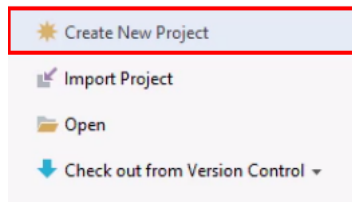


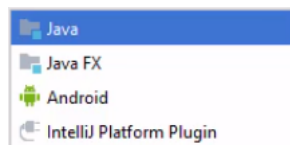
Conhecendo os atalhos

Transcrição

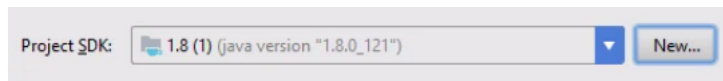
O primeiro passo é abrir o **IntelliJ IDEA Community**. Após o término do carregamento, teremos a tela inicial com algumas opções. Como estamos iniciando com a ferramenta, vamos selecionar a opção de criar novo projeto.



Selecionando a opção *Create new Project*, uma nova janela será aberta onde podemos configurar o projeto. Vamos selecionar a opção **Java**, nas opções de projeto.

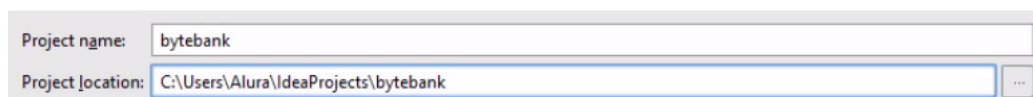


Em **Project SDK** deixamos como a versão java **1.8**. Caso essa opção não esteja aparecendo em seu projeto, basta clicar em "**New**" e selecionar o diretório de instalação do seu **JDK**. Após essa configuração basta clicar em "**Next**"



Na nova janela, o **IntelliJ** pergunta se gostaríamos de criar o nosso projeto baseado em algum **Template**, como não vamos usar nenhum **Template** basta clicar em "**Next**".

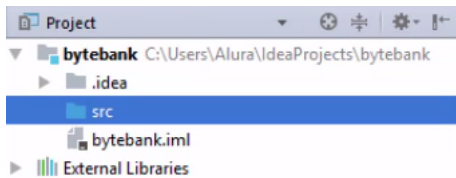
Agora apareceu a opção de nomear o projeto. Durante o curso vamos trabalhar em um projeto de uma banco digital, então vamos nomeá-lo no campo **Project name** como "**bytebank**". Em **Project location** fica o caminho onde se encontra nosso projeto.



Clicando em "**Finish**" o nosso projeto vai começar a carregar, inclusive aparecerá um barra de progresso de indexação dos arquivos. Isso é comum quando estamos criando ou carregando o projeto pela primeira vez, a IDE vai indexar os arquivos para ganhar mais desempenho durante o desenvolvimento do projeto.

Com tudo carregado, podemos notar que temos uma **view** chamada "**Project**", onde podemos visualizar toda a estrutura das pastas do projeto.

O primeiro diretório é o próprio nome de projeto `bytebank`. Dentro, possuímos o diretório `.idea` onde ficam todas as configurações do **IntelliJ**, está pasta não é o foco do nosso curso. Também temos a pasta `src` (**source**) onde iremos colocar todas as nossas classes Java.

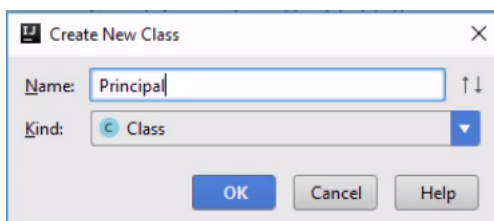


Para criar um classe, bastaria clicarmos com o botão direito do mouse na pasta `src` e selecionar a opção "New" e em seguida "Java Class". Porém a grande vantagem da IDE é poder fazer os mesmos passos com atalhos.

Para manipular a *view* do projeto, podemos usar o atalho "Alt + 1", dessa forma é possível ocultar e mostrar a *view*. Também é possível após o "Alt + 1", navegar pelos diretórios com as setas do teclado.

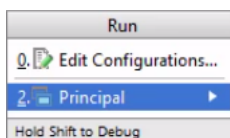
Selecionando com os atalhos a pasta `src`, podemos usar o atalho "Alt + insert", que permite criar algum arquivo. Como estamos na pasta `src` o *IntelliJ* vai mostrar as opções dos arquivos faz sentido estarem nesse diretório.

Selecionando a opção "Java Class", colocaremos o nome da classe de "Principal".



Mas como faremos para executa a classe? Nos cursos básicos de Java, aprendemos que para executar precisamos de uma método *main*. Poderíamos escrever toda a assinatura do método, porém a IDE também consegue nos auxiliar, basta escrevermos as iniciais `psvm` e apertamos a tecla "Tab" para que a assinatura seja escrita automaticamente.

Ainda não executamos a classe `Principal`, e existem diversas formas de executa-lá. Podemos clicar com o botão direito do mouse e selecionar a opção "Run Teste.main()". A segunda maneira é ir no caminho "Run > Run 'Teste'", mas ainda estamos usando o mouse. O terceira maneira é através de atalhos, clicamos na área do editor de código e usamos o "Ctrl + Shift + F10". Como é a primeira vez que estamos executando a classe, aparecerá um menu mostrando as classes que podemos executar, basta selecionar a classe `Principal` e clicar em "Enter".



O *IntelliJ* executou a nossa classe `Principal`, e abriu inclusive um Console informando que o Java foi executado e o processo foi finalizado.

```
Process finished with exit code 0
```

Porém, não foi apresentado nenhuma mensagem interessante para nós, e está certo, afinal não colocamos nenhuma instrução com alguma mensagem para que fosse exibida no Console.

Para exibirmos mensagem no Console, podemos usar o `System.out.println()`. Mas será que um comando como esse não teria uma forma mais simples de escrever? Sim, podemos escrever `sout` e pressionar a tecla "Tab" para que o comando seja completado. Agora podemos colocar a mensagem "Bem-vindo ao Bytebank".

```
public class Prncipal {  
  
    public static void main(String [] args){  
        System.out.println("Bem-vindo ao Bytebank");  
    }  
}
```

Podemos executar a classe novamente, mas como já executamos uma vez e o *IntelliJ* já indexou a nossa classe `Prncipal` como executável, podemos usar o atalho "Shift + F10". Agora recebemos no Console a mensagem de "Bem-vindo ao Bytebank".

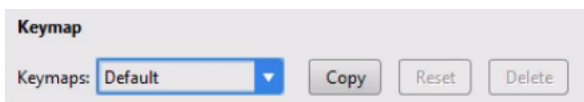
Mas você deve estar se perguntando: se eu não estivesse vendo os atalhos aqui no curso, como eu aprenderia? O primeiro pensamento provavelmente é seria internet.

Na internet realmente aprenderíamos muitas funcionalidades da ferramenta, mas o legal do *IntelliJ* é que ele permite acessarmos todos os atalhos baseados em nossa configuração.

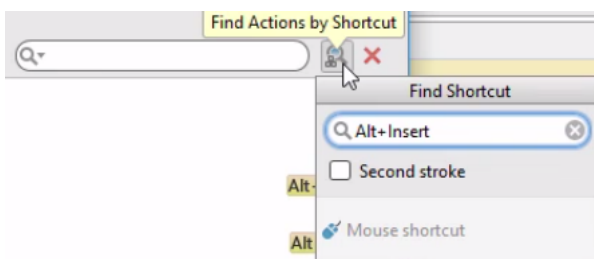
Mas como podemos buscar essa configuração? Podemos ir em "File > Settings" para chegarmos a todas as configurações do *IntelliJ*. No campo de busca, podemos digitar "**Keymap**" para filtrar a opção desejada.



Na seção de *Keymap* podemos ver a configuração usada na opção **Keymaps**, por padrão no Windows a configuração vem como **Default**, mas não se preocupe caso você esteja no Mac ou no Linux, durante o curso os atalhos serão ensinados nos três sistemas operacionais.



Do lado direito teremos o simbolo de uma lupa, que é a opção **Find Actions by Shortcut**. Clicando nessa opção, podemos usar os atalhos que a IDE nos retorna a ação correspondente.



Também temos ao lado da lupa, outro campo de busca. Nesse campo de busca podemos especificar a ação que queremos, por exemplo "new".



O *IntelliJ* irá nos retornar a listagem de todas as ações que contenha a palavra buscada e junto os atalhos correspondentes a ação.



Outra dica importante é que podemos ir em "Help > Keymap Reference" para que seja aberto um arquivo **PDF** com todos os atalhos e suas funções da configuração que estamos usando no momento. É bem interessante deixarmos a documentação aberta para que possamos consultar os atalhos de forma mais rápida.

