Aplicação de Teste de Unidade Baseado em Designs Combinatoriais no Ambiente *Geotools*

SER 300 – Introdução ao Geoprocessamento

Juliana Marino Balera

Agenda

- Introdução
- Fundamentação Teórica
- *GeoTools*
- Experimento
- Conclusões

Contexto e Motivação

- O Brasil é um país que convive constantemente com notícias relacionadas a desastres naturais;
- Geoprocessamento
  - Gestão de catástrofes;
  - Geração de planos de emergência;
  - Plano de prevenção;
  - Entre outros;
- Disponibilização de recursos computacionais relacionados a prevenção desses tipos de acidentes;
- Manipulação de ferramentas exige conhecimentos de engenharia de software.

Contexto e Motivação

- Ferramentas computacionais estão suscetíveis a falhas;
- Testar requer conhecimento técnico;
- Testar de maneira exaustiva é infactível.
  - Executar e analisar os resultados, em tempo hábil, de um conjunto de casos de teste extremamente grande.

Contexto e Motivação

- Geração/Seleção de casos de teste → atividade do processo de Teste de Software.
- *Designs* Combinatoriais
  - Simples de aplicar;
  - Revelação de falhas → cobertura de todas as interações de fatores;
  - Simples de gerar;
  - Conjuntos de casos de teste menores.

Objetivo

Realizar uma análise do código fonte da ferramenta *GeoTools* através de casos de teste gerados por meio de *designs* combinatoriais no contexto do teste de unidade.

- Casos de teste gerados de maneira rápida e fácil sem muito conhecimento técnico.

Algoritmo TTR

- Geração de casos de teste por meio de *Designs* Combinatoriais por através da realocação de elementos de matriz de interações.

Algoritmo TTR

```
public void metodoQualquer(boolean x1, boolean x2, ClasseQualquer x3){}
```

Fator	Nível
x1	$true$, $false$
x2	$true$, $false$
x3	$tipo1$, $tipo2$, $tipo3$

Algoritmo TTR

i	**x1**	**x2**	**x3**
0	$true$	$true$	
1	$true$	$false$	
2	$false$	$true$	
3	$false$	$false$	
4	$true$		$tipo1$
5	$true$		$tipo2$
6	$true$		$tipo3$
7	$false$		$tipo1$
8	$false$		$tipo2$
9	$false$		$tipo3$
10		$true$	$tipo1$
11		$true$	$tipo2$
12		$true$	$tipo3$
13		$false$	$tipo1$
14		$false$	$tipo2$
15		$false$	$tipo3$

i	**x1**	**x2**	**x3**
0	$true$	$true$	$tipo1$
1	$true$	$false$	$tipo2$
2	$true$	$true$	$tipo3$
3	$false$	$false$	$tipo1$
4	$false$	$true$	$tipo2$
5	$false$	$false$	$tipo3$

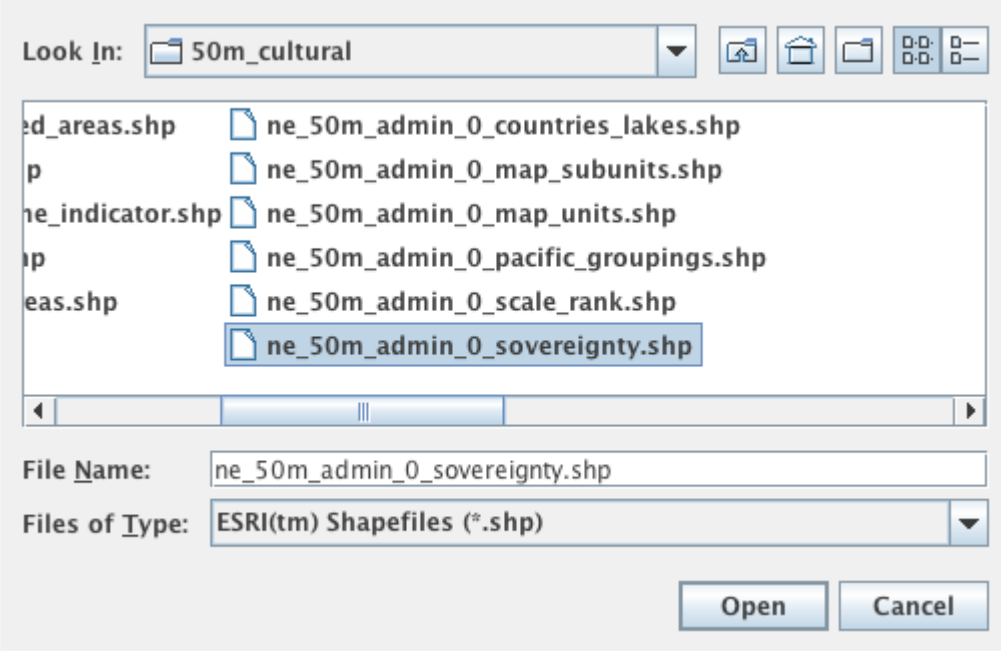
GeoTools

- Uma biblioteca Java que implementa um Sistema de Informações Geográfico
  - Definição de interfaces para a utilização de conceitos geo;
  - API de acesso a dados espaciais dos mais diversos formatos;
  - Utilização de formatos adicionais através da adição de *plugins* disponibilizados na biblioteca;
  - Operações típicas sobre mapas;

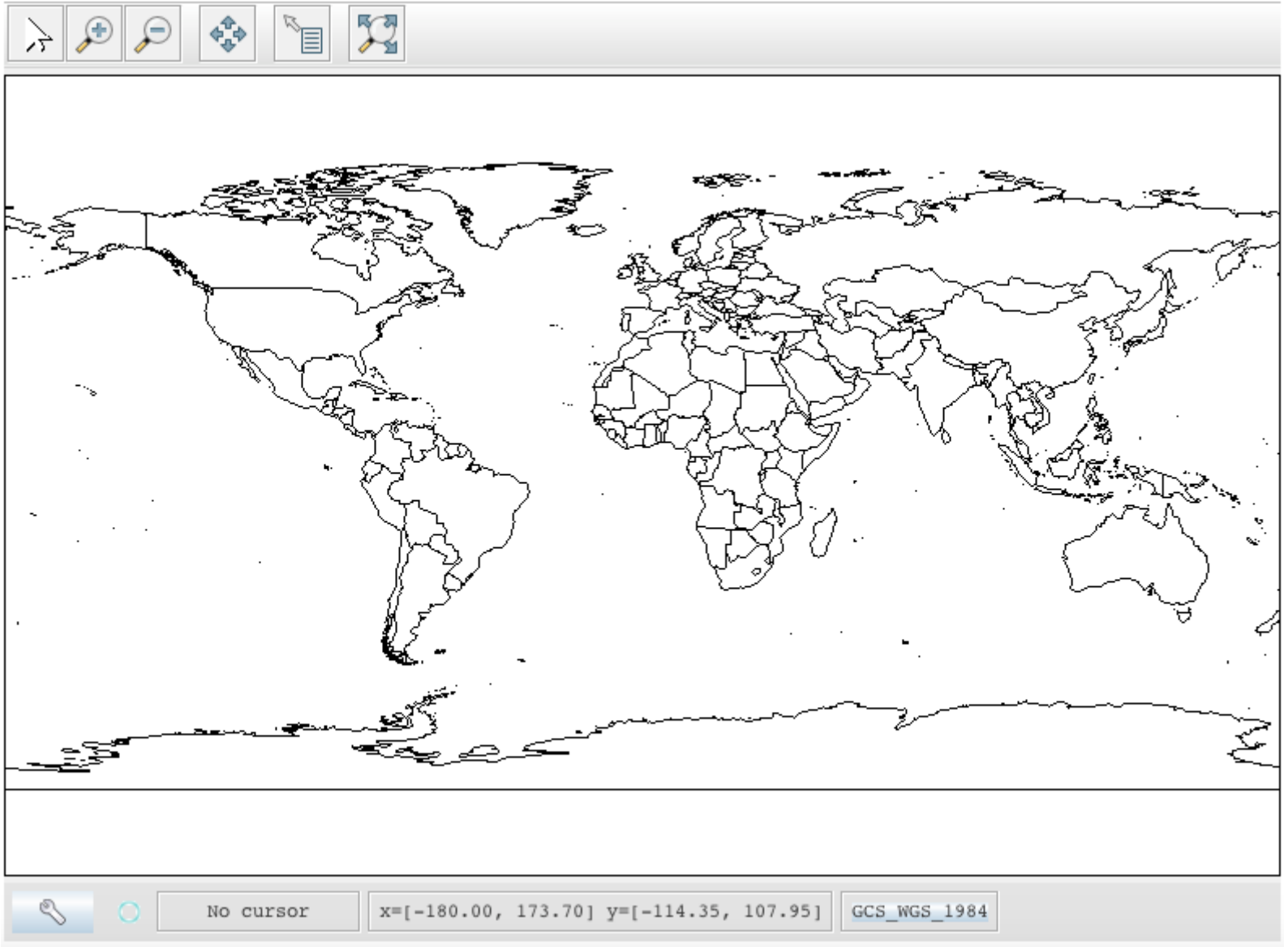
GeoTools

- Implementa os padrões da Open Geospatial Consortium (OGC)

GeoTools


Look In:
50m_cultural
ed_areas.shp
p
ne_indicator.shp
ap
eas.shp
ne_50m_admin_0_countries_lakes.shp
ne_50m_admin_0_map_subunits.shp
ne_50m_admin_0_map_units.shp
ne_50m_admin_0_pacific_groupings.shp
ne_50m_admin_0_scale_rank.shp
ne_50m_admin_0_sovereignty.shp
File Name:
ne_50m_admin_0_sovereignty.shp
Files of Type:
ESRI(tm) Shapefiles (*.shp)
Open
Cancel

GeoTools



Experimento

- Foi selecionada a classe *Circle.java*
  - Implementa funcionalidades e características do geo-objeto circunferência.
  - Seleção dos métodos: *linearizeArc* e *tolerance*
- Realização de análise do valor limite de cada uma das variáveis envolvidas → classes de equivalência;

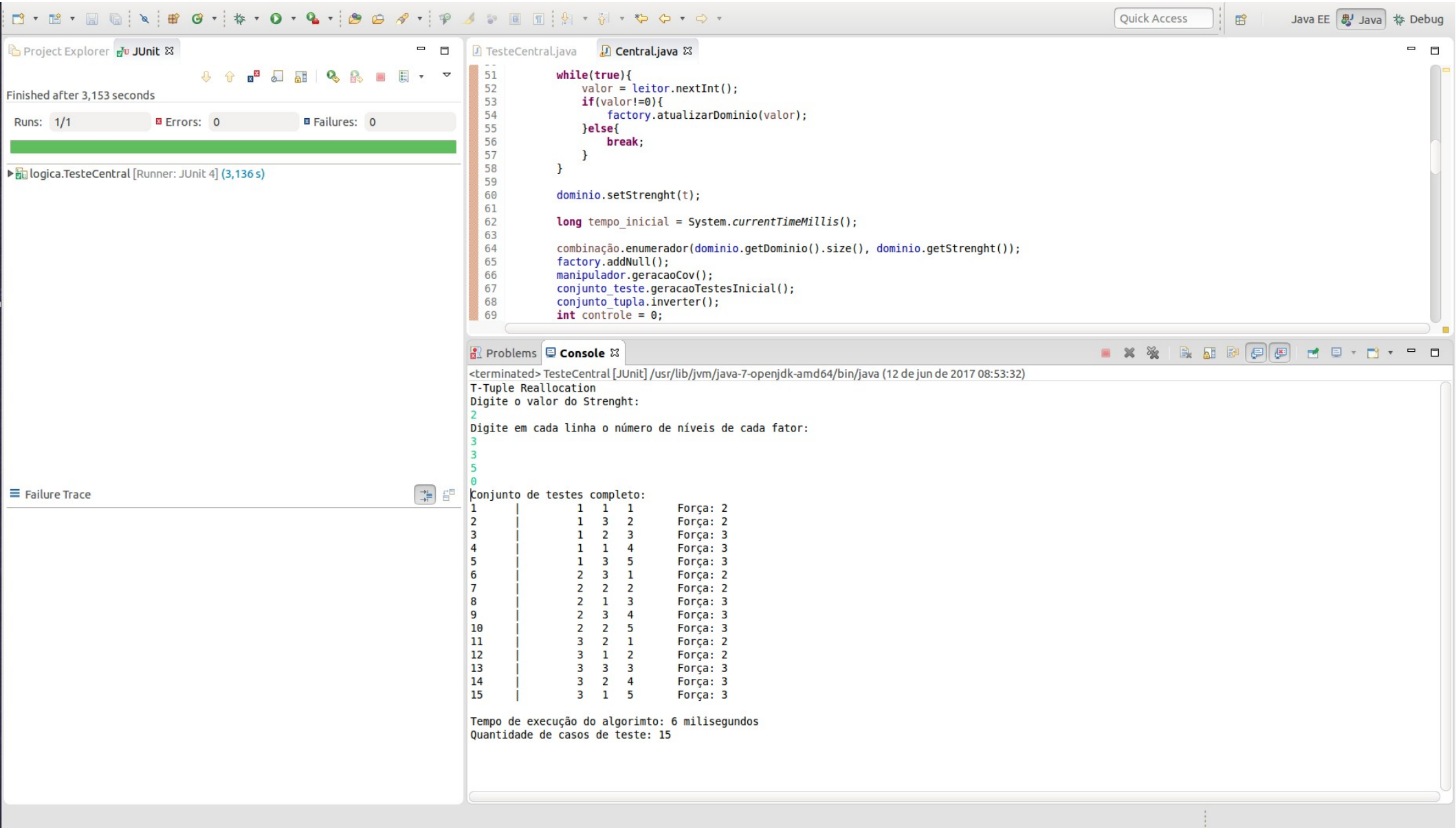
Experimento

métodos	**fatores**	**classes de equivalencia**
linearizeArc	(int) cx	-1, 0, 1
	(int) cy	-1, 0, 1
	(double) radius	-1.0, 0.0, 1.0
	(vector(int)) ordinates	-1, 0, menor 6, 6, maior 6
tolerance	(int) cx	-1, 0, 1
	(int) cy	-1, 0, 1
	(double) radius	-1.0, 0.0, 1.0
	(double) tolerance	-1.0, 0.0, 1.0
	(vector(int)) ordinates	-1, 0, menor 6, 6, maior 6

Experimento

- A partir daqui foi selecionado um conjunto de operações sob mapas para a simulação do experimento;
- Os arquivos *shapefile* foram retirados do repositório: *Natural Earth Data*

Experimento - Geração dos dados de teste



TesteCentral.java | Central.java | *AppTest.java | Circle.java

```
import org.geotools.styling.Style;
import org.geotools.swing.JMapFrame;
import org.geotools.swing.data.JFileDataStoreChooser;


/**
 * Unit test for simple App.
 */
public class AppTest{

    @Test
    public void testeInicial() throws Exception{

         // display a data store file chooser dialog for shapefiles
        File file = JFileDataStoreChooser.showOpenFile("shp", null);
        if (file == null) {
            return;
        }

        FileDataStore store = FileDataStoreFinder.getDataStore(file);
        SimpleFeatureSource featureSource = store.getFeatureSource();

        // Create a map content and add our shapefile to it
        MapContent map = new MapContent();
        map.setTitle("Quickstart");

        Style style = SLD.createSimpleStyle(featureSource.getSchema());
        Layer layer = new FeatureLayer(featureSource, style);
        map.addLayer(layer);

        // Now display the map
        JMapFrame.showMap(map);
    }

    @Test(expected = ArrayIndexOutOfBoundsException.class)
    public void teste01() {
        System.out.println("Caso de teste 01: [ 1 1 1 ]");
        double[] coords = {};
        Circle circle = new Circle(0, -45, -58);
        assertFalse(circle.clockwise(coords));
    }

    @Test(expected = ArrayIndexOutOfBoundsException.class)
    public void teste02(){
        System.out.println("Caso de teste 02: [ 1 1 2 ]");
        double[] coords = {};
        Circle circle = new Circle(0, -105, 0);

```

Problems | Console

No consoles to display at this time.

Resultados

- Foram gerados 30 casos de teste – 15 para o método *linearizeArc* e 15 para o *torelance;*
- Os casos de teste foram construídos com base nas exceções e retornos dos métodos;
- Nenhum dos casos de teste foi rejeitado.

Conclusões

- Foi utilizada a técnica de teste combinatorial em uma biblioteca que implementa os conceitos de SIG;
- SIG são sistemas complexos que integram diversos serviços → requerem um conjunto de técnicas para testá-lo;
- Profissionais da área de Geoprocessamento que não têm formação na área computacional podem aplicar a metodologia no desenvolvimento dos seus *scripts* !
- Designs combinatoriais foi facilmente aplicado
  - Não exige extenso conhecimento do código
  - Não é necessário saber conteúdo técnico para a geração de casos de teste

Obrigado !