

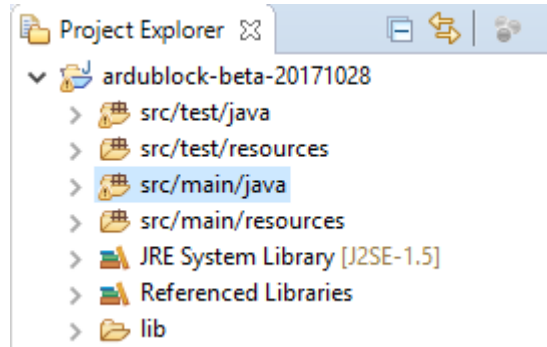
Tutorial para criar blocos para *Ardublock* no eclipse

Veículo Aquático de Baixo Custo

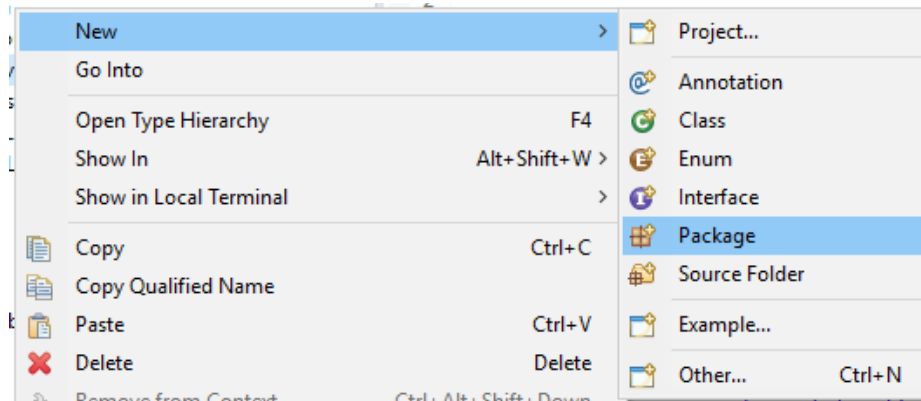
Bruno Silva, 2150634

Filipe Salgueiro, 2151226

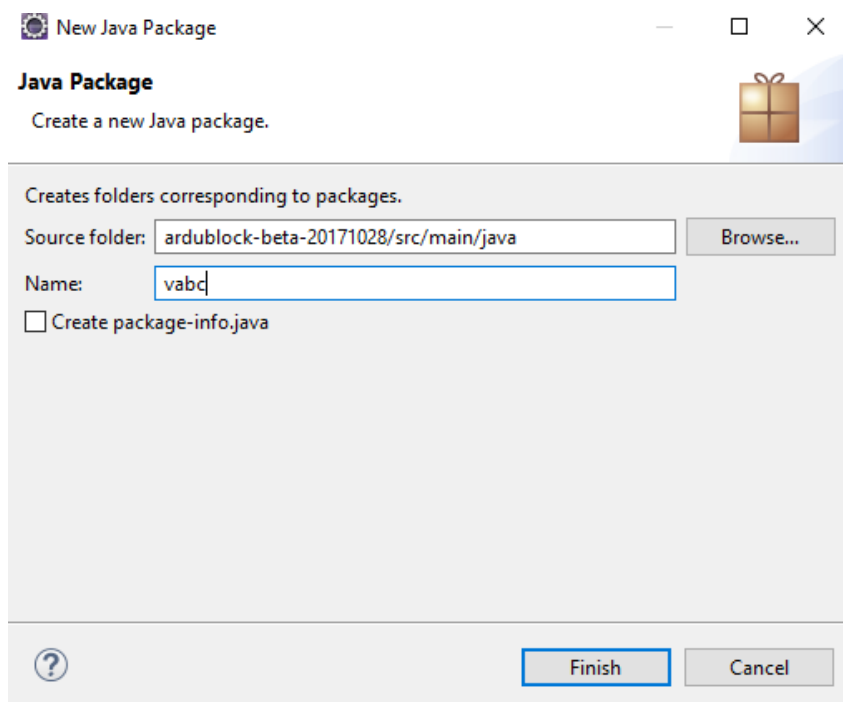
- 1) Selecionar a pasta “src/main/java”;



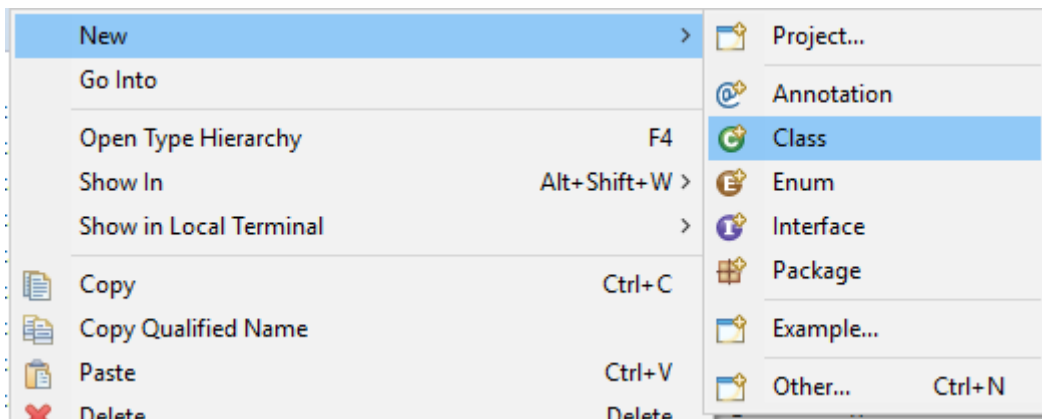
- 2) Clicar com o botão direito do rato. *New -> Package*;



- 3) Dar um nome no campo “Name” e clicar no botão “Finish”;



- 4) Agora que tem um *package* criado clique sobre ele com o botão direito do rato e selecione *New -> Class*;



- 5) Coloque um nome no campo “Name” e em seguida no botão “Finish”;

New Java Class

Java Class
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

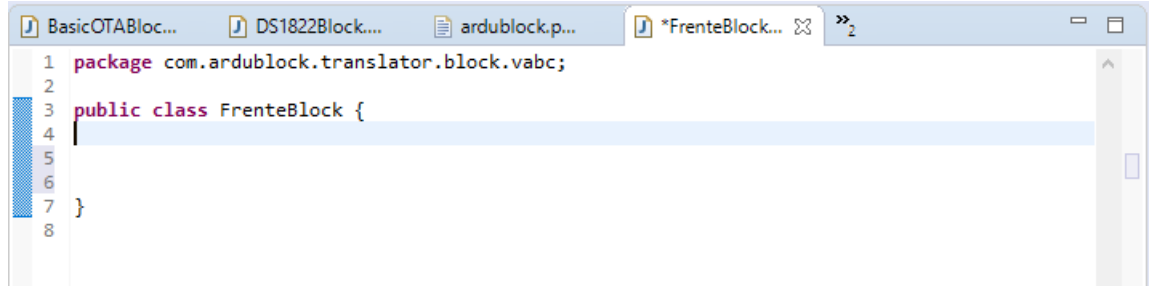
Superclass:

Interfaces:

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

6) Se abrir a classe vai ter algo como é demonstrado na figura seguinte;



```
1 package com.ardublock.translator.block.vabc;
2
3 public class FrenteBlock {
4
5
6
7 }
8
```

7) Para criar um bloco poderá ser utilizado o código que se encontra a seguir na caixa de texto. Sendo que as *Strings* introduzidas no “*translator.addHeaderFile*”, “*translator.addDefinitionCommand*”, “*translator.addSetupCommand*”, e “*return*” será o código final como mostra a imagem a seguir á caixa de texto;

```
package com.ardublock.translator.block.vabc;

import com.ardublock.translator.Translator;
import com.ardublock.translator.block.TranslatorBlock;
import com.ardublock.translator.block.exception.SocketNullException;
import com.ardublock.translator.block.exception.SubroutineNotDeclaredException;

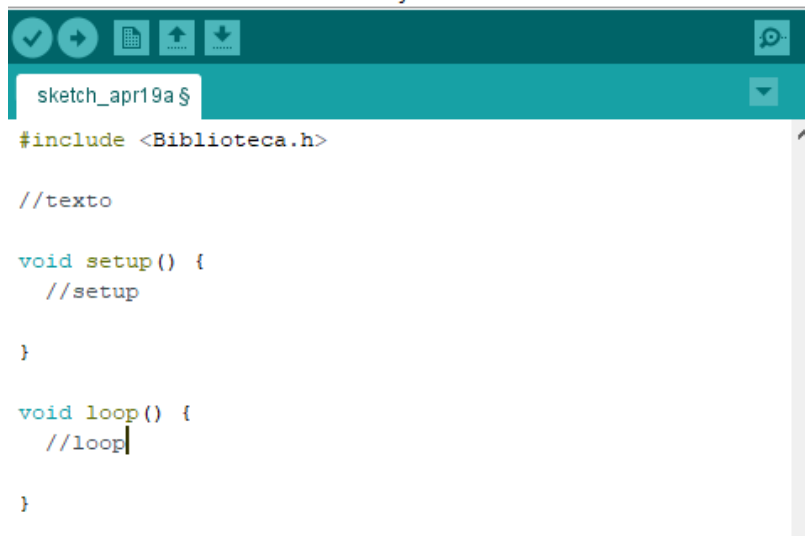
public class FrenteBlock extends TranslatorBlock{

    public FrenteBlock(Long blockId, Translator translator, String
codePrefix, String codeSuffix, String label){
        super(blockId, translator, codePrefix, codeSuffix, label);
    }

    public String toCode() throws SocketNullException,
SubroutineNotDeclaredException{
        translator.addHeaderFile("Biblioteca.h");
        translator.addDefinitionCommand("//texto");
        translator.addSetupCommand("//setup");
        return "//loop";
    }

}
```

Ficheiro Editar Rascunho Ferramentas Ajuda



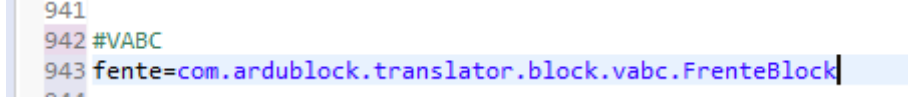
```
sketch_apr19a$
#include <Biblioteca.h>

//texto

void setup() {
    //setup
}

void loop() {
    //loop
}
```

- 8) Posto isto, abra "src/main/resources -> com.ardublock.block";
- 9) Abra o ficheiro "ardublock.xml", "ardublock.properties" e "block-mapping.properties";
- 10) No ficheiro "block-mapping.properties", faça *scroll* até ao fim e digite uma variável igual a uma diretoria do ficheiro, sendo neste caso:
"com.ardublock.translator.block.vabc.FrenteBlock", ficando com o resultado da imagem seguinte:



- 11) No ficheiro "ardublock.xml", faça *scroll* até quase ao fim e antes do ultimo "</BlockGenus>" crie uma nova estrutura como é mostrado na seguinte imagem:

```
</BlockGenus>

<!-- VABC -->
<BlockGenus name="frente" kind="command" color="255 0 255"
  initlabel="bg.frente_command">
  <description>
    <text>Frente Command</text>
  </description>
  <BlockConnectors>
  </BlockConnectors>
</BlockGenus>
```

- 12) Vá ao ficheiro "ardublock.properties", faça *scroll* até ao fim e adicione as seguintes linhas de código:

```
bd.vabc=VABC
bg.frente=Frente
bg.frente.description=frente
bg.frente_command=Frente
```

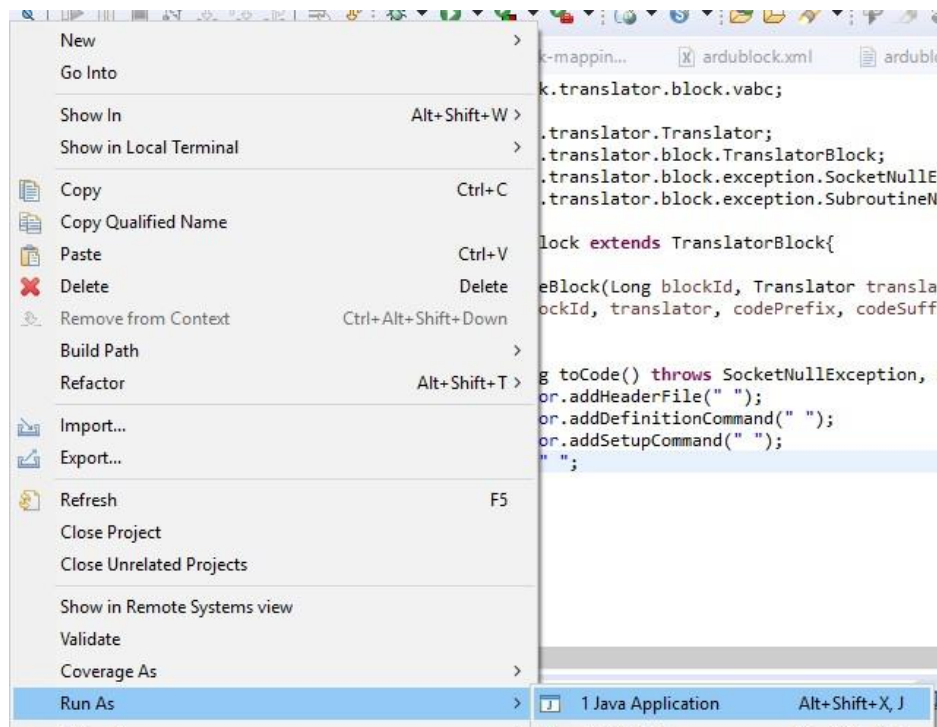
- 13) Volte ao "ardublock.xml", faça *scroll* até encontrar "<BlockDrawerSets>" e adicione no fim as seguintes linhas de código:

P.S.: Ignore as linhas "<BlockGenusMember>basicota</BlockGenusMember>" e "<BlockGenusMember>ds1822</BlockGenusMember>";

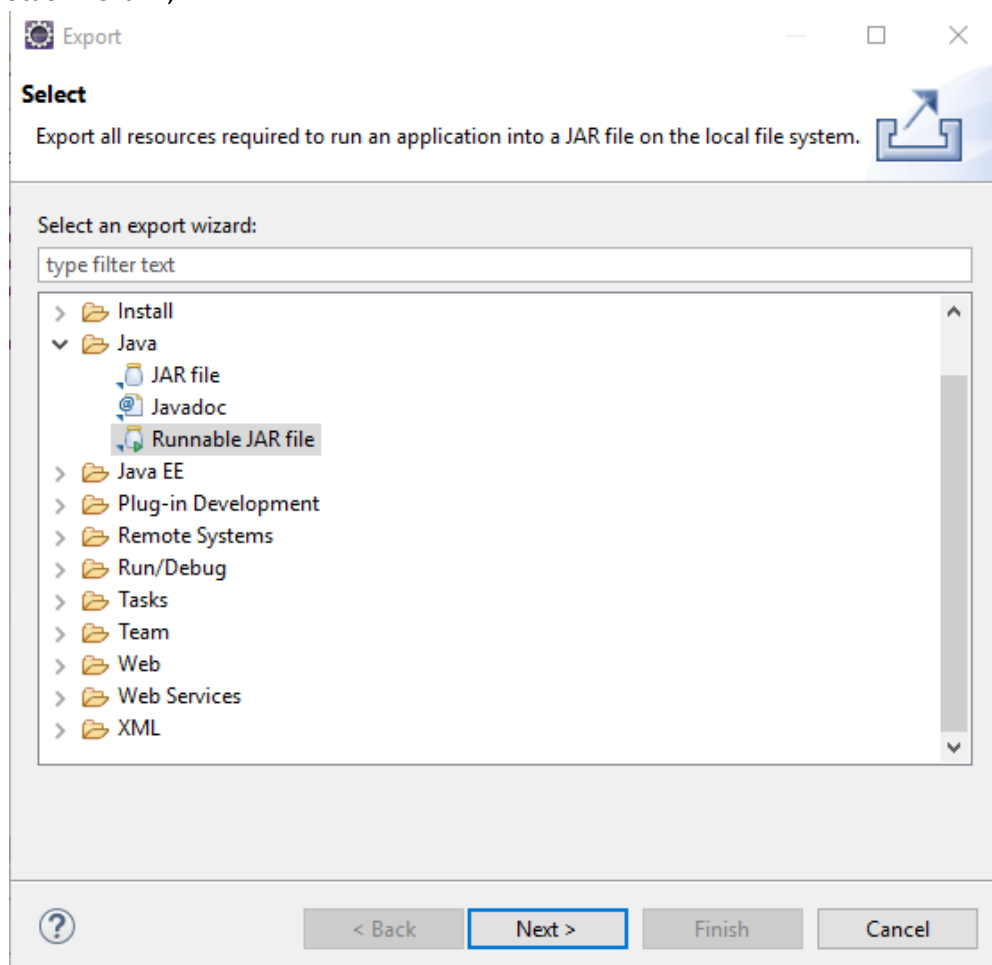
```
<BlockDrawer button-color="255 0 255" name="bd.vabc">
  <BlockGenusMember>frente</BlockGenusMember>
  <BlockGenusMember>basicota</BlockGenusMember>
  <BlockGenusMember>ds1822</BlockGenusMember>
</BlockDrawer>

</BlockDrawerSet>
</BlockDrawerSets>
```

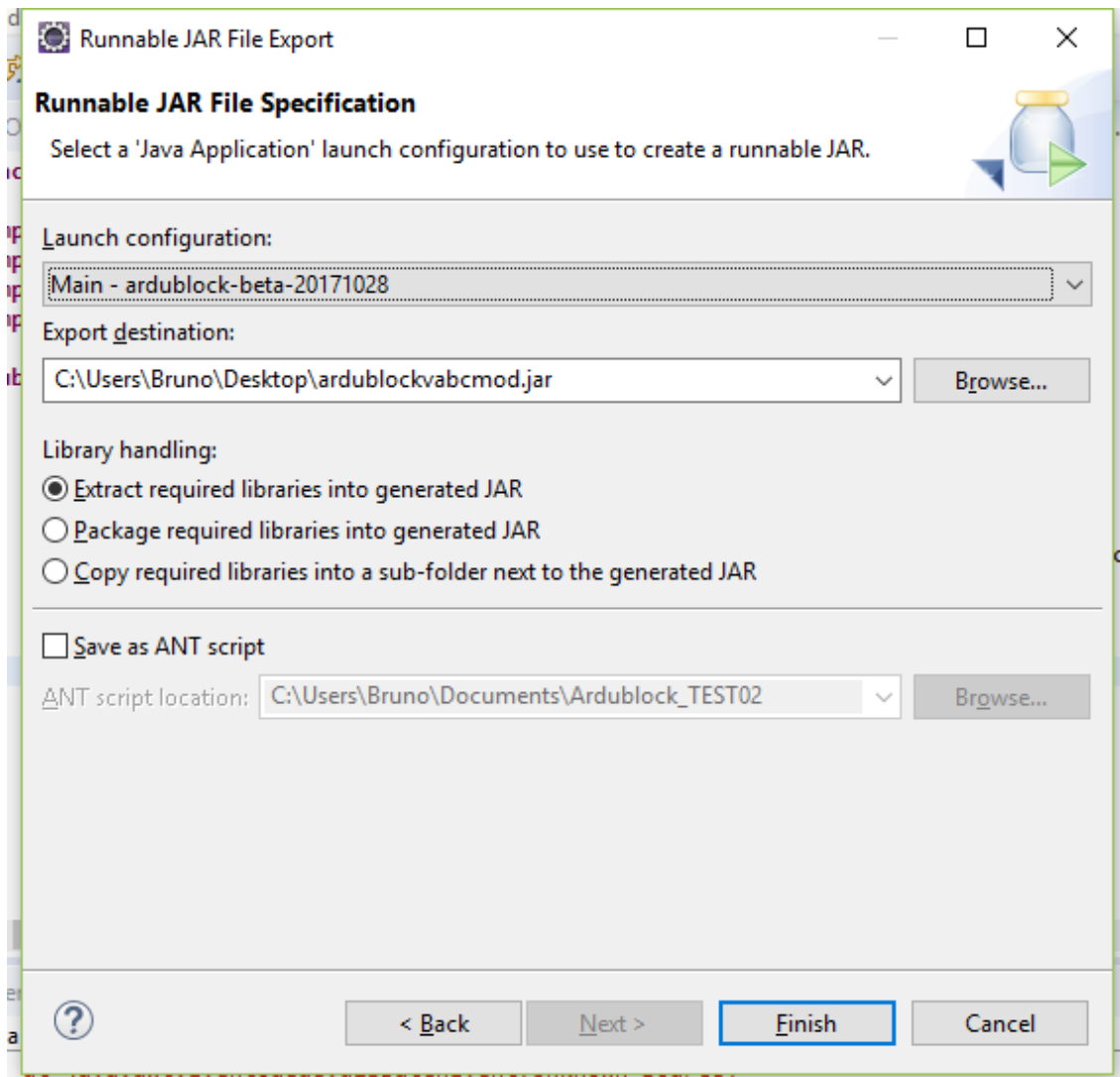
- 14) Agora que finalizou a criação de uma aba e de um bloco teste a aplicação. Poderá fazê-lo clicando com o botão direito na pasta principal em seguida "Run As -> Java Application"



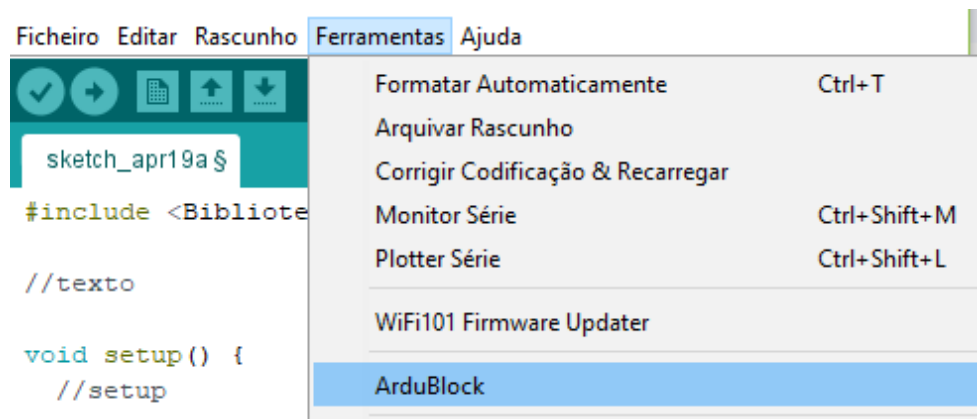
- 15) Se tudo estiver correto poderá finalmente exportar a aplicação. Clique com o botão direito na pasta principal em seguida “Export...”;
- 16) Vai abrir uma janela. Nela terá de clicar em “Java -> Runnable JAR file”, em seguida no botão “Next >”;



- 17) Selecione o destino para onde pretende o ficheiro “.jar” e configure a janela como é demonstrado na imagem. Clique no botão “Finish”;



- 18) Agora que já tem o ficheiro “.jar” é só coloca-lo na devida pasta do Arduino, substituindo o ficheiro antigo. No nosso caso será o seguinte destino “C:\Users\Bruno\Documents\Arduino\tools\ArduBlockTool\tool”;
- 19) Agora ao clicar em “Ferramentas -> Ardublock” é suposto já aparecer os novos blocos criados.



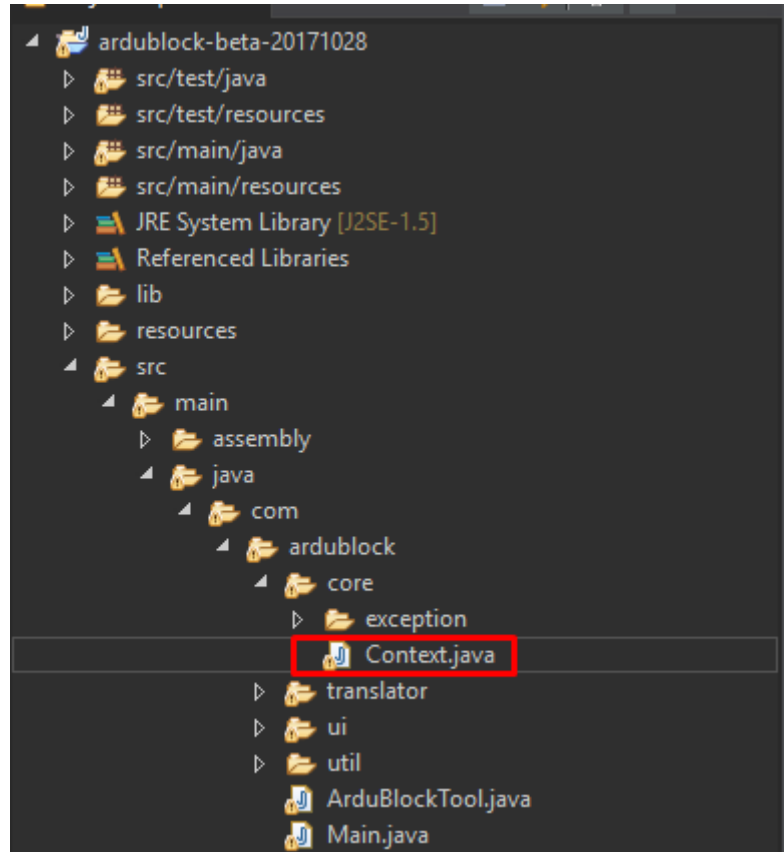
Tutorial para mudar bloco inicial no *Ardublock* pelo eclipse

Veículo Aquático de Baixo Custo

Bruno Silva, 2150634

Filipe Salgueiro, 2151226

- 1) Aceder ao ficheiro na imagem seguinte;



- 2) Alterar o "loop" pelo bloco pretendido, neste caso presente na linha 181

```
175  
176 Workspace workspace = workspaceController.getWorkspace();  
177 Page page = workspace.getPageNamed("Main");  
178  
179 FactoryManager manager = workspace.getFactoryManager();  
180 Block newBlock;  
181 newBlock = new Block(workspace, "loop", false);  
182 FactoryRenderableBlock factoryRenderableBlock = new FactoryRenderableBlock(workspace, manager);  
183 RenderableBlock renderableBlock = factoryRenderableBlock.createNewInstance();  
184 renderableBlock.setLocation(100, 100);  
185 page.addBlock(renderableBlock);  
186
```