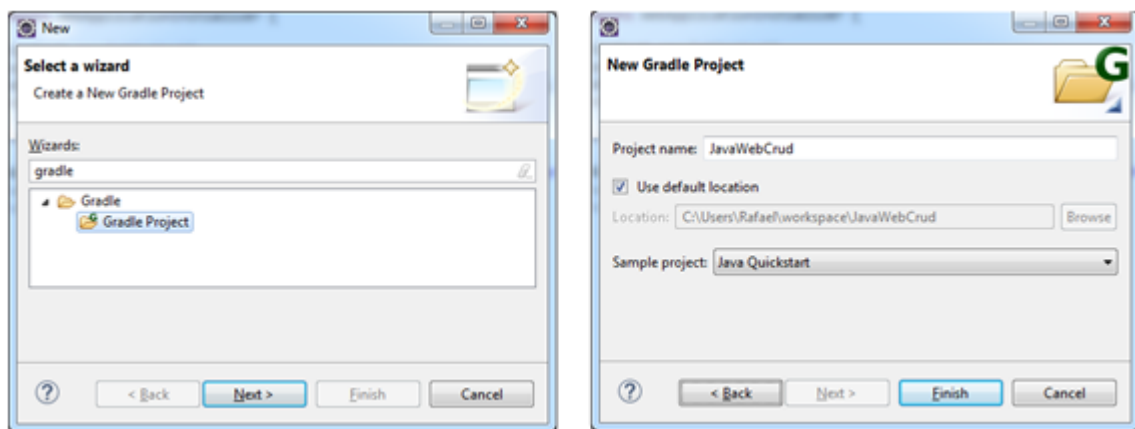


Figura 10: Criando um projeto do Gradle

É criado um projeto com a seguinte estrutura:

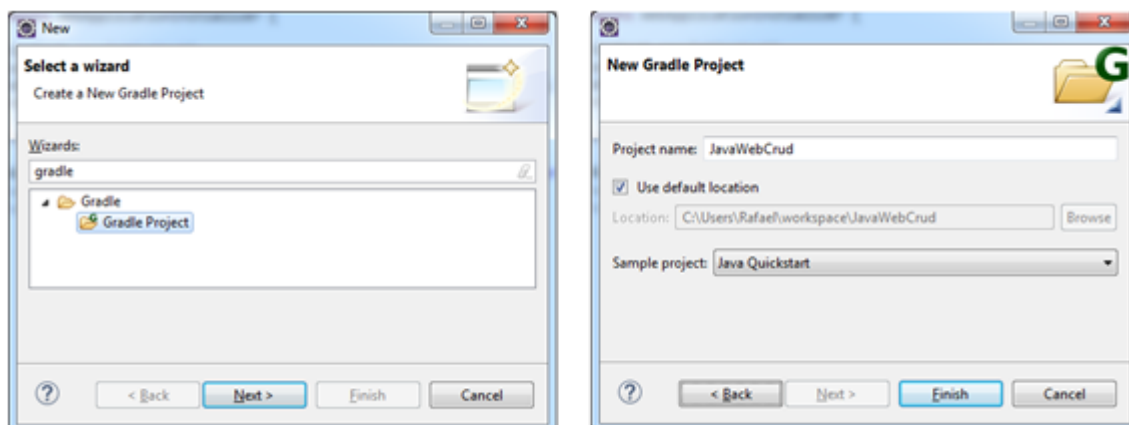


Figura 11: Estrutura inicial de um projeto do Gradle

As pastas “src/main/java” e “src/main/resources” devem armazenar, respectivamente, código fonte Java e recursos utilizados na aplicação. As pastas “src/test/java” e “src/test/resources” são utilizadas para testes unitários. As pastas de testes não serão utilizadas nos nossos exemplos e serão removidas. A pasta build é para onde irão todos os artefatos gerados pelo projeto.

O arquivo build.gradle é o arquivo de definição de projeto do Gradle. Nele podem ser criados scripts para controlar a construção de artefatos, baixar bibliotecas de terceiros, configurar testes unitários e realizar várias outras tarefas. Os scripts do Gradle são escritos em Groovy, outra linguagem compatível com a JVM.

```

apply plugin: 'java'
apply plugin: 'eclipse'

sourceCompatibility = 1.5
version = '1.0'
jar {
    manifest {
        attributes 'Implementation-Title': 'Gradle Quickstart', 'Implementation-Ver
    }
}

repositories {
    mavenCentral()
}

dependencies {
    compile group: 'commons-collections', name: 'commons-collections', version: '3.
    testCompile group: 'junit', name: 'junit', version: '4.+'
}

test {
    systemProperties 'property': 'value'
}

```

```
uploadArchives {  
  repositories {  
    flatDir {  
      dirs 'repos'  
    }  
  }  
}
```