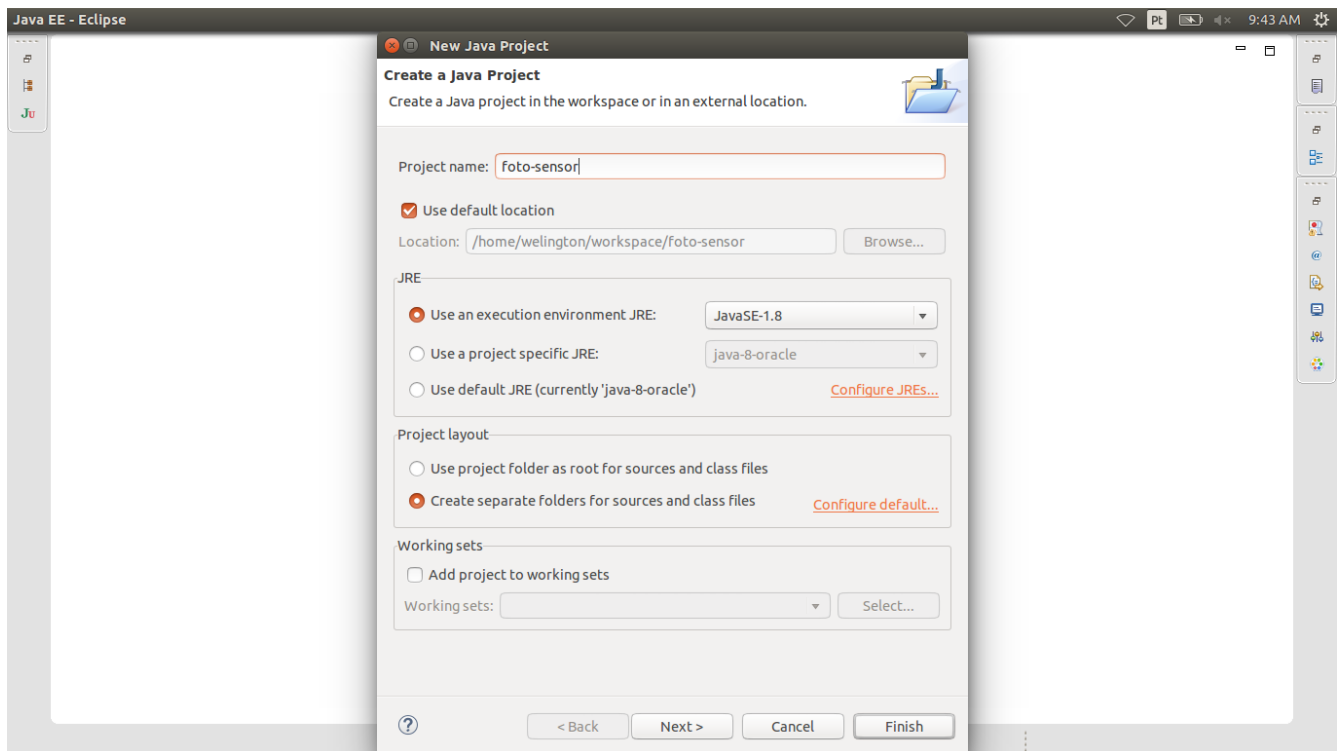


Tutorial do projeto foto-sensor

OBS. : Para rodar o projeto você precisa de um notebook ou um computador com **webcam**. Embora este tutorial seja criado utilizando a IDE Eclipse ele pode rodar também em outras IDEs.

1. Criando um novo projeto no eclipse

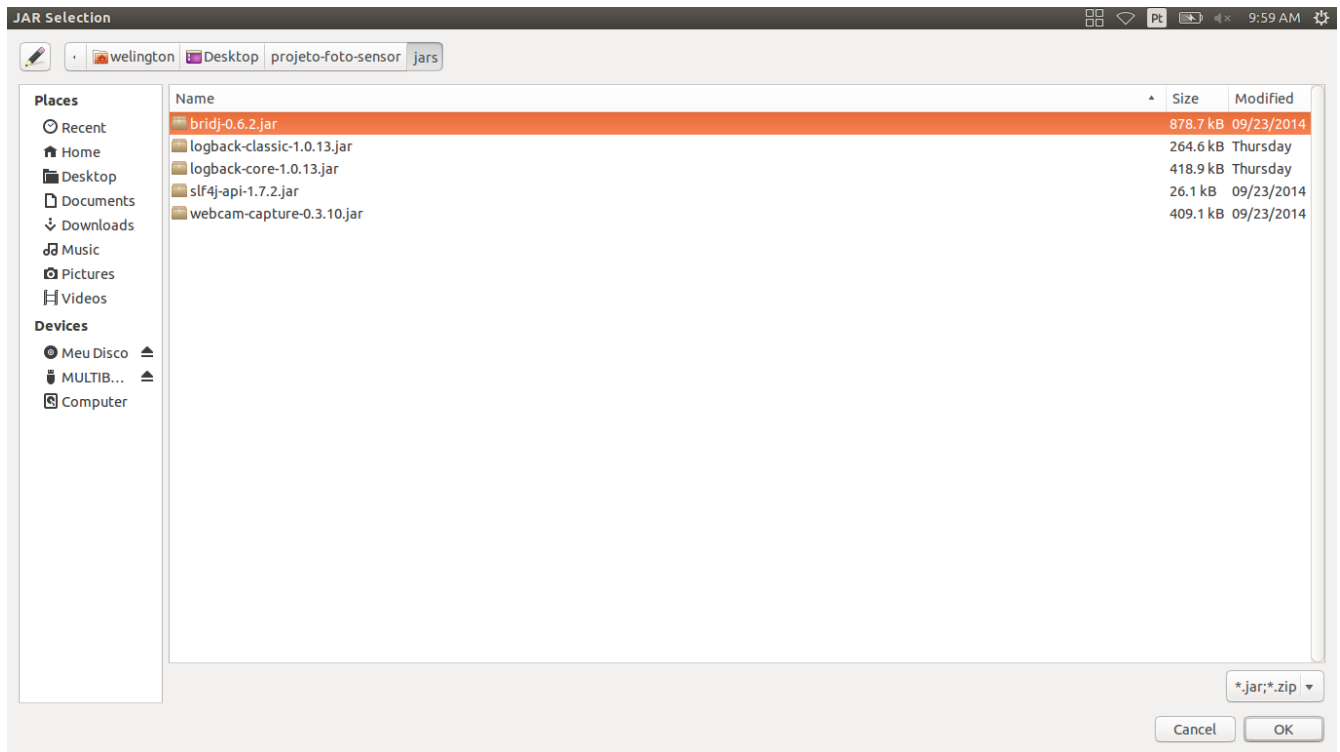
Va em **File** → **New** → **Java Project**. Crie um projeto chamado foto-sensor.



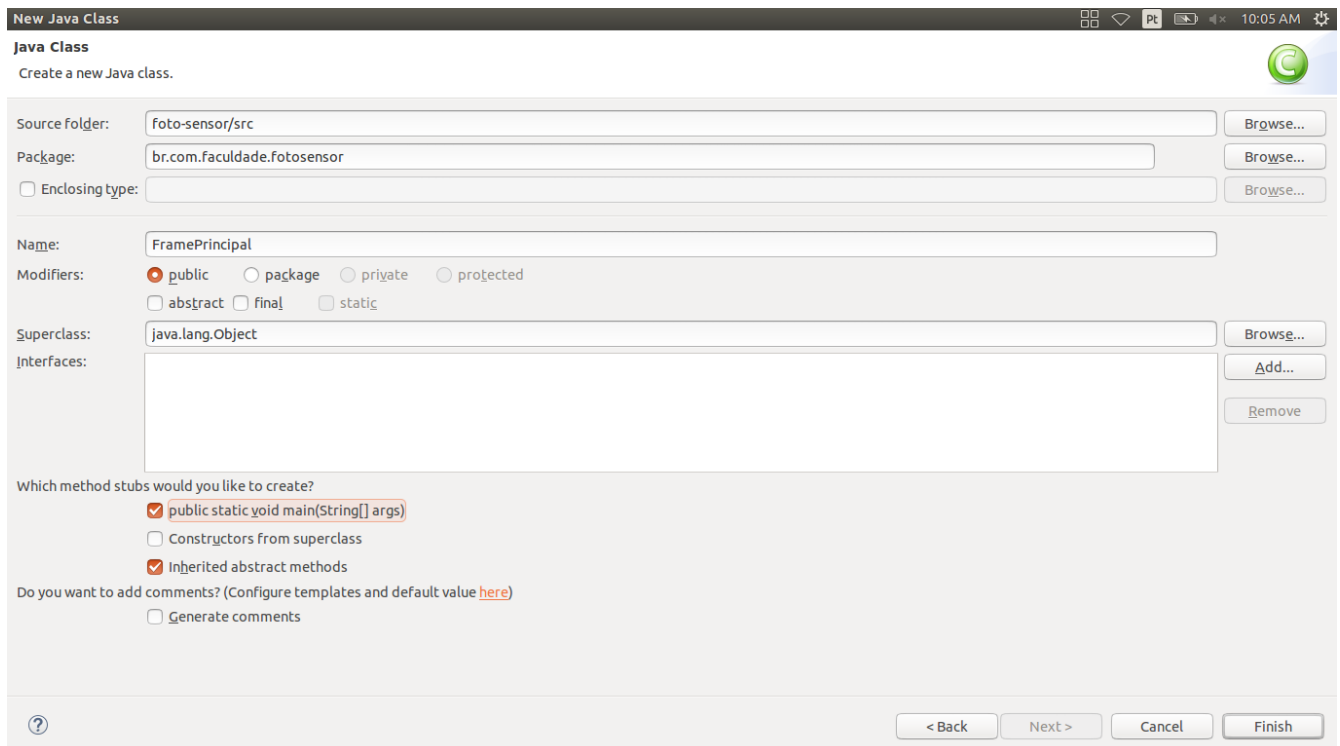
Agora precisamos colocar os JARs em nosso projeto, vá a pasta jars, você deverá encontrar os seguintes jars:

- webcam-capture-0.3.10.jar;
- slf4j-api-1.7.2.jar;
- bridj-0.6.2.jar;
- logback-classic-1.0.13.jar;
- logback-core-1.0.13.jar.

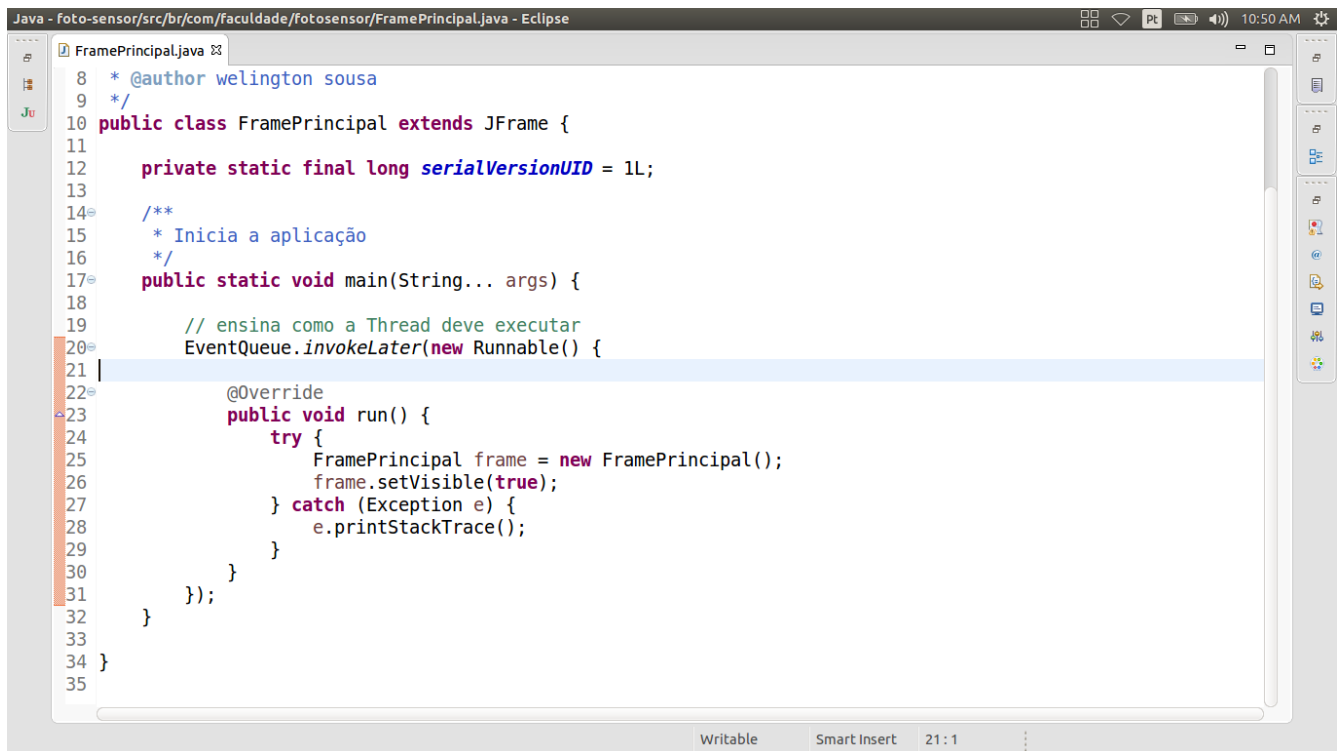
Para adicionar os jars clique no projeto vá **Build Path** → **Configure Build Path** → **Libraries**. Clique no botão **Add External JARs...**



Crie uma nova classe java chamada de FramePrincipal e marque a opção **public static void main** (String[] args).

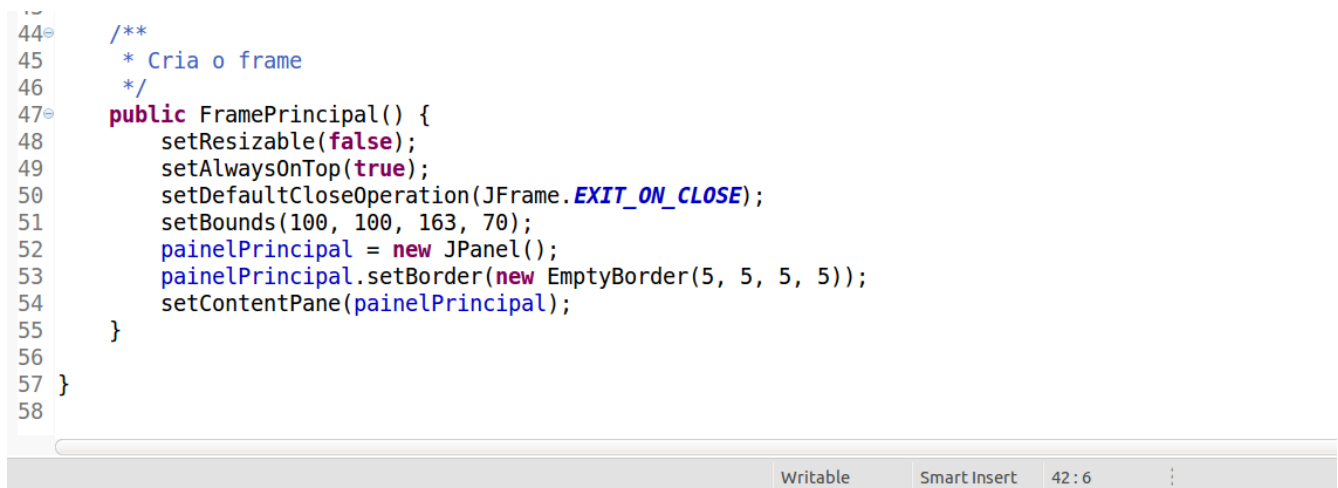


Extenda FramePrincipal para JFrame e no método, Main precisamos mostrar quando e como a Thread deve executar.



```
Java - foto-sensor/src/br/com/faculdade/fotosensor/FramePrincipal.java - Eclipse
FramePrincipal.java
8  * @author welington sousa
9  */
10 public class FramePrincipal extends JFrame {
11
12     private static final long serialVersionUID = 1L;
13
14     /**
15      * Inicia a aplicação
16      */
17     public static void main(String... args) {
18
19         // ensina como a Thread deve executar
20         EventQueue.invokeLater(new Runnable() {
21
22             @Override
23             public void run() {
24                 try {
25                     FramePrincipal frame = new FramePrincipal();
26                     frame.setVisible(true);
27                 } catch (Exception e) {
28                     e.printStackTrace();
29                 }
30             }
31         });
32     }
33
34 }
35
Writable Smart Insert 21:1
```

Inicie a criação do Frame no construtor



```
44     /**
45      * Cria o frame
46      */
47     public FramePrincipal() {
48         setResizable(false);
49         setAlwaysOnTop(true);
50         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
51         setBounds(100, 100, 163, 70);
52         painelPrincipal = new JPanel();
53         painelPrincipal.setBorder(new EmptyBorder(5, 5, 5, 5));
54         setContentPane(painelPrincipal);
55     }
56
57 }
58
Writable Smart Insert 42:6
```

Ok, agora crie um método privado para iniciar a WebCam da seguinte forma e insira-o no construtor, após setContentPane()

```

*/
public FramePrincipal() {
    setResizable(false);
    setAlwaysOnTop(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 163, 70);
    painelPrincipal = new JPanel();
    painelPrincipal.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(painelPrincipal);

    iniciaWebCam();
}

/**
 * Inicia a webcam
 */
private void iniciaWebCam() {
    JFrame janela = new JFrame("Camera");
    janela.getContentPane().add(panel);
    janela.setResizable(false);
    janela.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    janela.pack();
    janela.setSize(540, 540);
    janela.setVisible(true);
}

```

Precisamos tambem criar um layout e agrupa-lo e para isso vamos usar um **GroupLayout**

```

// agrupa nosso layout
GroupLayout contentLayout = new GroupLayout(painelPrincipal);
contentLayout
    .setHorizontalGroup(contentLayout.createParallelGroup(Alignment.LEADING)
        .addGroup(Alignment.TRAILING, contentLayout.createSequentialGroup()
            .addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(botaoCapturar, GroupLayout.PREFERRED_SIZE,
                135, GroupLayout.PREFERRED_SIZE)
            .addContainerGap());
contentLayout.setVerticalGroup(
    contentLayout.createParallelGroup(Alignment.LEADING).addGroup(contentLayout
        .createSequentialGroup()
        .addComponent(botaoCapturar).addContainerGap(GroupLayout
            .DEFAULT_SIZE, Short.MAX_VALUE)));
painelPrincipal.setLayout(contentLayout);

```

Vamos precisar de um botão para capturar a imagem do frame

```

JButton botaoCapturar = new JButton("Capturar");
botaoCapturar.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
        // trata o evento
        try {
            capturaImagem();
            gravaTextoImagem();
        } catch (IOException e1) {
            e1.printStackTrace();
        } catch (ParseException e1) {
            e1.printStackTrace();
        }
    }
});

```

Precisamos agora capturar a imagem criando alguma lógica

```

// captura a imagem
private void capturaImagem() throws IOException {
    BufferedImage imagem = webcam.getImage();
    ImageIO.write(imagem, "PNG", new File("foto.png"));
}
};

```

Precisamos escrever na imagem os seguintes dados como Hora, Velocidade e a Rua respectivamente

```

// escreve na imagem
private void gravaTextoImagem() throws IOException, ParseException {

    // Hora atual
    LocalDateTime agora = LocalDateTime.now();
    String dataEHora = agora.format(DateTimeFormatter.ofPattern("hh:mm:ss"));

    // Velocidade atual
    double max = Math.random() * 100;
    long i = Math.round(max);
    String velocidade = String.valueOf(i);

    BufferedImage imagem = webcam.getImage();

    BufferedImage bi = new BufferedImage(imagem.getWidth(), 30, BufferedImage.TYPE_INT_RGB);

    Graphics2D g = bi.createGraphics();
    g.setColor(Color.BLACK);
    g.fillRect(0, 0, bi.getWidth(), 40);
    g.setColor(Color.WHITE);
    g.setFont(new Font("Arial", Font.BOLD, 12));
    g.drawString("Rua: " + "Avenida Mister Hull", bi.getWidth() - 625, 30);
    g.drawString("Velocidade: " + velocidade + " Km/h", bi.getWidth() - 625, 15);
    g.drawString("Hora: " + dataEHora, bi.getWidth() - 125, 20);
}

```

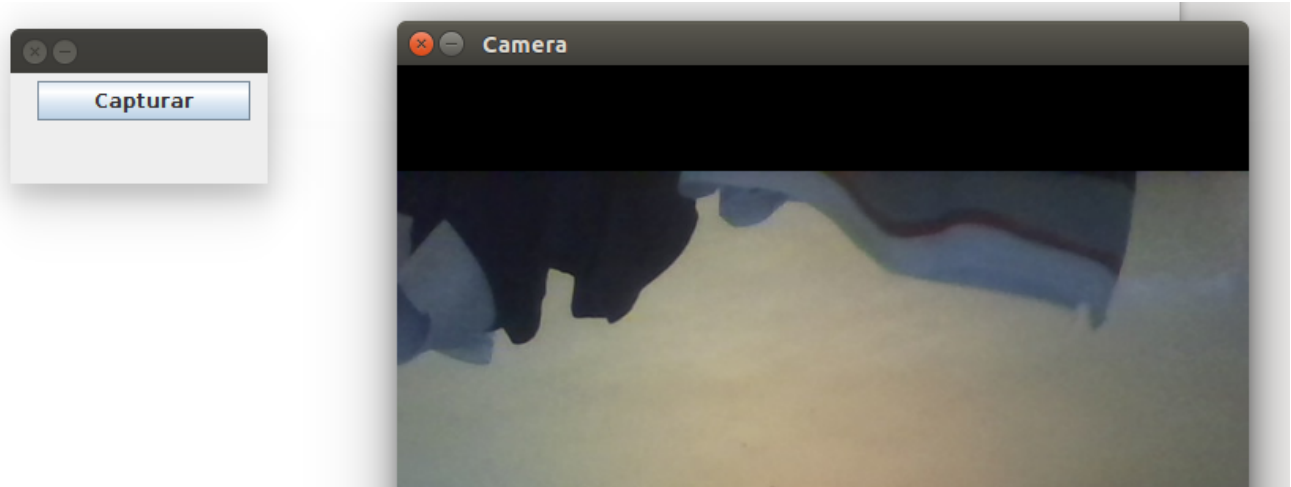
Agora devemos escrever o código responsável por inserir a tarja preta na imagem

```
// cria a tarja
@SuppressWarnings("unused")
public Graphics2D criaTarja(int x, String velocidade, String dataEHora) {
    BufferedImage bi = new BufferedImage(x, 30, BufferedImage.TYPE_INT_RGB);

    Graphics2D g = bi.createGraphics();
    g.fillRect(0, 0, bi.getWidth(), 40);
    return g;
}
```

Testando a aplicação

Para rodar o projeto vá **Run** → **Run As** → **Java Application**. É deverá aparecer o frame com o botão Capturar e o frame do Video a ser capturado.



Isso deve gerar uma imagem (foto.png) que ficará armazenada no projeto.