



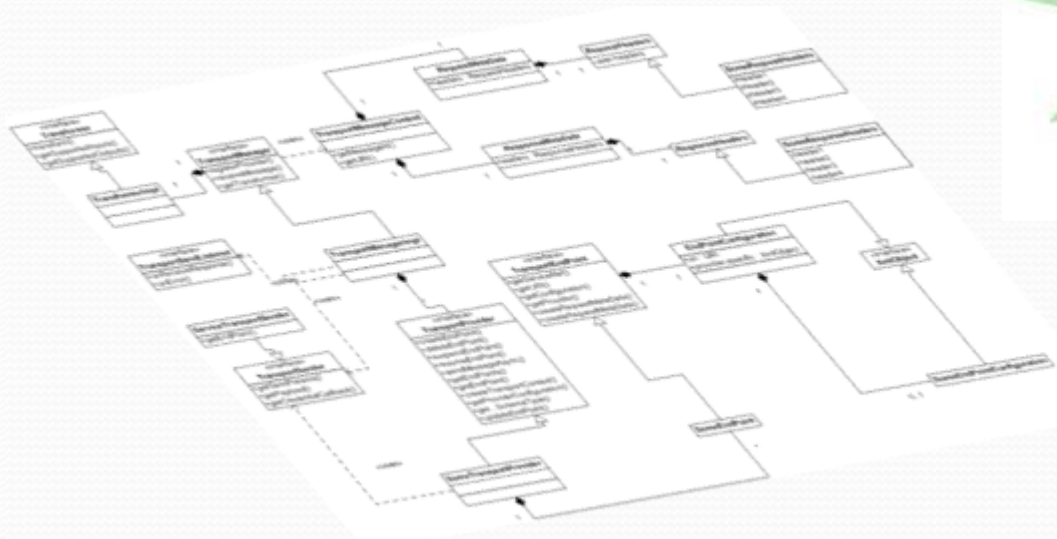
Experimento

Bad smells em diagramas de classe UML

Bruno Liberal de Araújo
Flávio Airjan Madureira
Tamara Márcia Mendes

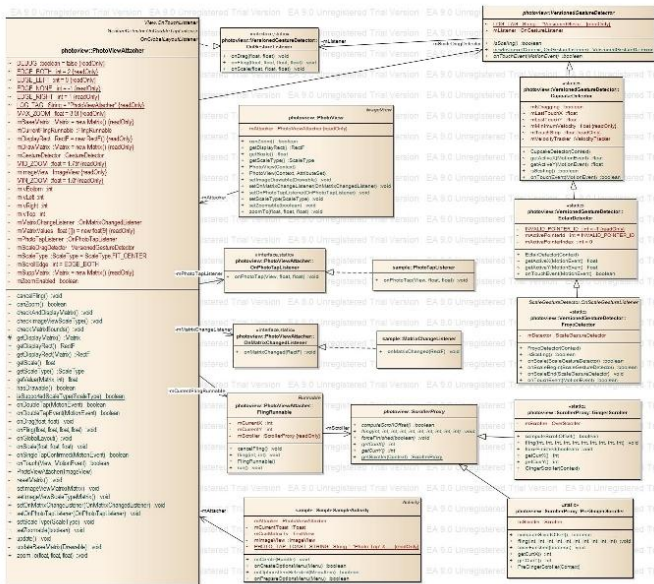
Hipótese

É possível identificar *bad smells* no código de um sistema usando como fonte de informação apenas o diagrama de classes UML?



Planejamento

- Extração dos diagramas



Planejamento

- Análise do histórico de alterações
- Formulários
 - Experiência
 - *Bad smells*
 - Métricas
 - Tempo
- Execução do experimento
- Análise dos dados
- Conclusão



Dados do GitHub

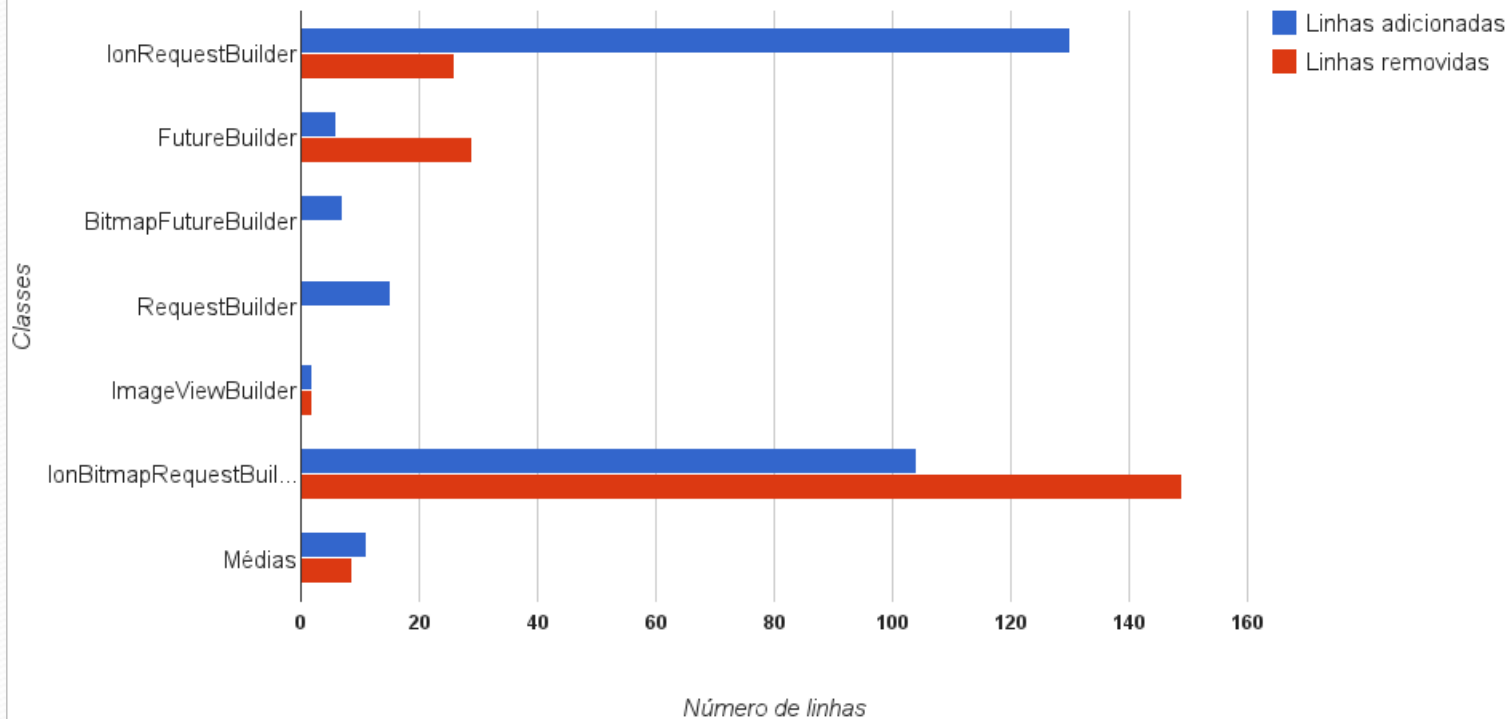
A análise feita a partir do GitHub seguiu os seguintes critérios:

- Número de linhas adicionadas
- Número de linhas retiradas
- Classe refatorada e classe não refatorada

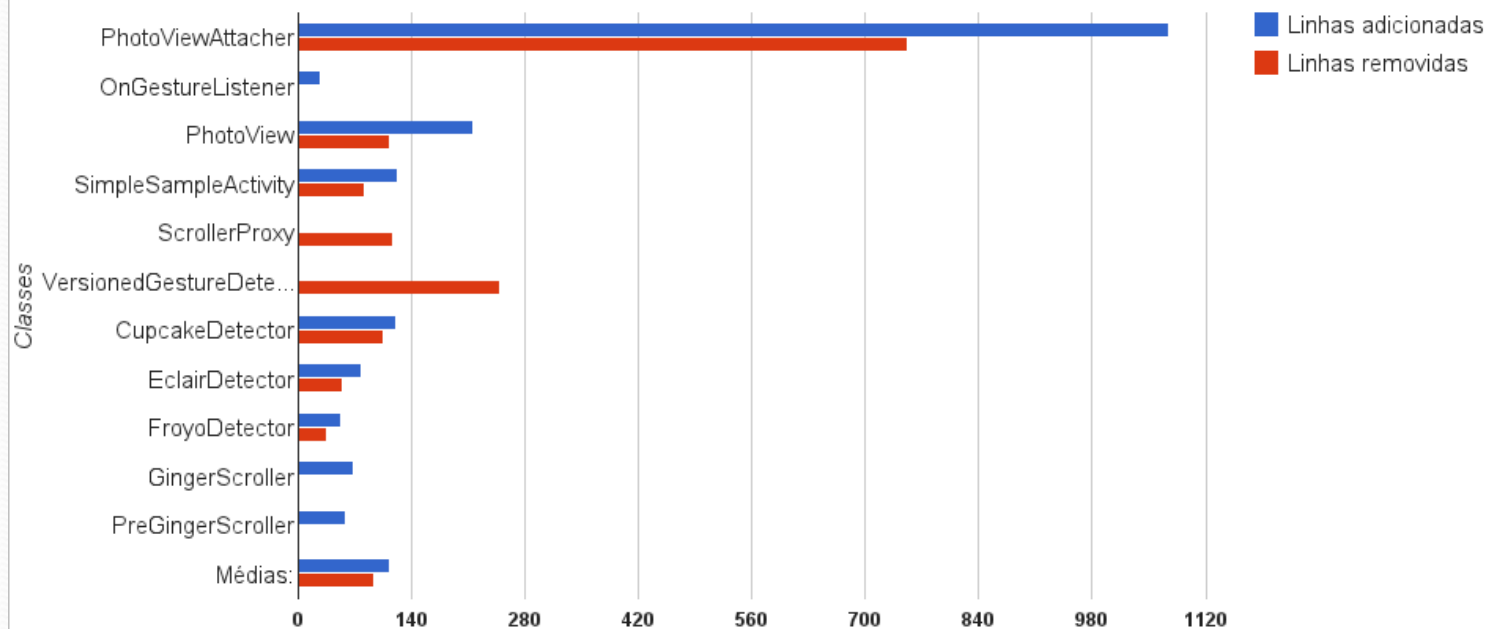
Métricas Análise do Código

| <i>Bad smell</i> | Com refatoração | Sem refatoração |
|------------------------|---|--------------------------------------|
| Shotgun surgery | Linhas removidas Linhas não adicionadas Coesão | Linhas modificadas Acoplamento |
| God class | Linhas removidas Linhas não adicionadas | Linhas modificadas |
| Lazy class | Classe removida Linhas adicionadas Linhas não removidas | Poucas ou nenhuma linhas modificadas |

GitHub-Ion



GitHub-PhotoView



Número de Linhas

Classes refactoradas

Ion:

- LoadBuilders
- ImageViewFutureBuilder
- LoadImageViewFutureBuilder

PhotoView:

- Nenhuma classe refactorada

Conhecimento dos participantes

Orientação a objetos

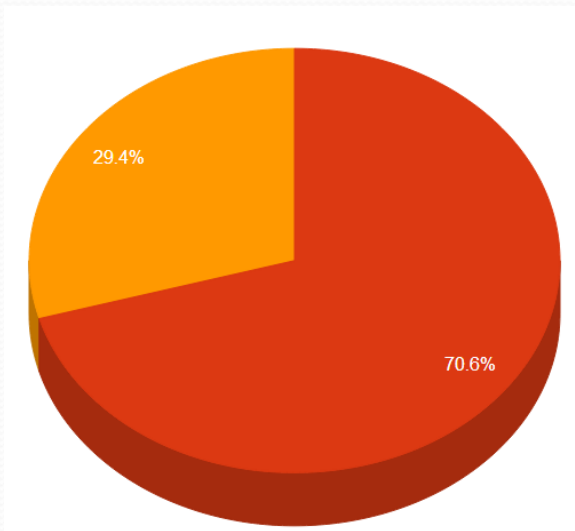
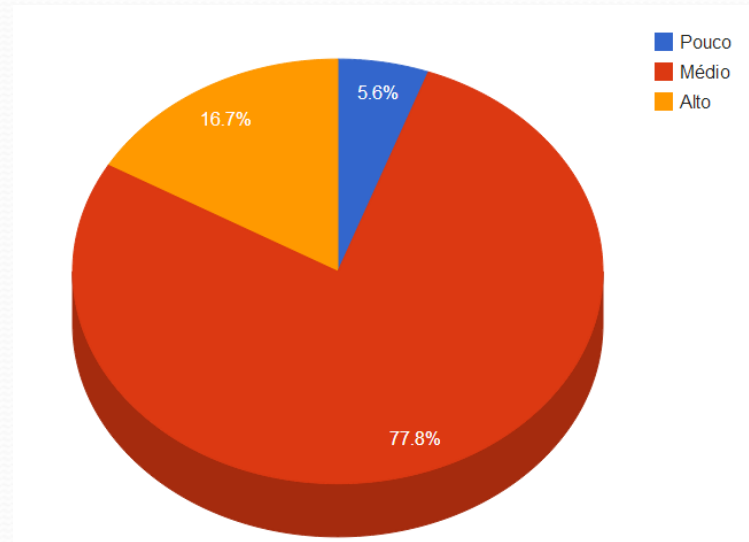


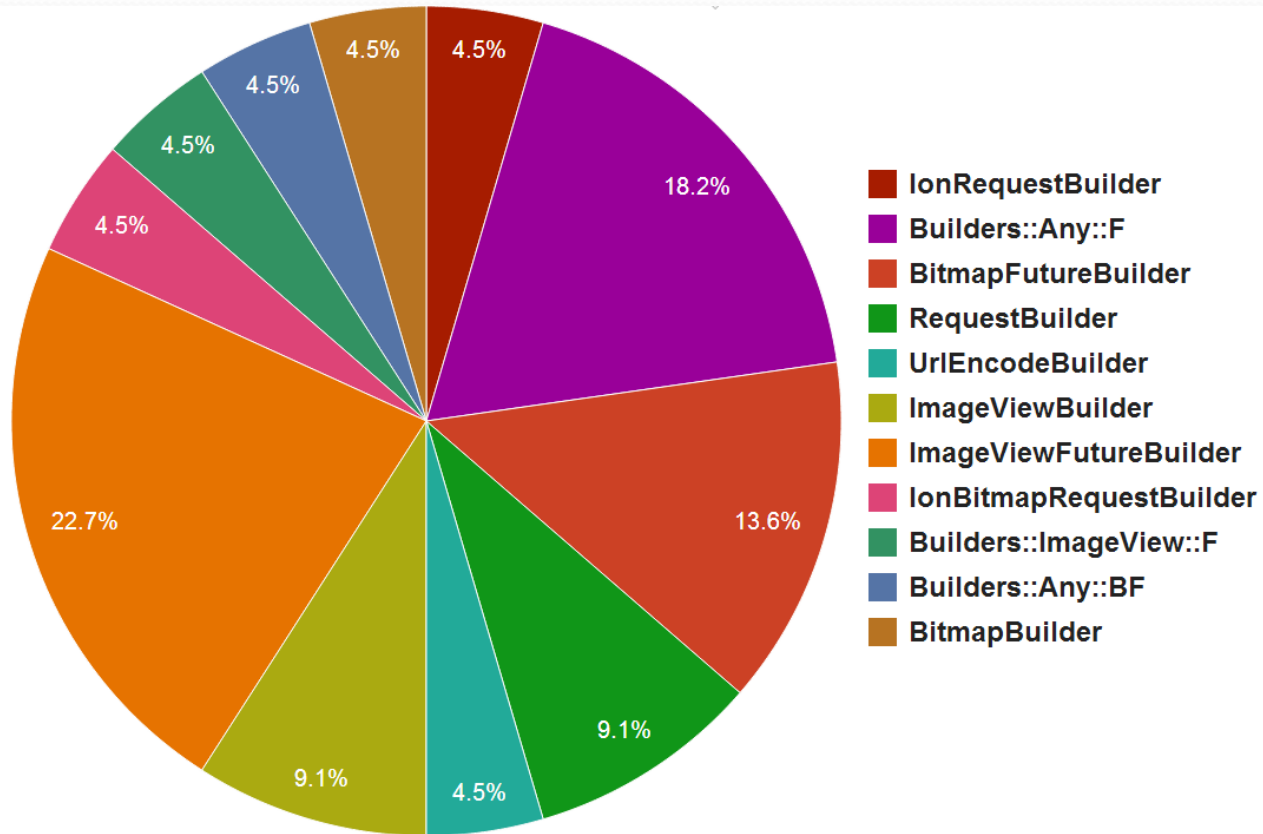
Diagrama de Classes



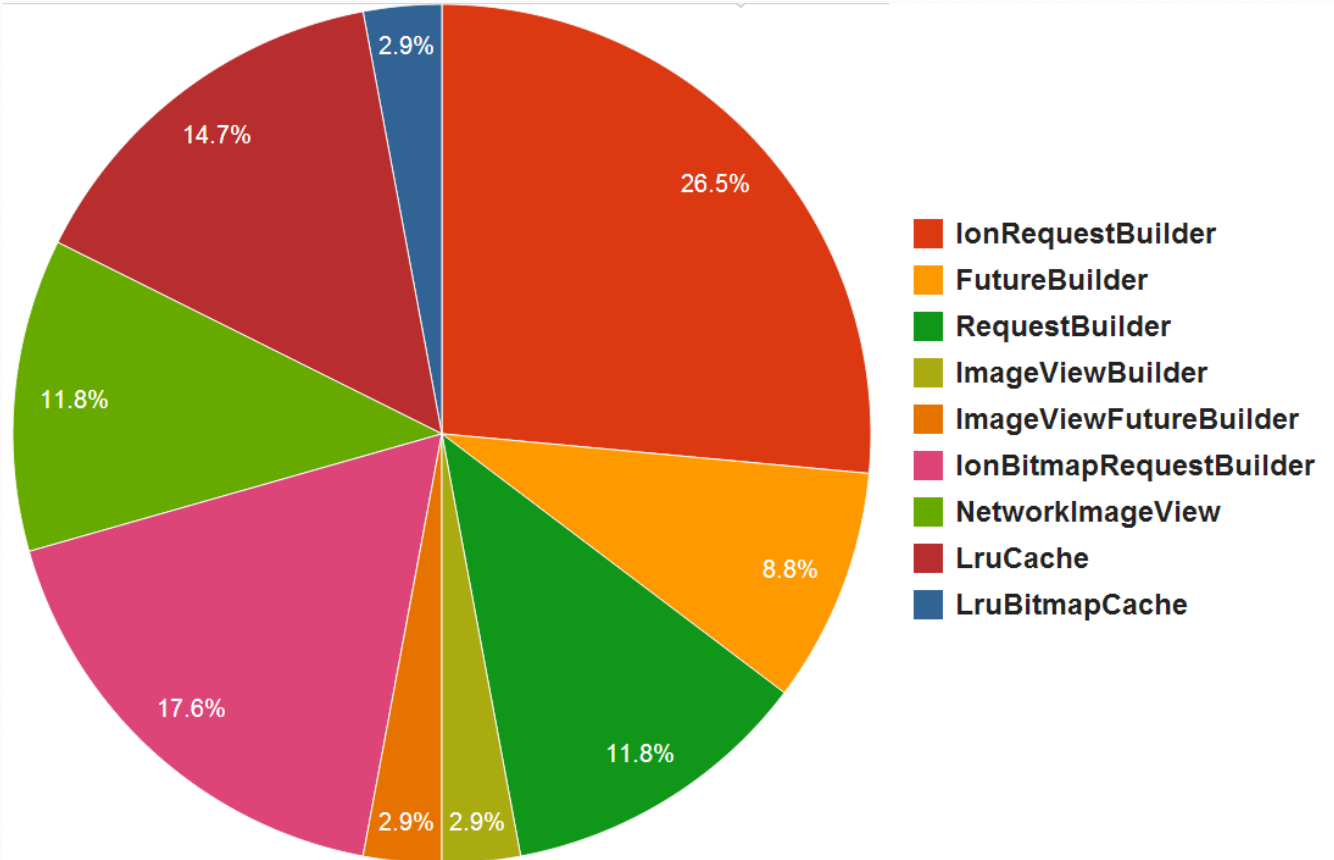
Análise dos participantes

- Tempo médio de 47 minutos
- Métricas
 - Shotgun surgery - CBO, NOC e DIT
 - God Class - NOA e NOO
 - Lazy Class - NOA e NOO

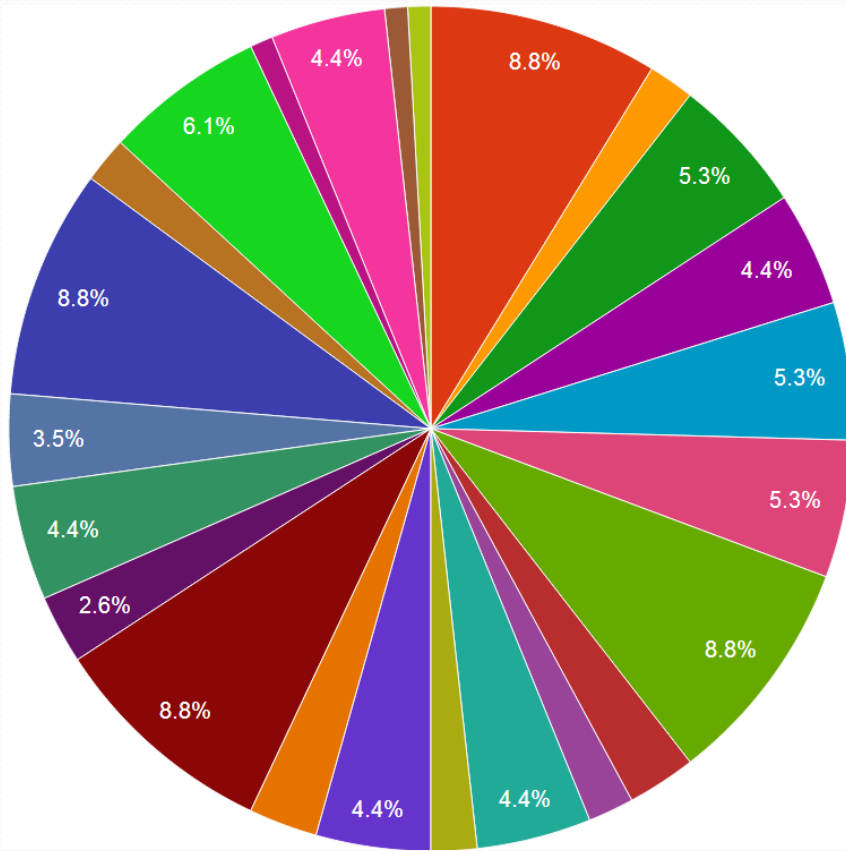
Ion - Shotgun Surgery



Ion - God Class

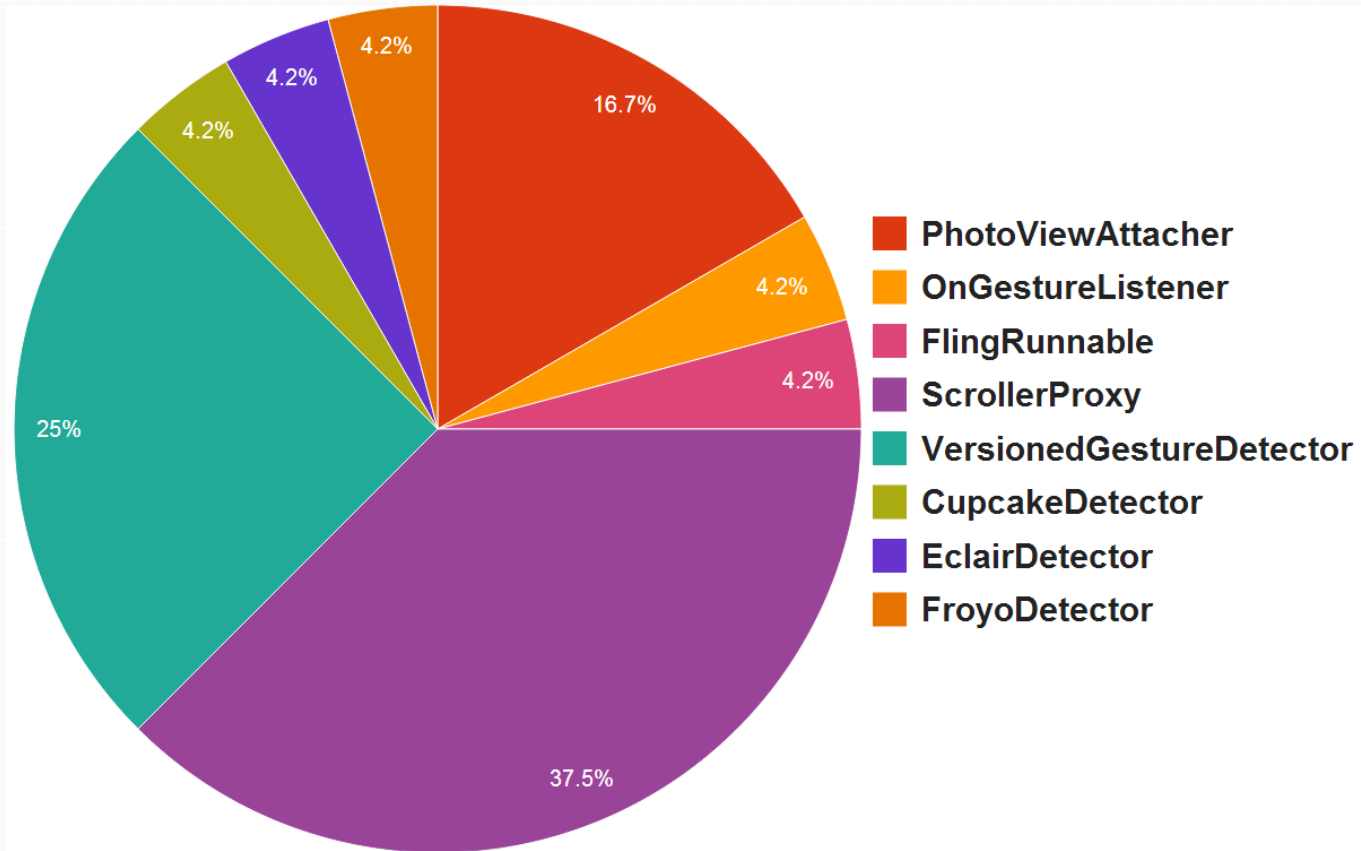


Ion - Lazy Class

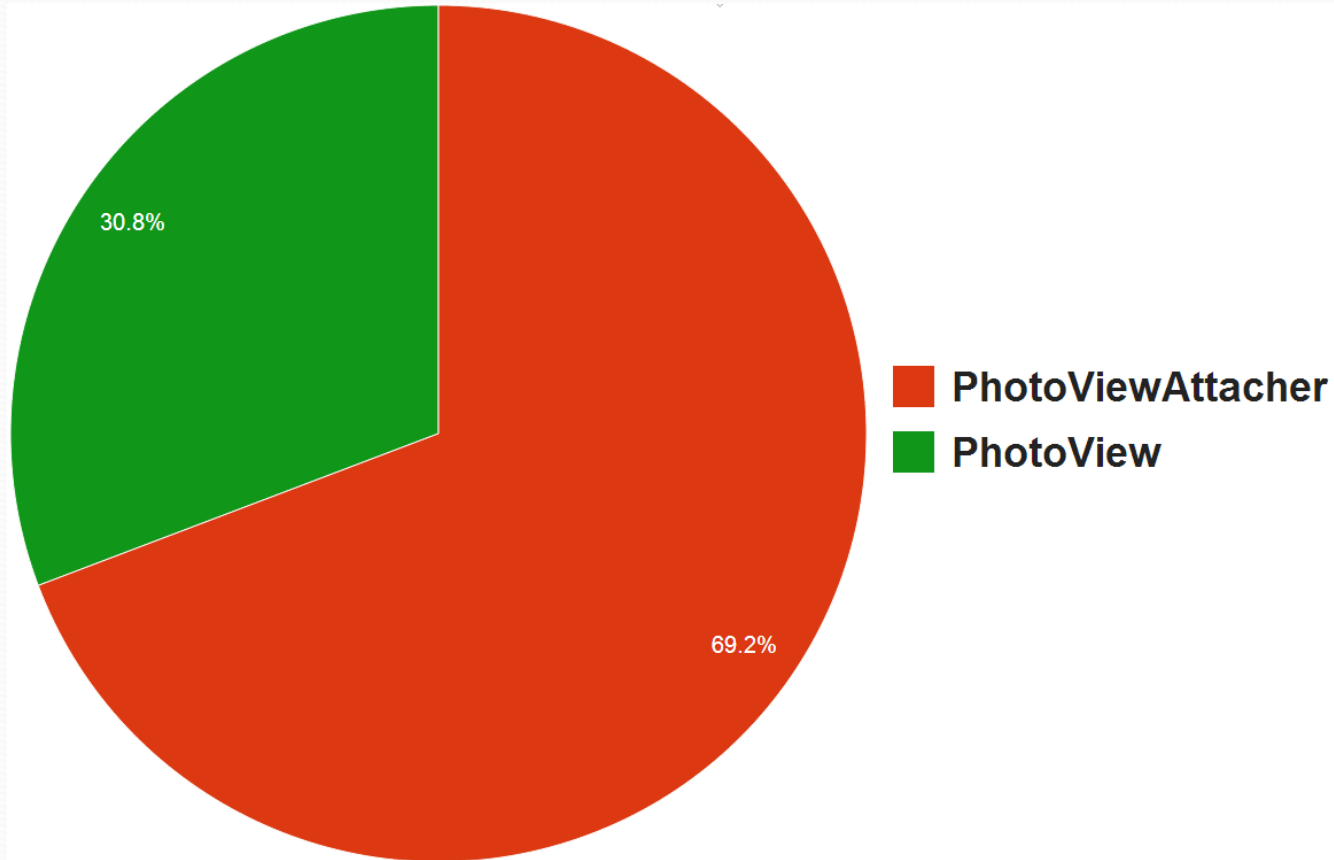


- IonRequestBuilder
- LoadBuilders
- Builders::Any::B
- Builders::Any::F
- Builders::Any::M
- Builders::Any::U
- FutureBuilder
- BitmapFutureBuilder
- MultipartBodyBuilder
- UrlEncodeBuilder
- ImageViewBuilder
- Builders::Any::IF
- ImageViewFutureBuilder
- IonBitmapRequestBuilder
- LoadImageViewFutureBuilder
- Builders::ImageView::F
- Builders::Any::BF
- NetworkImageView
- BitmapBuilder
- Builders::ImageView
- EmitterTransform
- ResponseFuture
- LruCache
- LruBitmapCache

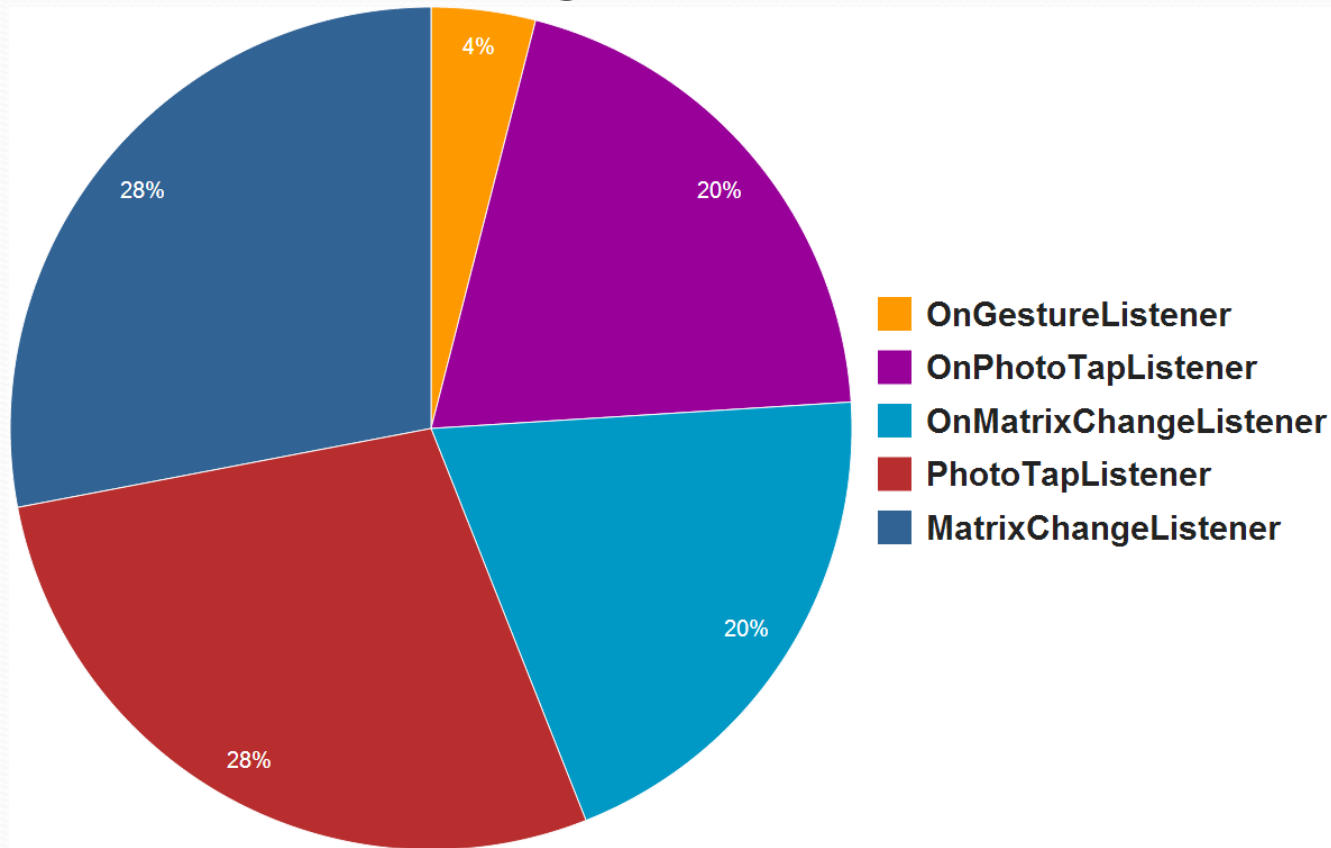
PhotoView - Shotgun Surgery



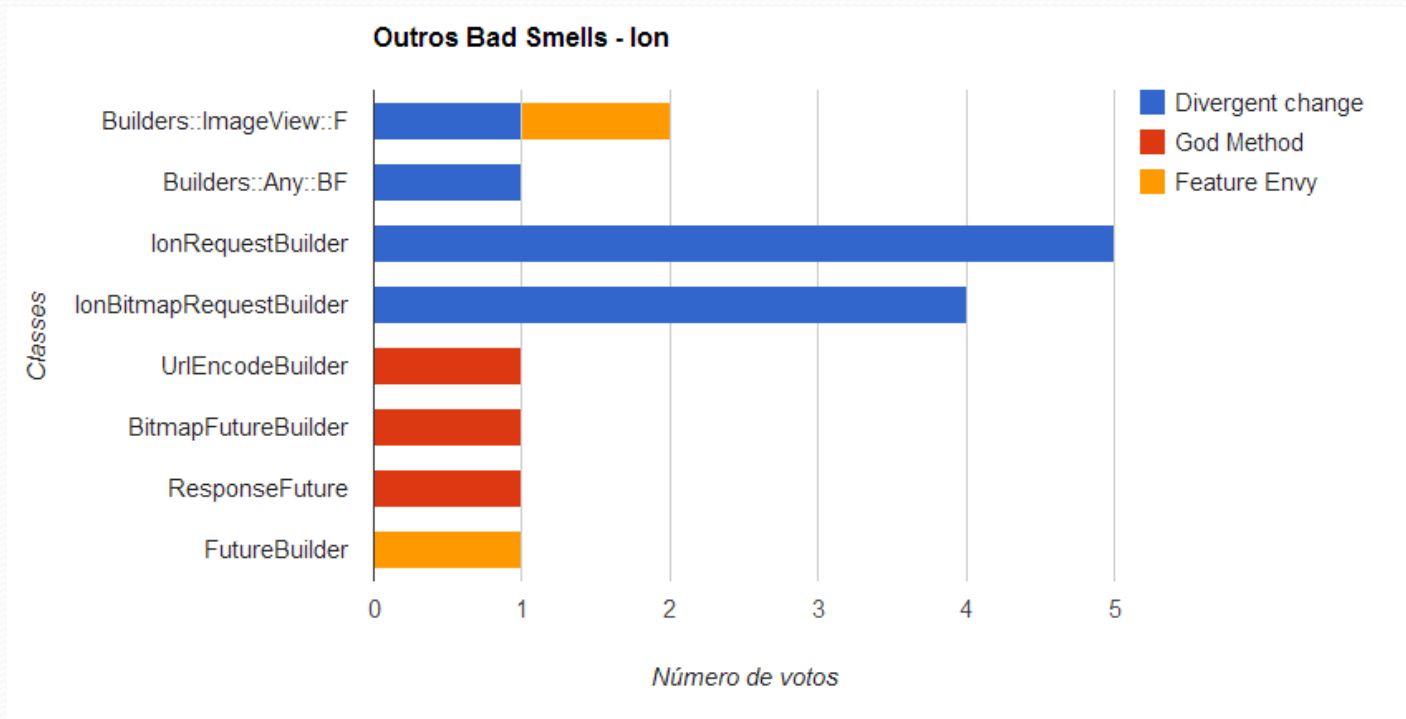
PhotoView - God Class



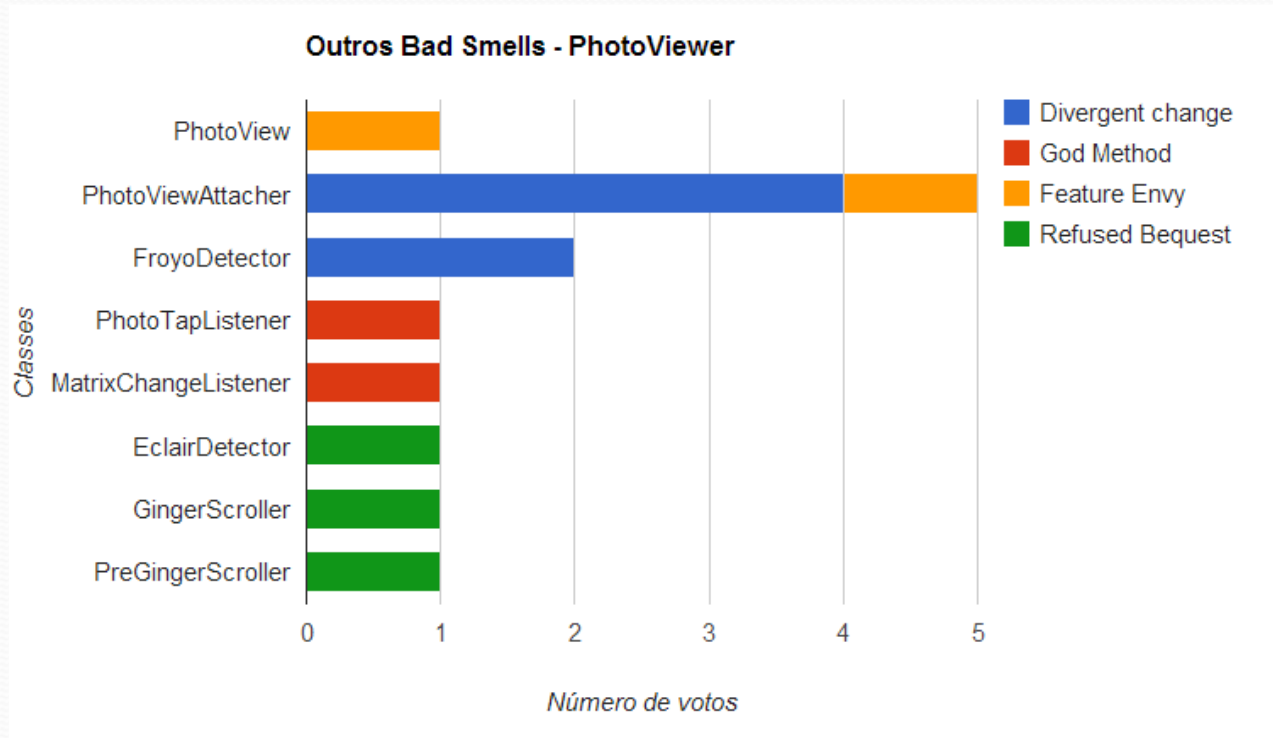
PhotoView - Lazy Class



Outros *bad smells*



Outros *bad smells*



Resultados Ion

| <i>Bad smell</i> | Participantes | Repositório |
|-------------------------|--|---|
| Shotgun Surgery | ImageViewFutureBuilder, Builders::Any::F, BitmapFutureBuilder. | IonBitmapRequestBuilder. |
| God Class | <i>IonRequestBuilder,</i> <i>IonBitmapRequestBuilder.</i> | <i>IonRequestBuilder,</i> <i>IonBitmapRequestBuilder.</i> |
| Lazy Class | <i>Builders::ImageView,</i> <i>Builders::Any::B (F, M e U).</i> | <i>Builders::ImageView,</i> <i>Builders::Any::B (F, M e U),</i> ... |

Resultados PhotoView

| <i>Bad smell</i> | Participantes | Repositório |
|-------------------------|--|--|
| Shotgun Surgery | <i>ScrollerProxy,</i> <i>VersionedGestureDetector.</i> | PhotoViewAttacher, <i>ScrollerProxy,</i> <i>VersionedGestureDetector.</i> |
| God Class | <i>PhotoView,</i> <i>PhotoViewAttacher.</i> | <i>PhotoView,</i> <i>PhotoViewAttacher.</i> |
| Lazy Class | <i>PhotoTapListener,</i> <i>MatrixChangeListener,</i> <i>OnPhotoTapListener,</i> <i>OnMatrixChangeListener.</i> | <i>PhotoTapListener,</i> <i>MatrixChangeListener,</i> <i>OnPhotoTapListener,</i> FlingRunnable, <i>OnMatrixChangeListener.</i> |

Conclusão

- É possível identificar determinados *bad smells* em diagramas de classe
- Depende da particularidade do sistema
- Verificação de classe problemática depende do especialista
- Hipótese foi verificada parcialmente
- Futuros experimentos

