

C O D E R A G E

TDD com Código Legado

Cesar Romero, Trier Sistemas, Desenvolvedor Sênior
cesarliws@gmail.com / @cesarliws



Sobre o Apresentador

- Cesar Romero Silva
 - Programador Analista Sênior e Lider Técnico
Trier Sistemas - Tubarão SC
 - MVP Embarcadero - Delphi
 - Programador Delphi desde 1996
 - Graduação em Ciências Contábeis
 - Pós Graduação em Software Orientado a Objetos



Agenda

- TDD com Código Legado
- O que é Código Legado
- O que é TDD
- Refatorar ou Reescrever?
- Definir um plano de ação
- Code Smell e Anti Patterns
- Refatorar com Testes
- Exemplos

TDD com Código Legado

Código não testado é código legado.

O que é Código Legado

The 5 people in your organization that grow legacy code



O que é Código Legado

[Wikipedia](#)

É um método, tecnologia, computador ou aplicação que continua sendo usado, tipicamente por que ele ainda funciona para as necessidades de alguém, embora tecnologia ou metodologia mais eficiente já esteja disponível.

- Falta de manutenção regular.
- Falta de testes.
- Sistema "preso" a uma plataforma mais antiga.
- Dependencia de bibliotecas de terceiros que não são mais mantidas.
- Código fonte total ou parcial não está mais disponível.

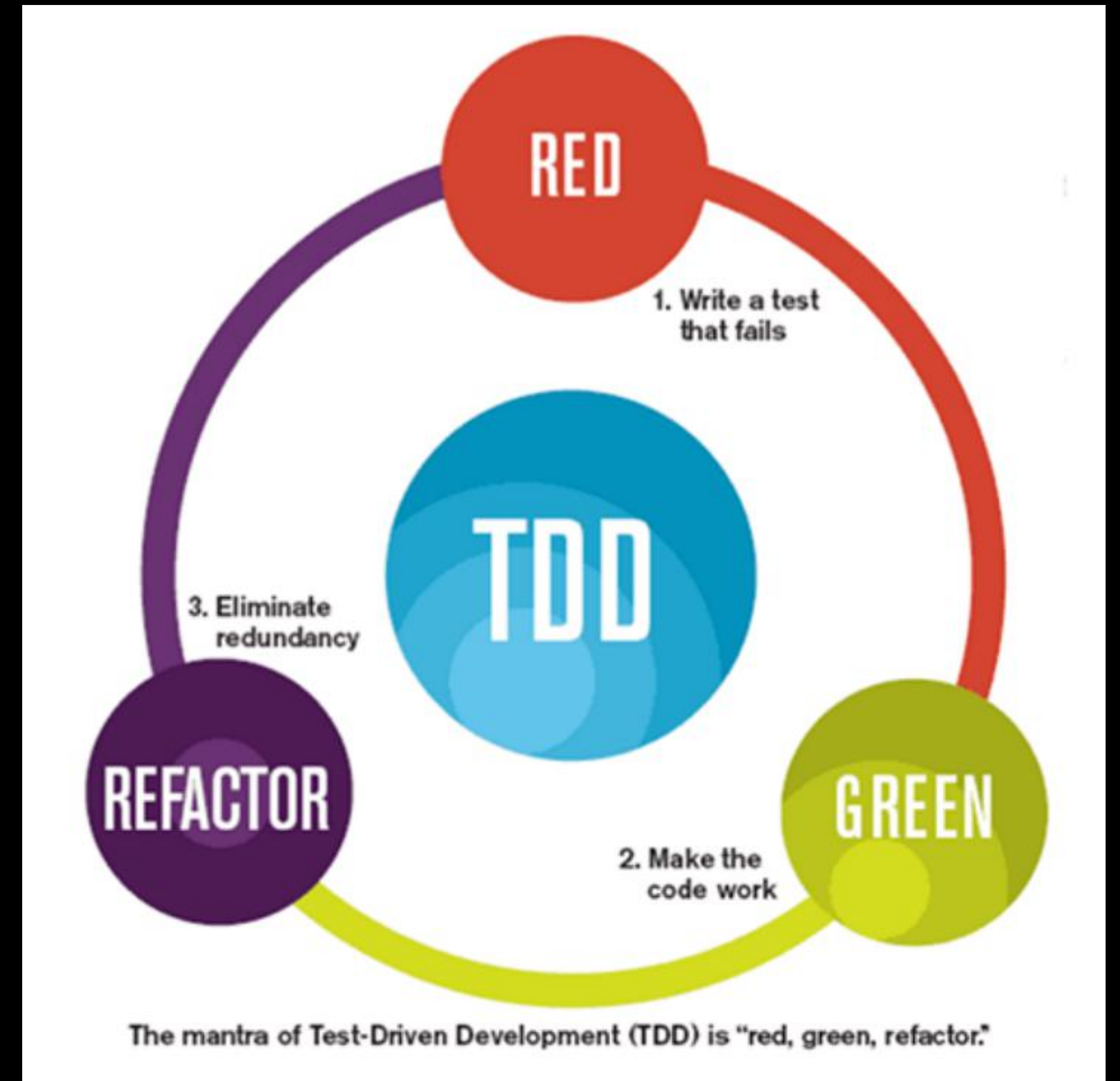
O que é Código Legado

Código legado é o código que alguém tem medo de alterar, por que ele pode quebrar.

Código legado não está relacionado com a idade, você pode escrever código legado desde o início de um projeto.

O que é TDD

- Test Driven Development
Desenvolvimento Dirigido a Testes.



O que é TDD

Kent Beck recebe o crédito como inventor do TDD, ainda assim, ele afirma que ele só redescobriu o TDD.

Existem referências de técnicas de desenvolvimento utilizando técnicas descritas exatamente como TDD em várias situações no passado:

- Report of The Nato Software Engineering Conference (1968).
- The Humble Programmer, Edsger W. Dijkstra (1972).
- De acordo com Craig Larman e Victor Basili, no início dos anos 1960, a IBM executou um projeto para a NASA, usando técnicas equivalentes a TDD.
- Digital Computer Programming D.D. McCracken, 1957

[You won't believe how old TDD is](#)



O que é TDD

Test Driven Design

O teste é valioso, mas o desenvolvimento incremental de uma arquitetura ideal, como efeito colateral do teste não tem preço.

[Allen Holub - Dr. Dobb's](#)

- Design Dirigido a Testes, pois o TDD auxilia no design do código.

Refatorar ou Reescrever

Refatorar

- Refatorar sem testes = insegurança.
- Nunca se sabe onde o código vai quebrar.
- Alterações no banco de dados
- Código dependente de rotinas complexas e não documentadas.
- Código que depende de bibliotecas ou componentes de versão antiga ou descontinuados.
- Referências inválidas.

Refatorar ou Reescrever

Reescrever

- Reescrever Sistema = alto custo.
- Se for escrito sem testes, vai nascer legado.
- Se não foi bem feito da primeira vez, por que seria agora?
- Deve ter uma boa justificativa:
 - Mudança de tecnologia para suportar novas plataformas ou versão mais recente da mesma plataforma.

Definir um plano de ação

- Crie uma branch para o refactory.
- Crie um projeto de testes.
- Definir Metas:
 - Definir como o seu projeto deve ser organizado.
 - Definir o padrão de formatação.
 - Definir o padrão para nome.
 - Definir quais componentes são essenciais e quais devem ser removidos/substituídos.
 - Não adicione um conjunto de componentes a não ser que ele seja realmente necessário.

Definir um plano de ação

- Analytics
 - Identificar que partes do sistema seu cliente usa.
- Cobertura
 - Nem sempre é necessário ter uma cobertura de testes completa
 - Classes e métodos públicos devem ser testados
 - Métodos usados internamente pela classe devem ser movidos para private ou protected

Definir um plano de ação

Por onde começar

- Remover todos Warnings e Hints do projeto.
- Remover código morto
 - Comentários, componentes não usados e desnecessários.
 - Código, variáveis e constantes não usadas.
 - Código duplicado e Código comentado.
 - Formulários e DataModulos não usados.

Definir um plano de ação

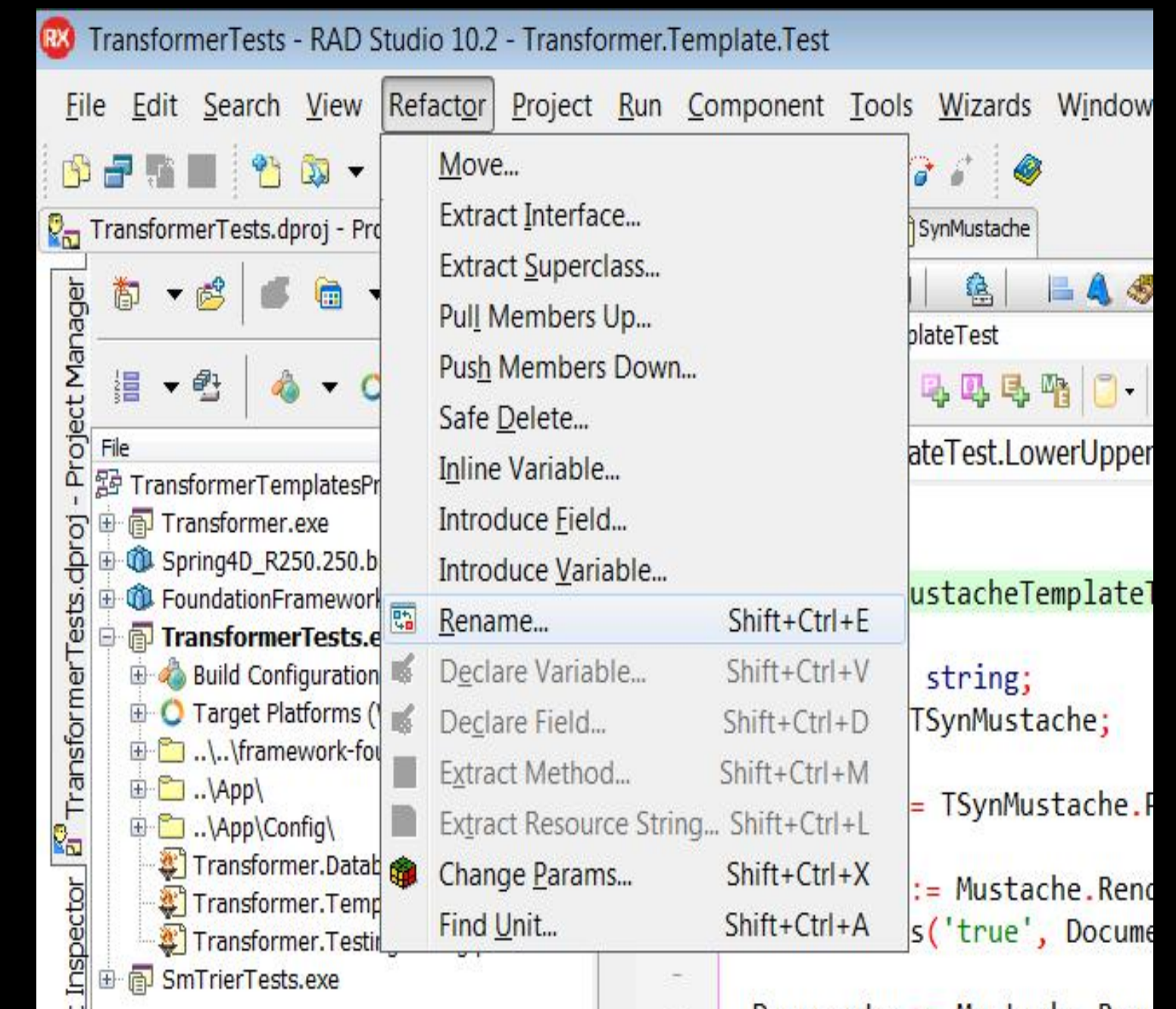
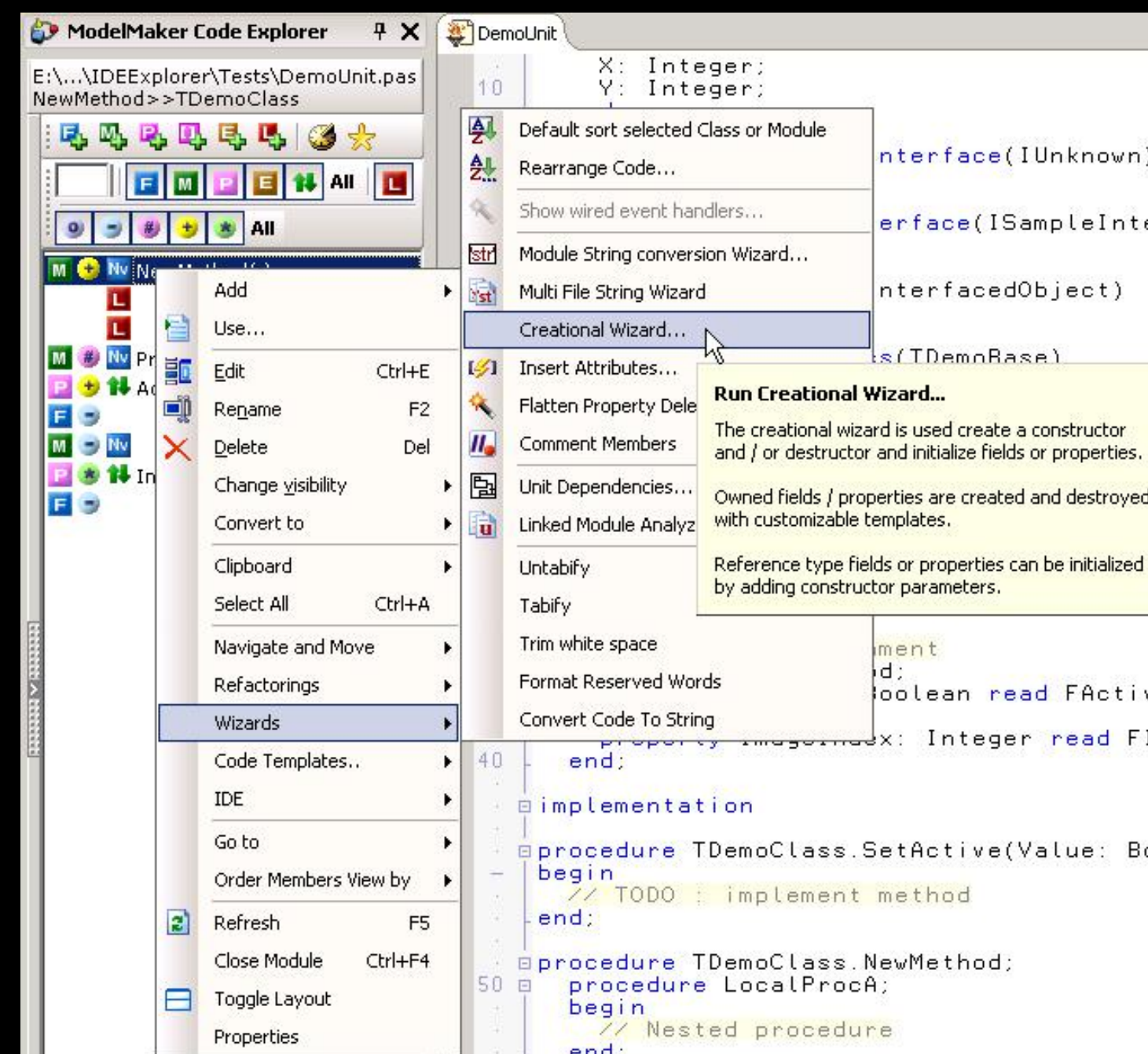
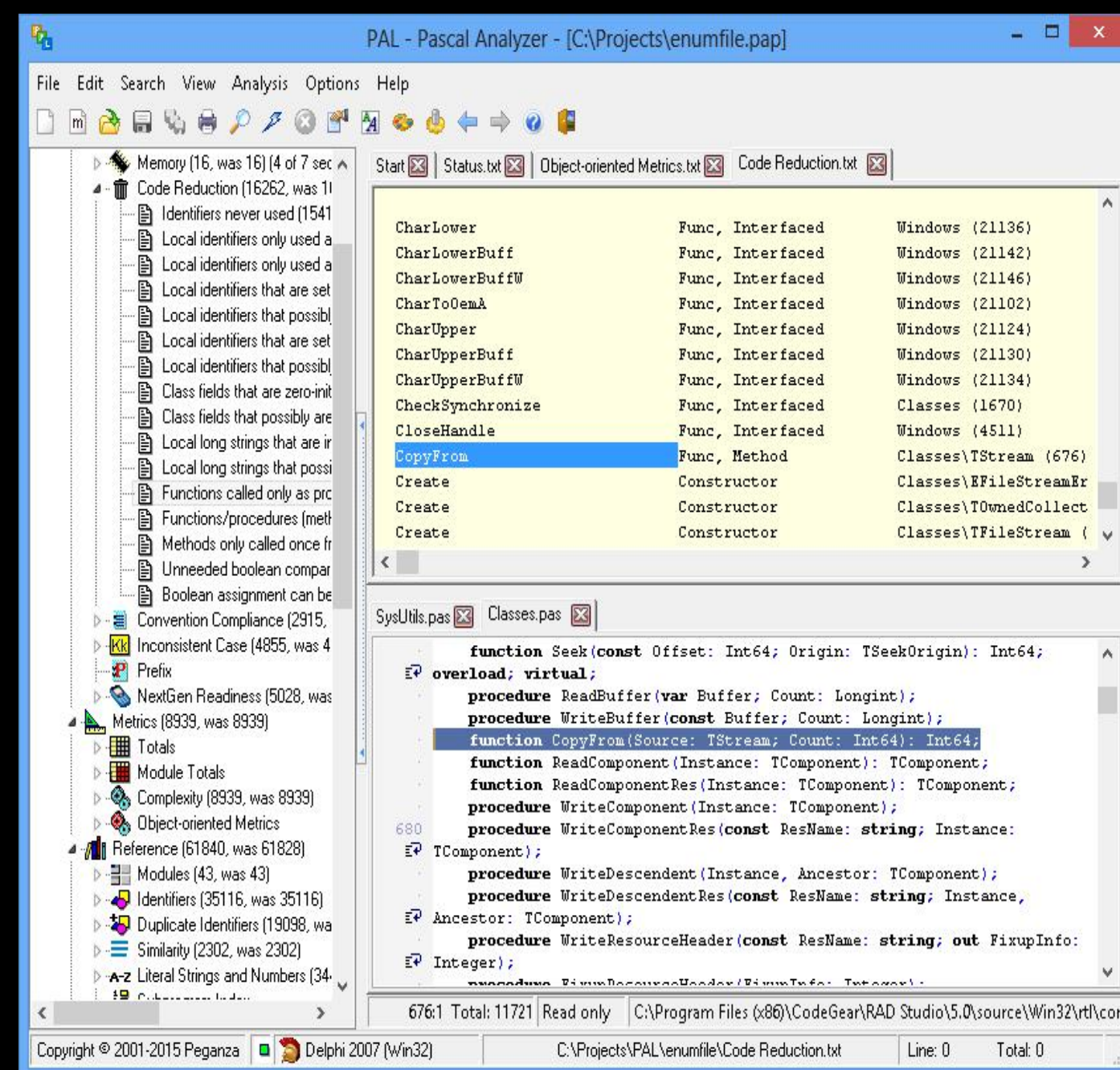
Por onde começar

- Renomear identificadores para ter um código mais claro.
 - Se você não consegue descrever uma classe ou um método em uma sentença, então há muita responsabilidade .
- Remover dependências - units não usadas nas "uses".
- Mover para uses da implementation, as units de dependências que não são usadas na interface.
- Utilize ferramentas para refactory e para identificar código morto.

Definir um plano de ação

Utilize ferramentas para refactory e para identificar código morto.

- [Peganza Pascal Analyzer](#)
- [MMX - ModelMaker Code Explorer](#)
- [Menu Refactor do Delphi](#), [CnWizards](#), [GExperts](#), [TMS FixInsight](#)



Code Smell e Anti Patterns

Code Smell

- É qualquer sintoma no código que possivelmente indica um problema maior.

[Coding Horror - Code Smell](#)

O que são Anti Patterns?

- É tudo que que fazemos repetidamente e que produz algum resultado negativo.

[Common AntiPatterns](#)

[Software Development AntiPatterns](#)

Refatorar com Testes

- Somente altere código que está em produção com testes.
- Se não houver testes, antes de alterar o código escreva os testes.
- Se houver dependências, isole o código sem alterar.
- Utilize "Adapters" para manter a integração entre as partes.
- Correção de Bugs
 - Para cada problema reportado, deve haver um teste reproduzindo o problema.
 - Foco em apenas um teste de falha por vez.

Exemplos

- Criando Projeto de Testes.
- Organizando código que utiliza componentes de terceiros.
- Extraindo código de Formulários e DataModulos.
 - Criando um Adapter.
- Testando código de banco de dados
 - Extrair SQL
 - DataModules
- Testando procedures e functions.

Exemplos

- Separar configuração de inicialização de parâmetros do sistema.
- Mover parâmetros do sistema para o banco de dados.
- Remover código do .DPR
- Extrair SQL dos fontes
- Extrair regra de negócio de formulários e data modules

Exemplos

Hands ON - FireDAC Getting Started SQLite

- Melhorias
 - Abrir o banco automaticamente
- Backlog - Débito Técnico
 - Separar componentes de acesso ao banco de dados
 - Extrair SQL do código do formulário
 - Extrair regra de negócios
 - Remover units não utilizadas
 - Extrair Strings
 - Remover "with"
 - Padronizar nome das units
 - Padronizar nome dos componentes

C:\Users\Public\Documents\Embarcadero\Studio\19.0\Samples\Object Pascal\Database\FireDAC\Samples\Getting Started

C O D E R A G E *XII*

C O D E R A G E

Perguntas

 embarcadero[®] NOVEMBER 7-9, 2017

 embarcadero

C O D E R A G E *XII*

C O D E R A G E

<https://github.com/cesarliws/CodeRageBrasil2018>

