 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA MINAS GERAIS</p>	<p align="center">Ciência da Computação</p> <p align="center">Trabalho Prático 01</p> <p align="center">Criação de um protótipo de sistema de monitoramento da Amazônia.</p>	
Disciplina: Programação Orientada a Objetos	Data: ____/____/____	
Professor: Dr. Bruno Ferreira	Turma: Integral	
Aluno	Valor: 20,0	Nota:

Objetivo: Reforçar conceitos de programação orientada a objetos (classes, objetos, pacotes, encapsulamento, variáveis), introdução as interfaces gráficas.

Forma de Entrega: O código fonte e um diagrama de classe representando a implementação deverão ser enviados via e-mail para bruno.ferreira@ifmg.edu.br em um arquivo zipado **ou com um link do Dropbox/GoogleDrive**. A data limite para entrada é até o dia **16/10/2019**. A apresentação do trabalho será realizada em sala para o professor no dia **17/10/2019**.

OBS: Será observada e avaliada a legibilidade do código sendo fundamental uma boa indentação e a utilização de comentários.

O trabalho é em dupla, se for verificada cópia de trabalho os envolvidos receberão nota zero.

O governo federal requisitou que você crie um protótipo para leitura de dados de queimadas da Amazônia. Eles desejam que esse projeto seja composto de dois módulos (software), um chamado de “**análise**” e outro chamado de “**satélite**”. O módulo de análise armazena e cadastra todas as **informações** capturadas da Amazônia, das **regiões** da floresta (nome, se é área de proteção, esquadrão do exército responsável) e dos **esquadrões** do exército (nome, especialidade, quantidade de soldados). O módulo de análise é responsável também por enviar dados para o módulo de satélite (regiões, esquadrões), por requisitar informações (um código, imagem - matriz de pixels que representa a floresta, data da imagem e se ela já foi baixada pela central). Por fim, esse módulo deve apresentar informações como porcentagem de área em chamas de uma região em uma determinada data, e se as queimadas estão aumentando ao longo de um período nessa região.

O módulo satélite busca e armazena localmente os esquadrões e regiões recebidas do módulo de análise e também gera as informações (código, imagem, data, baixada?) quando requisitada. Essas imagens são geradas manualmente, pois não temos os equipamentos necessários. Assim, o usuário cria uma imagem com tons de vermelho (fogo) e verde (vegetação) - outras cores são ignoradas e, informa um código, uma data e uma região que se refere aquela informação.

Em relação a comunicação entre os dois módulos, por se tratar de um protótipo, os funcionários do governo disseram que pode ser conforme mostra a Figura 01. Ou seja, o módulo de análise envia dados para uma pasta no *google drive* e o satélite busca os dados nessa mesma pasta, a comunicação inversa é feita da mesma forma (Arquivos serão enviados e solicitados para uma pasta do *google drive*). Eles exigiram apenas que o formato dos arquivos devem ser via JSON (<https://pt.wikipedia.org/wiki/JSON>) – menos das imagens.

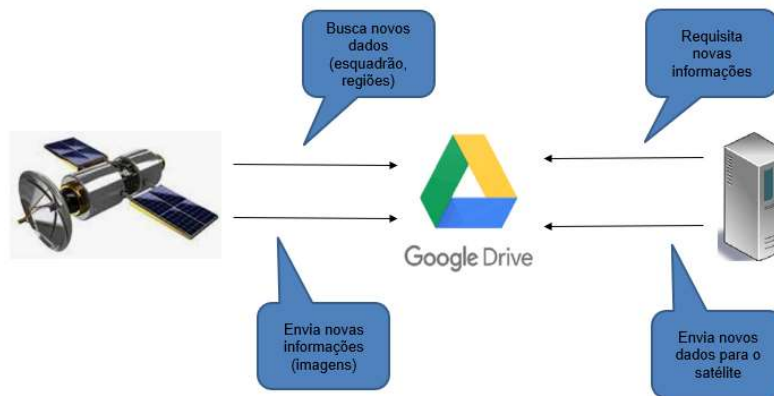


Figura 01 – Arquitetura de comunicação

Em relação as imagens da floresta, eles disseram que podemos usar imagens de entrada no formato **.ppm** (ASCII, P3) com representação de 8 bits por componente de cor (R, G, e B) de cada pixel (valores podem ir de 0 a 255). Mais detalhes sobre o formato **ppm** pode ser encontrado no endereço http://en.wikipedia.org/wiki/Netpbm_format

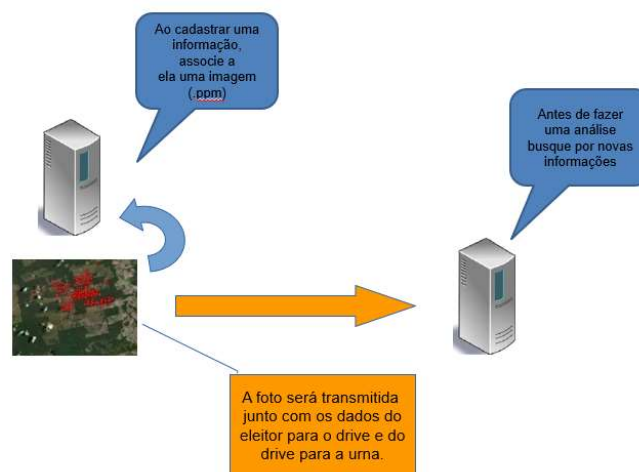


Figura 02 – Foto da região da Amazônia com extensão ppm

Alguns requisitos extras são:

- Use o pacote de componentes gráficos Swing para gerar a interface gráfica da aplicação;
- Trate ao máximo os possíveis erros do usuário ao utilizar sua interface gráfica;
- Valide os dados dos cadastros;
- Use "Java docs" para documentar todas as classes e seus métodos e o aidUML para gerar o diagrama de classe;
- Use pacotes dentro dos módulos Análise e Satélite.
- gere os .jar dos dois módulos.

Ponto extra?

- Criptografe as informações que são transmitidas.

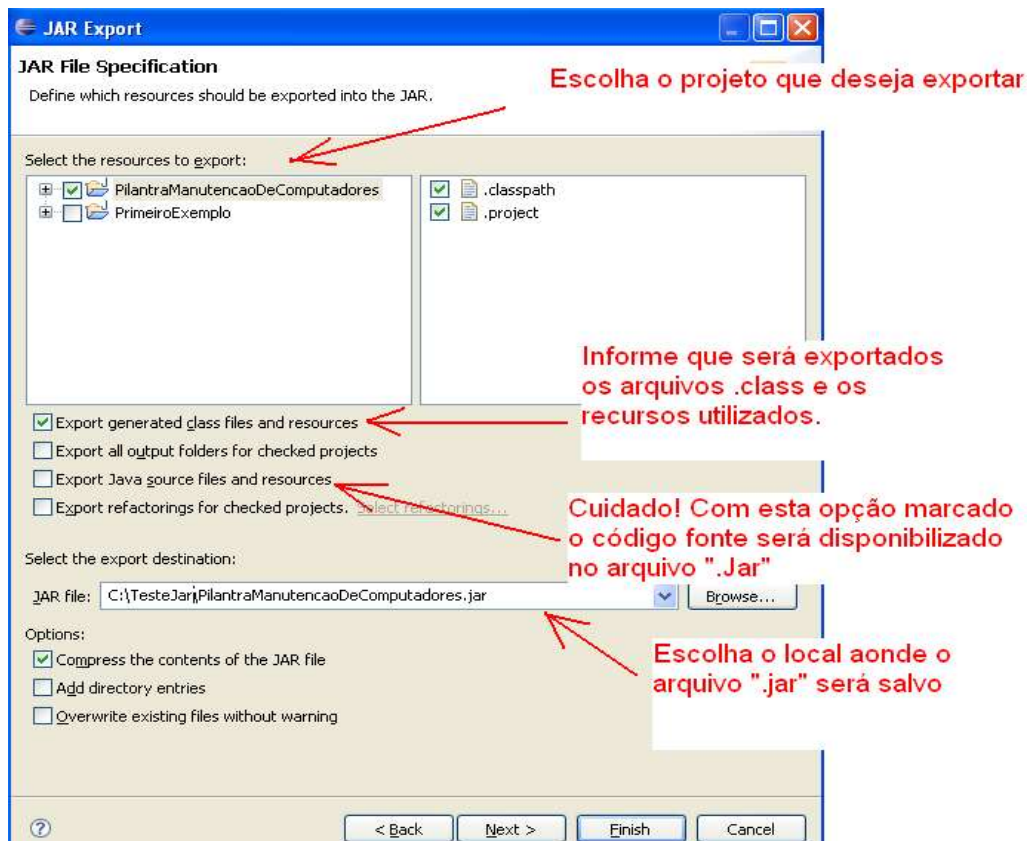
Último passo:

Gerar o executável: **(Atenção: lembre-se que o nome do seus dois projetos são: “Análise” e “Central”)**

- a) A primeira etapa é exportar o projeto. Para isso, clique com o botão direito do mouse sobre o projeto no “Package Explorer” e escolha a opção “Export...”.
- b) Com a janela “Export” aberta, escolha a opção “Jar Files” e clique no botão “Next”:



Na próxima tela será informado qual o projeto será exportado e qual o diretório que ele será salvo:



Na tela seguinte, deixe as configurações padrões e clique em “Next”:

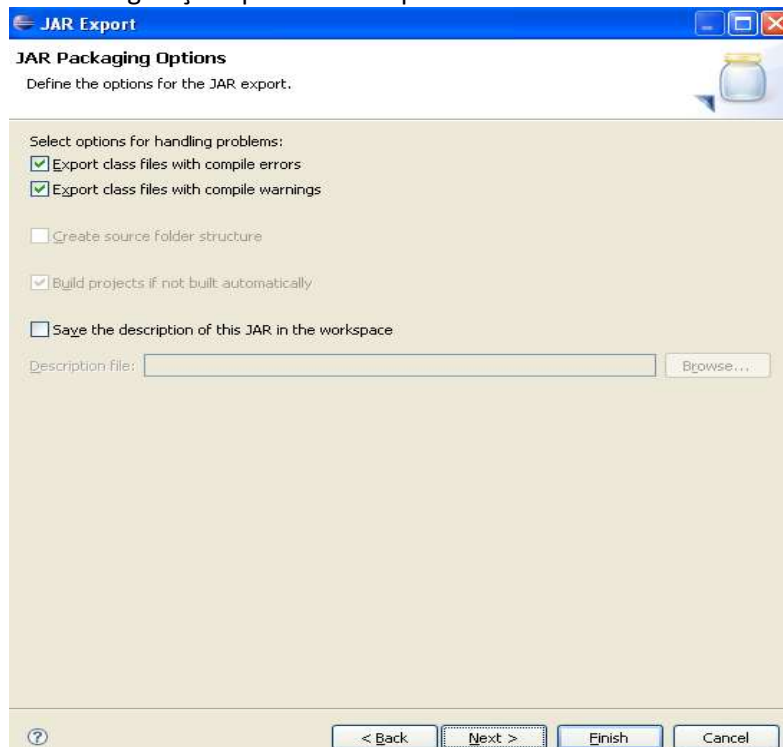


Figura 17

Para finalizar, informe qual a classe principal de sua aplicação e clique em “Finish”:

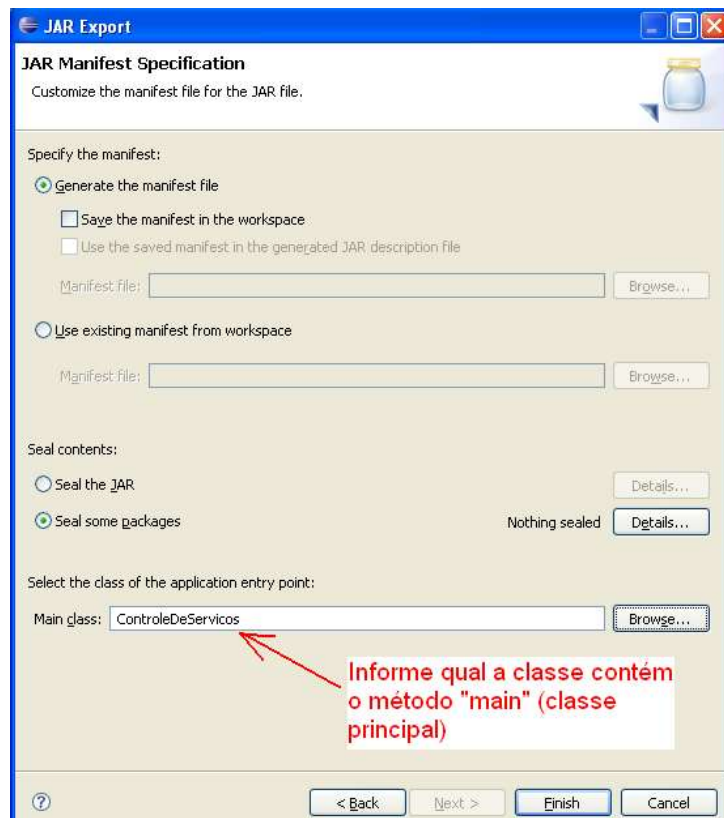


Figura 18

Pronto, o arquivo “Jar” está criado. Agora precisamos executá-lo. Para isso, devemos usar o comando: `java -jar ./NomeDoArquivoJar.jar`. Mas para facilitar vamos criar um arquivo “.bat”. Crie este “.bat” no mesmo diretório que o arquivo “.Jar” foi salvo.

Veja na Figura abaixo o que este arquivo deve conter.

```
@echo off

if exist \"./PilantraManutencaoDeComputadores.jar\" goto programaEncontrado
echo Nao foi possivel encontrar o programa especificado.
pause
goto fim

:programaEncontrado
java -jar ./PilantraManutencaoDeComputadores.jar
:fim
```

OBS: Pra criar um arquivo “.bat” basta abrir o Bloco de Notas e salvar o arquivo com a extensão “.bat”.

Por fim, dê dois cliques no arquivo “.bat” para executá-lo e verifique se sua aplicação está rodando normalmente.

Pontuação

Tópico	Valor
Diagrama de Classe e java doc	1,0
Implementou o módulo de comunicação com GoogleDrive	3,0
Manipulação do arquivo Json	3,0
Validação do cadastros (datas, etc..)	1,0
Implementou o módulo de criação das informações (imagens).	3,0
Criou o .bat	1,0
Implementação geral dos requisitos entre as classes	3,0
Interface com usuário agradável e com tratamento de erros de entrada	2,0
Criou uma boa estrutura de classes	2,0
Sou bom aluno e implementei algo a mais para melhorar o sistema	1,0

Fim