

# CICLOS TDD

**Integrantes:** André Wallas Soares Costa Araújo, Cristiana Santana Silva, Gabriel Fernando Pinheiro de Oliva, Jenifer Henrique Moreira Borges

## 1. Teste não compila

- Não existe a classe Sprint;
- Não existe o método setDtFim(data);
- Não existe o método getDtFim().

```
import java.text.ParseException;
import java.util.Date;
import junit.framework.TestCase;
import org.junit.Test;

public class TesteBurnDown extends TestCase{

    @Test
    public void testaSprintDataFim01Dezembro2014() throws ParseException {

        Sprint sprint = new Sprint();

        @SuppressWarnings("deprecation")
        Date data = new Date(2014,12,01);
        sprint.setDtFim(data);
        assertEquals(data, sprint.getDtFim());
    }
}
```

## 2. Teste falha

- Adicionando classe e métodos

```
import java.util.Date;

public class Sprint {

    private Date dtFim;

    public Date getDtFim() {
        return dtFim;
    }
    public void setDtFim(Date dtFim) {

    }

}

import java.text.ParseException;
import java.util.Date;
import junit.framework.TestCase;
import org.junit.Test;
import br.com.tcc.model.Sprint;
```

```

public class TesteBurnDown extends TestCase{

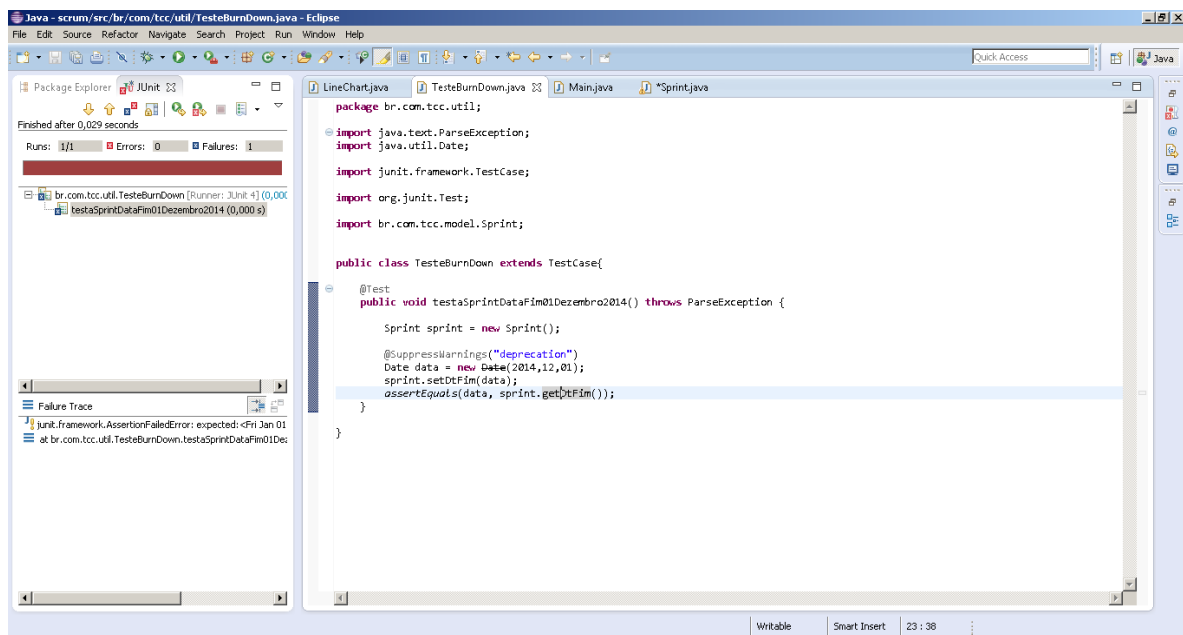
@Test
public void testaSprintDataFim01Dezembro2014() throws ParseException {

    Sprint sprint = new Sprint();

    @SuppressWarnings("deprecation")
    Date data = new Date(2014,12,01);
    sprint.setDtFim(data);
    assertEquals(data, sprint.getDtFim());
}

}

```



### 3. Teste passa

```

import java.util.Date;

public class Sprint {

    private Date dtFim;

    public Date getDtFim() {
        return dtFim;
    }

    public void setDtFim(Date dtFim) {
        this.dtFim = dtFim;
    }

}

import java.text.ParseException;
import java.util.Date;
import junit.framework.TestCase;
import org.junit.Test;
import br.com.tcc.model.Sprint;

```

```

public class TesteBurnDown extends TestCase{

    @Test
    public void testaSprintDataFim01Dezembro2014() throws ParseException

    {

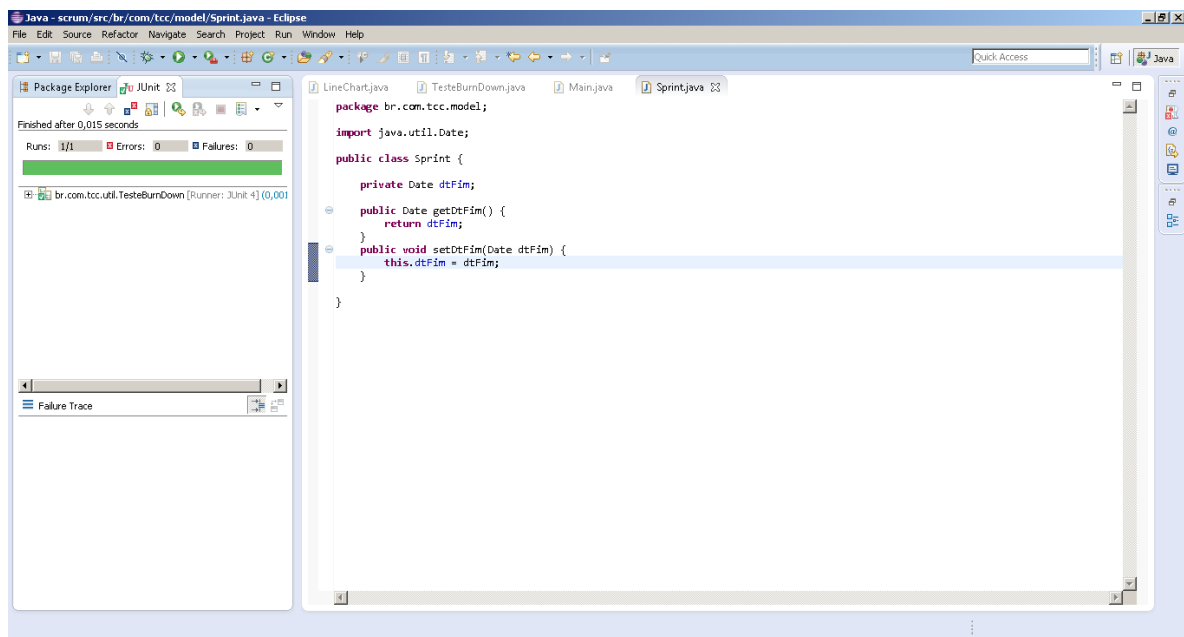
        Sprint sprint = new Sprint();

        @SuppressWarnings("deprecation")
        Date data = new Date(2014,12,01);
        sprint.setDtFim(data);
        assertEquals(data, sprint.getDtFim());

    }

}

```



### 3. Código refatorado

- Não foi necessário refatoração.

## OUTROS TESTES

### 1. Testes não compilam

```

import java.util.Date;

public class Sprint {

    private Date dtFim;

    public Date getDtFim() {
        return dtFim;
    }

    public void setDtFim(Date dtFim) {

```

```

        this.dtFim = dtFim;
    }
}

import java.text.ParseException;
import java.util.Date;
import junit.framework.TestCase;
import org.junit.Test;
import br.com.tcc.model.Sprint;

public class TesteBurnDown extends TestCase{

    @Test
    public void testaSprintDeveTer170HorasTotais() throws ParseException{

        Sprint sprint = Main.criaSprint();
        assertEquals(170.0, sprint.getTotalHoras());
    }

    @Test
    public void testaSprintDeveTer7Dias() throws ParseException{

        Sprint sprint = Main.criaSprint();
        assertEquals(7, sprint.getQtdeDias());
    }

    @Test
    public void testaDeveRetornarDataNula() throws ParseException{

        Date date = DataUtil.converteStringParaDate("");
        assertNull("mensagem", date);
    }

    @Test
    public void testaSprintDataInicioNula() throws ParseException {

        Sprint sprint = new Sprint();
        assertEquals(null, sprint.getDtInicio());
    }

    @Test
    public void testaSprintDataFim01Dezembro2014() throws ParseException {

        Sprint sprint = new Sprint();

        @SuppressWarnings("deprecation")
        Date data = new Date(2014,12,01);
        sprint.setDtFim(data);
        assertEquals(data, sprint.getDtFim());
    }
}

```

## 2. Testes falham

- Adicionando classe e métodos

```

import java.util.Date;
import java.util.List;

```

```

public class Sprint {

    private List<Estoria> estorias;
    private List<ItemHistorico> itensHistorico;
    private Date dtInicio;
    private Date dtFim;
    private int qtdeDias;
    private Double totalHoras;

    public Date getDtInicio() {
        return dtInicio = new Date();
    }
    public void setDtInicio(Date dtInicio) {
        this.dtInicio = dtInicio;
    }
    public Date getDtFim() {
        return dtFim;
    }
    public void setDtFim(Date dtFim) {
    }
    public List<Estoria> getEstorias() {
        return estorias;
    }
    public void setEstorias(List<Estoria> estorias) {
    }
    public Double getTotalHoras() {
        return totalHoras;
    }
    public void setTotalHoras(Double totalHoras) {
    }

    public int getQtdeDias() {
        return qtdeDias;
    }
    public void setQtdeDias(int qtdeDias) {
    }

    public List<ItemHistorico> getItensHistorico() {
        return itensHistorico;
    }
    public void setItensHistorico(List<ItemHistorico> itensHistorico) {
    }
}

```

---

```

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

```

```

public class DataUtil {

    private static final String FORMATO_DATA = "dd/MM/yyyy";

    public static Date converteStringParaDate(String data){

        Date date = new Date(System.currentTimeMillis());
        return date;
    }

    public static String dateToString(Date data){
        SimpleDateFormat formatBra = new SimpleDateFormat(FORMATO_DATA);
    }
}

```

```

        String result = formatBra.format(data);
        return (result);
    }

    public static Date adicionarDiasData(Date data, int dias){
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(data);
        calendar.add(Calendar.DAY_OF_MONTH, dias);
        return calendar.getTime();
    }
}

-----

import java.text.ParseException;
import java.util.ArrayList;
import java.util.List;

import br.com.tcc.bo.BurndownBO;
import br.com.tcc.bo.SprintBO;
import br.com.tcc.model.Estoria;
import br.com.tcc.model.ItemHistorico;
import br.com.tcc.model.Sprint;

public class Main {

    private static final int QTDE DIAS SPRINT = 7;
    private static final String DT_INICIO_SPRINT = "08/09/2014";
    private static SprintBO sprintBO = new SprintBO();

    public static void main(String[] args) throws ParseException {
        Sprint sprint = criaSprint();

        BurndownBO burndown = new BurndownBO();
        burndown.gerarBurndownHoras(sprint);
        burndown.gerarBurndownPontos(sprint);
    }

    /* Essa aplica  o n o possui comunica  o com banco de dados.
       Sendo assim, os m todos abaixo servem para criar os objetos com os
valores necess rios para o teste da gera  o dos gr ficos.
    */

    /**
     * Cria um objeto do tipo Sprint com valores fixos
     * @return
     * @throws ParseException
     */
    public static Sprint criaSprint() throws ParseException {
        Sprint sprint = new Sprint();

        return sprint;
    }

    -----

import java.text.ParseException;
import java.util.Date;
import junit.framework.TestCase;
import org.junit.Test;
import br.com.tcc.model.Sprint;

```

```

public class TesteBurnDown extends TestCase{

    @Test
    public void testaSprintDeveTer170HorasTotais() throws ParseException{

        Sprint sprint = Main.criaSprint();
        assertEquals(170.0, sprint.getTotalHoras());
    }

    @Test
    public void testaSprintDeveTer7Dias() throws ParseException{

        Sprint sprint = Main.criaSprint();
        assertEquals(7, sprint.getQtdeDias());
    }

    @Test
    public void testaDeveRetornarDataNula() throws ParseException{

        Date date = DataUtil.converteStringParaDate("");
        assertNull("mensagem", date);
    }

    @Test
    public void testaSprintDataInicioNula() throws ParseException {

        Sprint sprint = new Sprint();
        assertEquals(null, sprint.getDtInicio());
    }

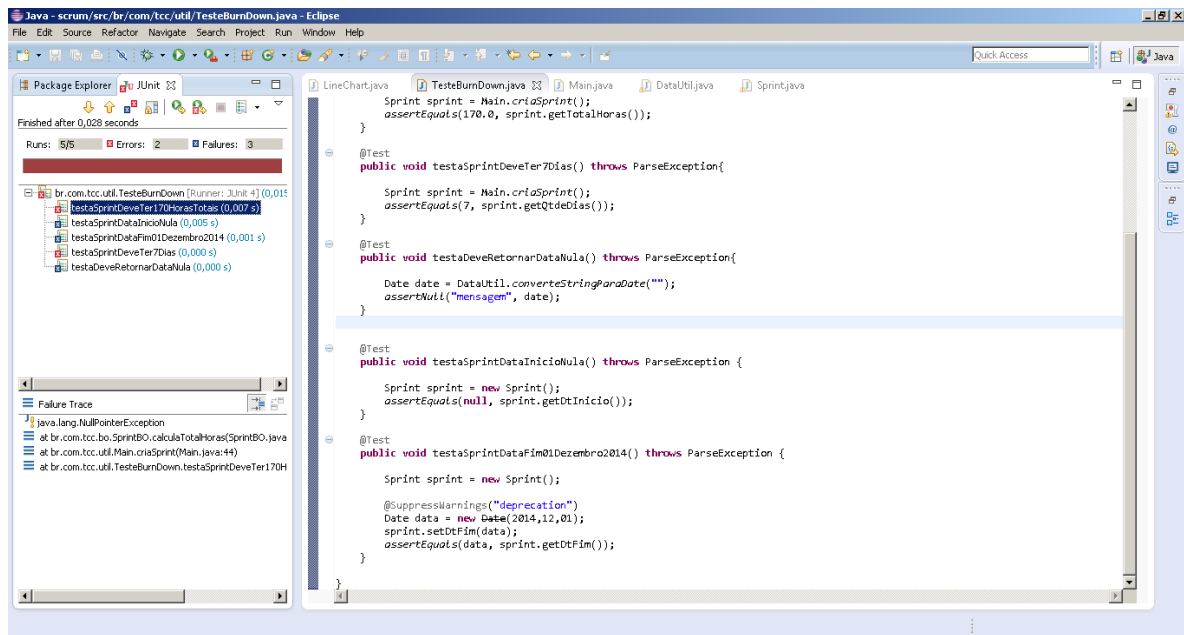
    @Test
    public void testaSprintDataFim01Dezembro2014() throws ParseException {

        Sprint sprint = new Sprint();

        @SuppressWarnings("deprecation")
        Date data = new Date(2014,12,01);
        sprint.setDtFim(data);
        assertEquals(data, sprint.getDtFim());
    }

}

```



### 3. Testes passam

