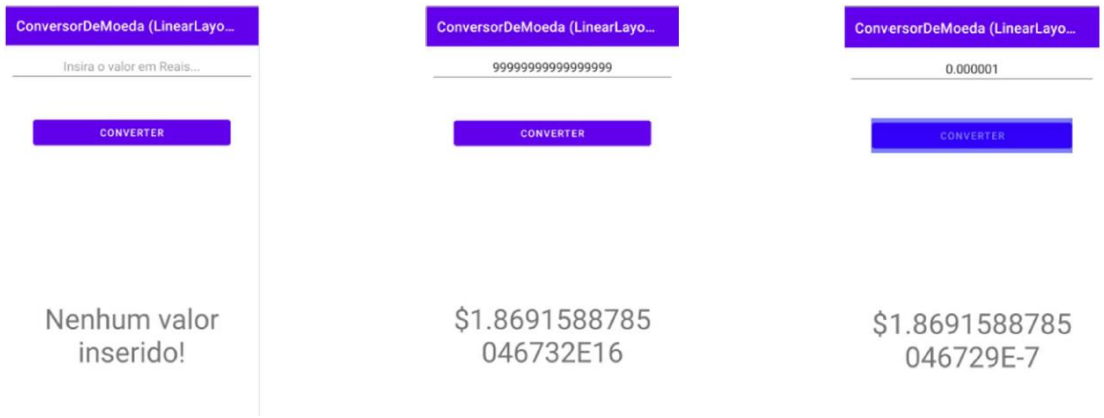
FUNDAÇÃO UNIVERSIDADE FEDERAL DE RONDÔNIA
NÚCLEO DE TECNOLOGIA
DEPARTAMENTO ACADÊMICO DE CIÊNCIA DA COMPUTAÇÃO
PALOMAKOBA – LABORATÓRIO – PROF. PABLO VARGAS
ATIVIDADE AVALIATIVA
03/12/2021

Discente: Carolina Yukari Veludo Watanabe

1) *Faça um teste automatizado no App Conversor de Moeda*



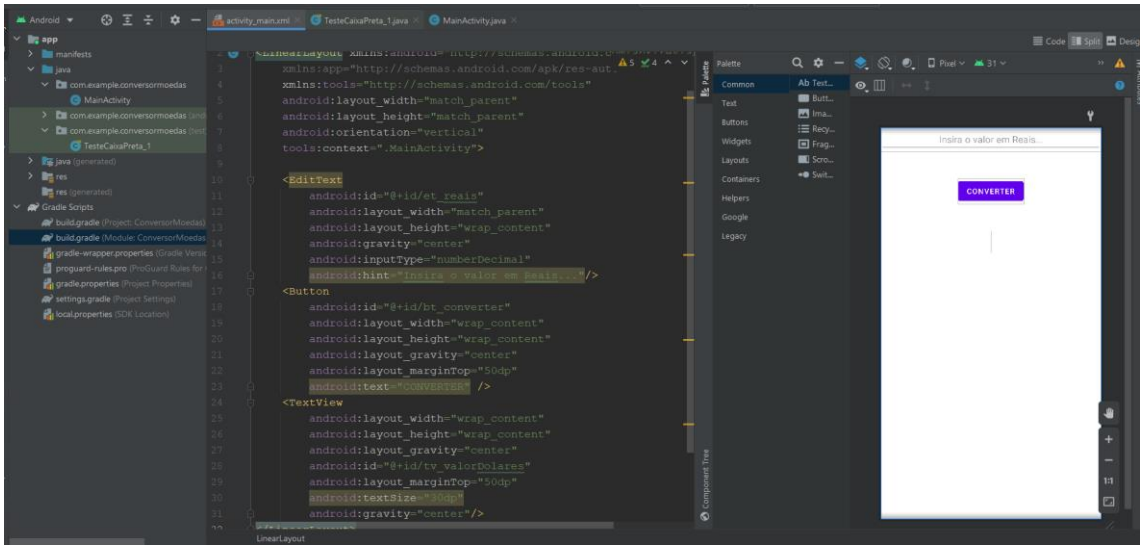
Resposta:

A gravação da realização desta atividade está no link: https://drive.google.com/file/d/1kteiXBnTdHb13o1pIbVWWIls0hh2csD7/view?usp=sharing

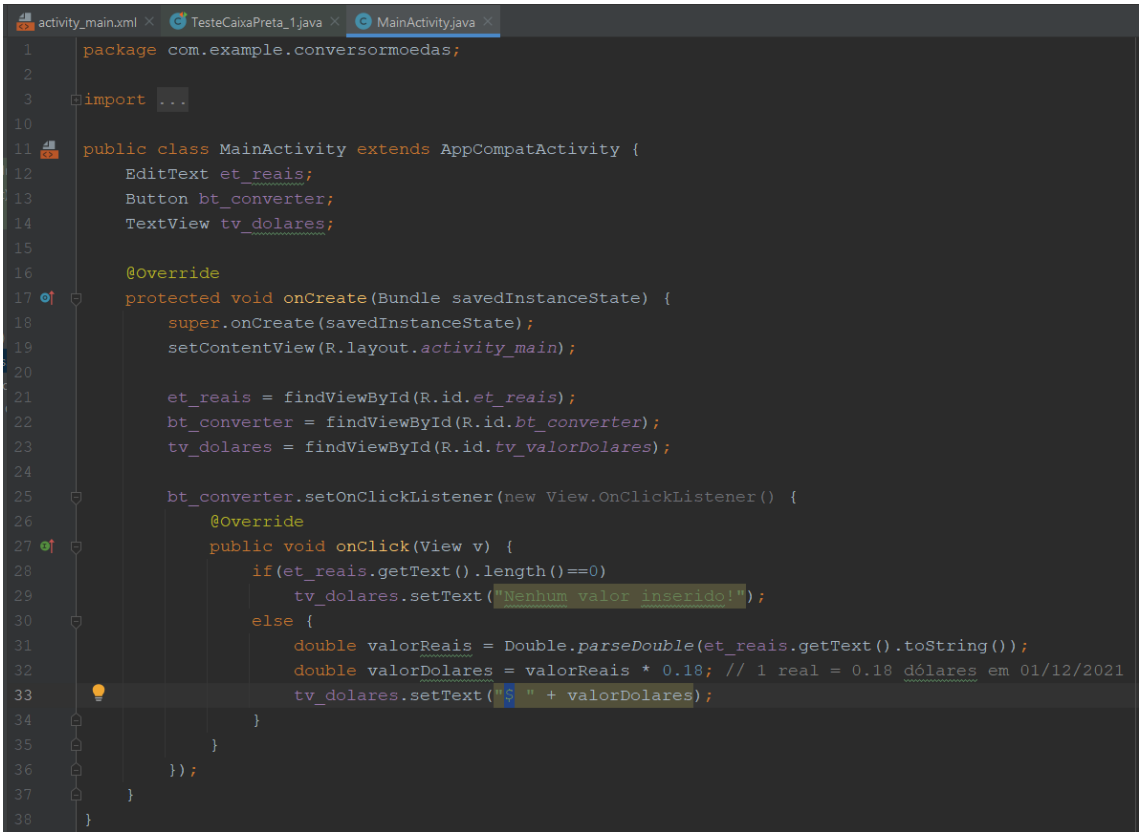
Todo o código está disponível no arquivo ConversorMoedas_ex1.rar.

A seguir estão apresentados os *prints* da tela da resolução do problema.

activity_main.xml



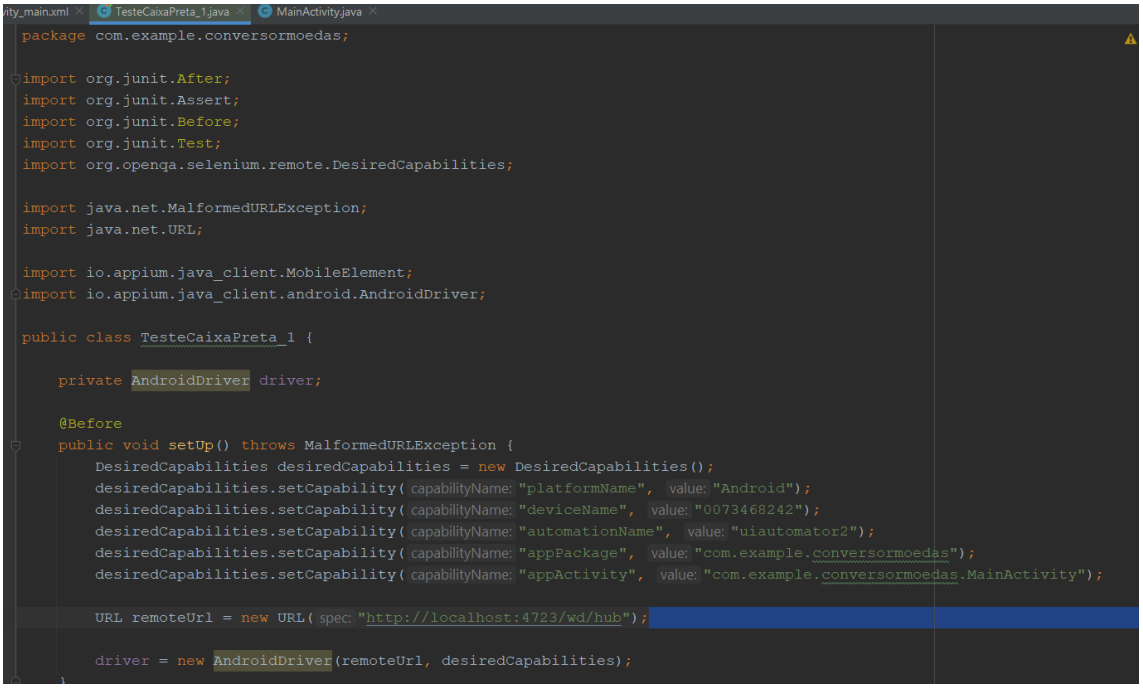
MainActivity.java



```
package com.example.conversormoedas;

import ...

public class MainActivity extends AppCompatActivity {
    EditText et_reais;
    Button bt_converter;
    TextView tv_dolares;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        et_reais = findViewById(R.id.et_reais);
        bt_converter = findViewById(R.id.bt_converter);
        tv_dolares = findViewById(R.id.tv_valorDolares);

        bt_converter.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(et_reais.getText().length()==0)
                    tv_dolares.setText("Nenhum valor inserido!");
                else {
                    double valorReais = Double.parseDouble(et_reais.getText().toString());
                    double valorDolares = valorReais * 0.18; // 1 real = 0.18 dólares em 01/12/2021
                    tv_dolares.setText("$ " + valorDolares);
                }
            }
        });
    }
}
```

TesteCaixaPreta_1.java

Início

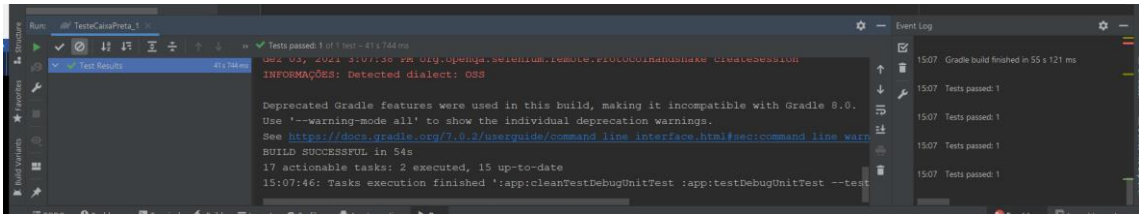


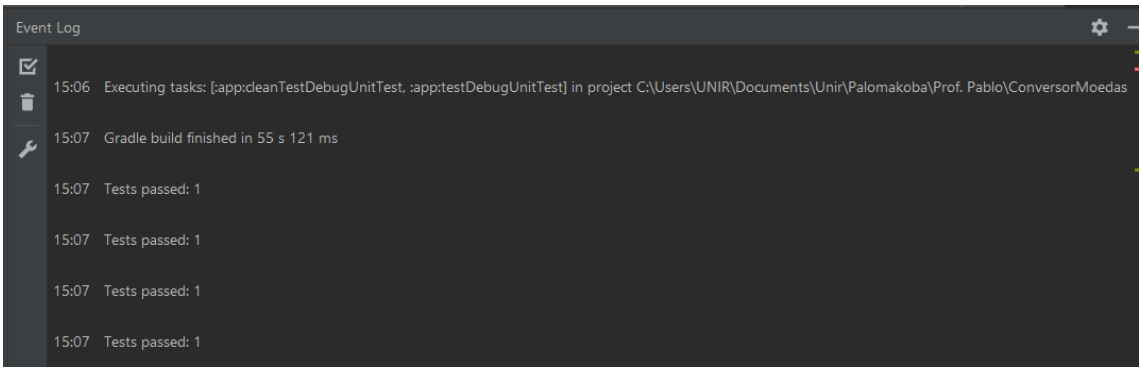
```
package com.example.conversormoedas;

import org.junit.After;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.remote.DesiredCapabilities;

import java.net.MalformedURLException;
import java.net.URL;

import io.appium.java_client.MobileElement;
import io.appium.java_client.android.AndroidDriver;

public class TesteCaixaPreta_1 {

    private AndroidDriver driver;

    @Before
    public void setUp() throws MalformedURLException {
        DesiredCapabilities desiredCapabilities = new DesiredCapabilities();
        desiredCapabilities.setCapability( capabilityName: "platformName", value: "Android");
        desiredCapabilities.setCapability( capabilityName: "deviceName", value: "0073468242");
        desiredCapabilities.setCapability( capabilityName: "automationName", value: "uiautomator2");
        desiredCapabilities.setCapability( capabilityName: "appPackage", value: "com.example.conversormoedas");
        desiredCapabilities.setCapability( capabilityName: "appActivity", value: "com.example.conversormoedas.MainActivity");

        URL remoteUrl = new URL( spec: "http://localhost:4723/wd/hub");

        driver = new AndroidDriver(remoteUrl, desiredCapabilities);
    }
```

Testes

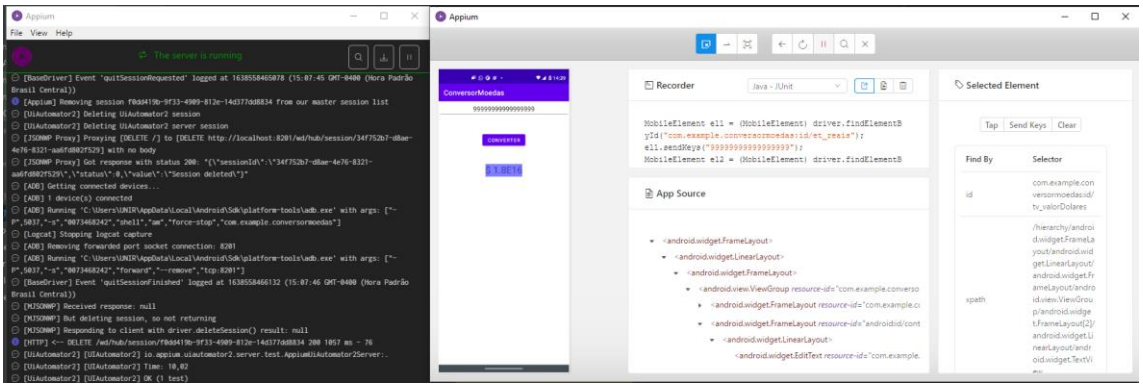
```java
public void testarConversorMoedas() {
    //Teste 1
    MobileElement el1 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
    el1.click();
    MobileElement el2 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
    el2.click();
    Assert.assertEquals( expected: "Nenhum valor inserido!", el2.getText());
    //Teste 2
    MobileElement el3 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/et_reais");
    el3.sendKeys( ...keysToSend: "99999999999999999");
    MobileElement el4 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
    el4.click();
    MobileElement el5 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
    el5.click();
    Assert.assertEquals( expected: "$ 1.8E16", el5.getText());
    //Teste 3
    MobileElement el6 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/et_reais");
    el6.sendKeys( ...keysToSend: "0.000001");
    MobileElement el7 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
    el7.click();
    MobileElement el8 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
    el8.click();
    Assert.assertEquals( expected: "$ 1.8E-7", el8.getText());
}

@After
public void tearDown() {
    driver.quit();
}
}
```

Teste passou





Exemplo do funcionamento Appium



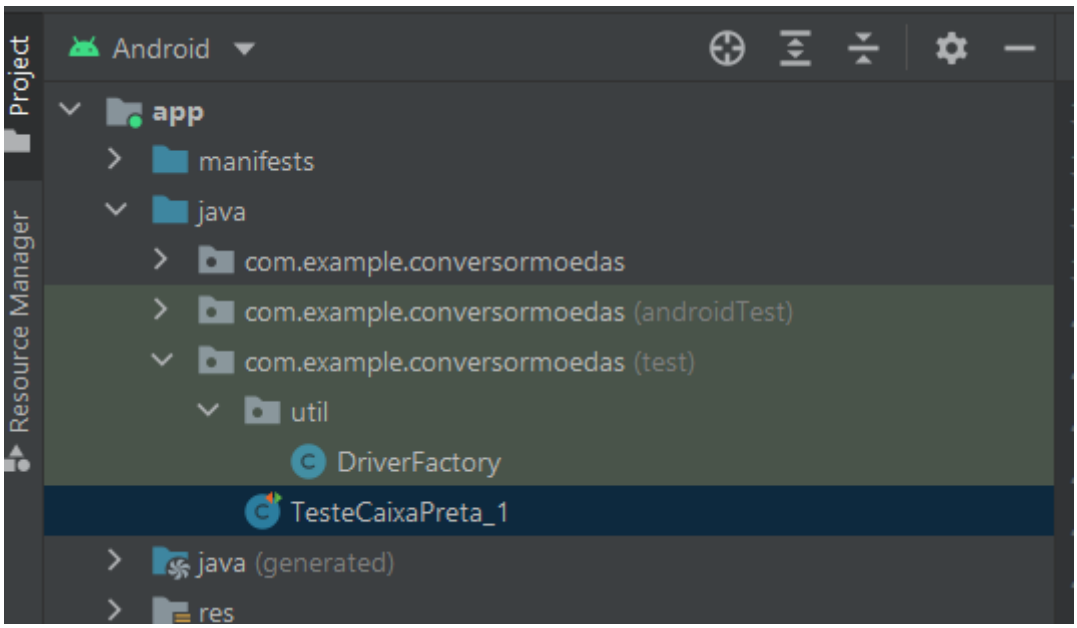
2) *Crie um DriverFactory no App Conversor de Moeda utilizado no Exercício 1.*

Resposta

Os códigos deste exercício estão no arquivo ConversorMoedas_ex2.rar.

A gravação do teste com explicação pode ser visto em: https://drive.google.com/file/d/1ezIlXW1QkWO4YRW9DHmF9Zw0Tk29_G8-/view?usp=sharing

Inseri o package útil e criei a classe DriverFactory



Implementação da classe DriverFactory.java, inseri as Capabilities e criei um try/catch para tratar a exceção da URL.

```
activity_main.xml | TesteCaixaPreta_1.java | DriverFactory.java | MainActivity.java
package com.example.conversormoedas.util;

import ...

public class DriverFactory {
    private static AndroidDriver<MobileElement> driver;
    public static AndroidDriver<MobileElement> getDriver(){
        if (driver == null) {
            criarDriver();
        }
        return driver;
    }

    private static void criarDriver() {
        DesiredCapabilities desiredCapabilities = new DesiredCapabilities();
        desiredCapabilities.setCapability( capabilityName: "platformName", value: "Android");
        desiredCapabilities.setCapability( capabilityName: "deviceName", value: "0073468242");
        desiredCapabilities.setCapability( capabilityName: "automationName", value: "uiautomator2");
        desiredCapabilities.setCapability( capabilityName: "appPackage", value: "com.example.conversormoedas");
        desiredCapabilities.setCapability( capabilityName: "appActivity", value: "com.example.conversormoedas.MainActivity");

        URL remoteUrl = null;
        try {
            remoteUrl = new URL( spec: "http://localhost:4723/wd/hub");
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }

        driver = new AndroidDriver(remoteUrl, desiredCapabilities);
    }
```

Implementação do método para finalizar

```
    public static void finalizarDriver(){
        if(driver != null){
            driver.quit();
            driver = null;
        }
    }
}
```

No TesteCaixaPreta_1.java, atualizei o método setUp(), já que as capabilities agora estão na classe DriverFactory:

```
public class TesteCaixaPreta_1 {

    private AndroidDriver driver;

    @Before
    public void setUp(){
        driver = DriverFactory.getDriver();

    }
```

O teste ficou do mesmo jeito que o anterior

```
    @Test
    public void testarConversorMoedas() {
        //Teste 1
        MobileElement el1 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
        el1.click();
        MobileElement el2 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
        el2.click();
        Assert.assertEquals( expected: "Nenhum valor inserido!", el2.getText());
        //Teste 2
        MobileElement el3 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/et_reais");
        el3.sendKeys( ...keysToSend: "99999999999999999");
        MobileElement el4 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
        el4.click();
        MobileElement el5 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
        el5.click();
        Assert.assertEquals( expected: "$ 1.8E16", el5.getText());
        //Teste 3
        MobileElement el6 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/et_reais");
        el6.sendKeys( ...keysToSend: "0.000001");
        MobileElement el7 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/bt_converter");
        el7.click();
        MobileElement el8 = (MobileElement) driver.findElementById("com.example.conversormoedas:id/tv_valorDolares");
        el8.click();
        Assert.assertEquals( expected: "$ 1.8E-7", el8.getText());
    }
```

e o método tearDown também foi atualizada, para chamar o método finalizarDriver() da classe DriverFactory:

```
    @After
    public void tearDown() {
        DriverFactory.finalizarDriver();
    }
```

Após rodar o teste, o mesmo passou: