

## Primeira classe - Contas

### Transcrição

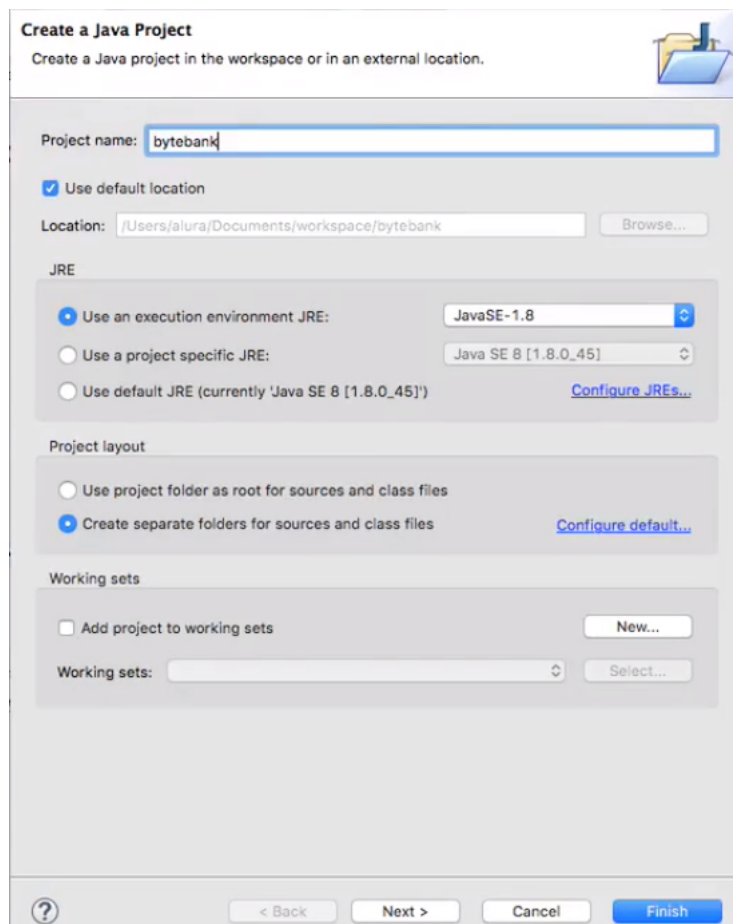
Faremos o **ByteBank**, um projeto de banco da **Alura**. Para isso, precisaremos criar o sistema que o compõe. Iremos descobrir quais dados serão necessários para esse sistema.

A menor unidade que iremos trabalhar é a *conta bancária*, e para isso, precisaremos fazer uma breve pesquisa para sabermos do que ela é composta.

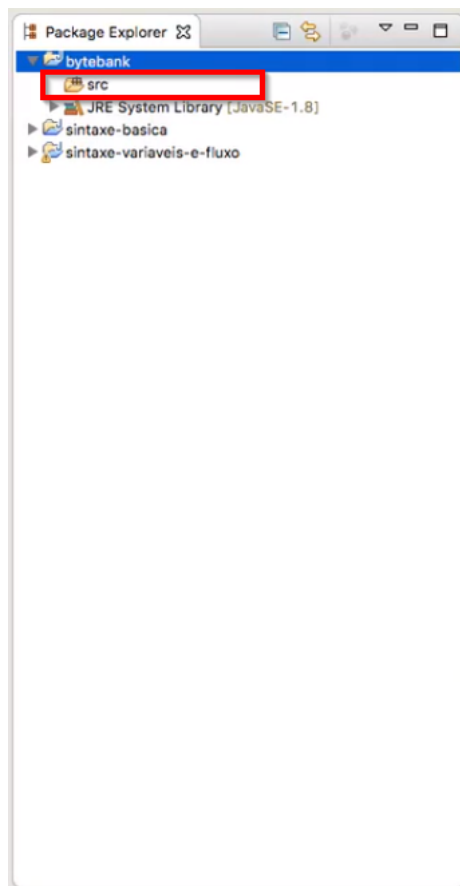
Se olharmos a definição de conta corrente no [WikiPedia \(https://pt.wikipedia.org/wiki/Conta-corrente\)](https://pt.wikipedia.org/wiki/Conta-corrente), descobriremos uma série de referências acerca de juros e taxas, mas a questão que iremos nos atentar é que as contas bancárias acionam informações diferentes para fins variados.

Façamos uma analogia com uma seguradora de veículos: é importante que ela tenha informações como **marca**, **modelo**, **chassi** e a **cor** do carro. Mas é irrelevante para a seguradora saber quantos cilindros o carro tem ou a versão do motor. Já para uma empresa que fabrica veículos, esses dados fazem parte do **domínio** do problema.

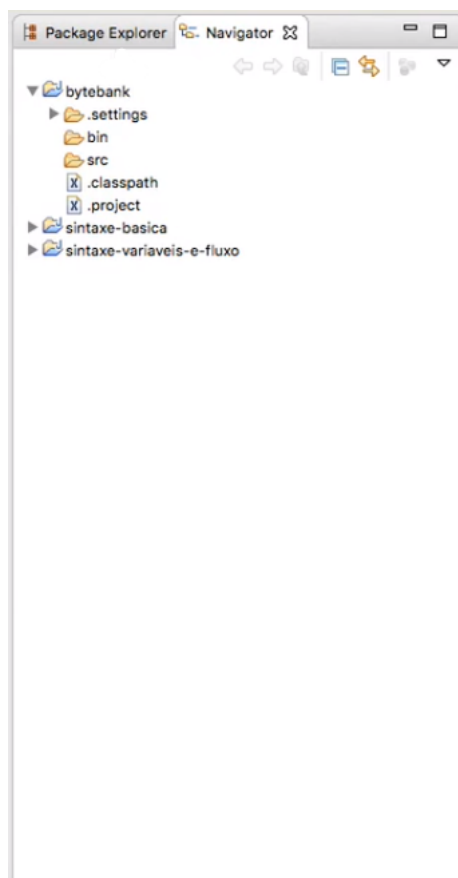
Criaremos um domínio para o nosso sistema baseado em um projeto de banco. No Eclipse, faremos um novo projeto Java intitulado `bytebank`. Depois, clique em "Finish".



Feito isso, será criado um novo projeto na área do *Package Explorer*. Ao clicarmos no projeto `bytebank`, veremos a pasta `src`.



*Lembrando:* você pode selecionar "Window > Show View > Navigator". O *Navigator* possui uma ideia parecida com o Windows Explorer ou Finder, nele você visualizará mais diretórios internos. O *Package Explorer* oculta a visualização desses diretórios.



Em `src`, criaremos alguns códigos para guardarmos os dados sobre nossas contas bancárias. Uma característica interessante do Java é que ele nos possibilita a criação de **tipos**. Em um "tipo" do Java, armazenaremos todas as informações que constituem uma conta bancária.

Usaremos o [Figma \(https://www.figma.com/\)](https://www.figma.com/) - um *software* de criação de interface do usuário - para definirmos as informações básicas de uma conta bancária.

A nossa conta possui quatro características importantes para o nosso banco: **saldo**, **agência**, **número**, **titular**.



Um diagrama de uma conta bancária no Figma. No topo, há um ícone de seta para cima e o texto "Conta". Abaixo, há uma tabela com quatro linhas e duas colunas. As colunas são rotuladas "saldo", "agencia", "numero" e "titular".

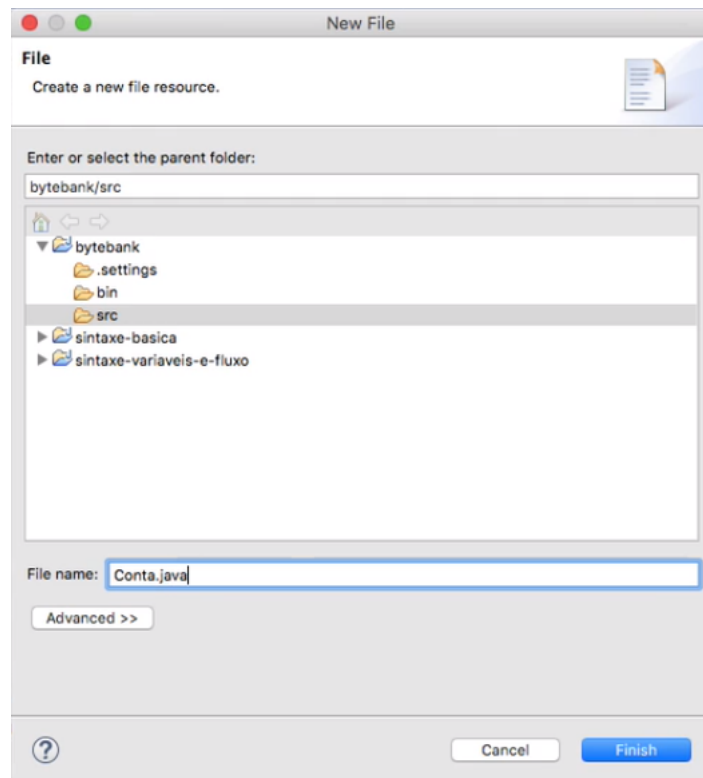
saldo	
agencia	
numero	
titular	

Veremos que no Java não serão guardados apenas os dados de conta bancária, mas também serão atribuídos comportamentos para a conta. Iremos definir algumas transações bancárias típicas, como realizar saques, depósitos e transferências.

O esquema que criamos no Figma pode ser reproduzido no Java, mas antes precisamos refletir: esse quadro com campos a serem preenchidos, pode ser caracterizado como uma conta? Não podemos realizar funções básicas de uma conta bancária apenas com esse quadro cinza, portanto, não se trata de uma conta, e sim, de uma especificação de um tipo `conta`.

Podemos fazer a seguinte analogia: a planta de uma casa não é uma casa, mas o esquema ou design de uma.

Recriaremos esse esquema de dados de conta bancária no Java. Em `src`, criaremos um novo arquivo chamado `Conta.java`.



Faremos a especificação dos componentes do tipo `Conta`.

```
tipo Conta:
    saldo
    agencia
    numero
    titular
```

Obviamente, essa não é a sintaxe de Java. Precisaremos utilizar as regras da linguagem para realizarmos com sucesso a especificação do tipo `Conta`. Escreveremos ao lado de `Conta` a palavra `class`. Trata-se de uma palavra-chave que define um tipo. Para indicarmos que abrimos e fechamos um bloco de informações usaremos as chaves, `{}`.

```
class Conta {
    saldo
    agencia
    numero
    titular
}
```

É também necessário declarar o tipo das outras categorias da conta. Para **saldo**, usaremos o tipo `double`, pois ele armazena dados de ponto flutuante.

Os componentes **agencia** e **numero** serão do tipo `int`. Para finalizar, **titular** será uma `String`, que guarda um conteúdo de texto.

Acionaremos o atalho "Ctrl + S" para salvarmos nosso trabalho. Com isso, já temos um código fonte válido em Java que já foi compilado. Ao observarmos a área *Navegador* e selecionarmos o diretório `bin`, veremos o arquivo `Conta.class`.

Porém, não podemos iniciar a execução do programa. Para que um programa possa ser iniciado em Java, ele precisa ter o ponto de entrada `public static void main(String[] args)`, a classe que criamos não possui esse ponto de entrada,

pois trata-se de um arquivo de suporte.

Em uma aplicação Java é comum existirem vários arquivos que se comunicam, o primeiro a ser executado contém o método `main` com a "assinatura" (tipo de retorno e parâmetros) acima.