

TDD

Grasielle e Vitor.

Visão Geral:

Como já é sabido, vivemos em 2015 uma crise hídrica em nosso país, devido à escassez de chuvas nos últimos anos e à falta de planejamento de longo prazo das várias esferas de governo envolvidas. Neste cenário, políticas de multas e sobretaxas para desperdício de água e energia já foram implementadas, visto que nossa matriz energética se baseia fortemente em usinas hidrelétricas. As pessoas, portanto, estão se interessando crescentemente em descobrir maneiras de reduzir seus consumos de tais insumos e, assim, controlar seus gastos e minimizar o risco de agravar a crise. Tendo em vista este cenário, desenvolvemos um site no qual os interessados possam cadastrar seu perfil doméstico de consumo de energia, auxiliando estas pessoas a descobrirem como gastam energia e como podem fazer para economizar.

Esse projeto consiste na implementação de um dos casos de uso do Sistema Economy Energy, um sistema web para o cálculo do consumo de energia, utilizando a linguagem PHP e as classes utilizadas em outros casos de uso.

Objetivos:

- Realizar testes(TDD) nesse caso de uso
- Implementação do caso de uso Estimativa de Conta em PHP.

Descrição do Projeto:

Esse caso de uso(Estimativa de Conta), permite o usuário estimar o valor da conta, através dos KW consumidos ou estimar os KW consumidos através do valor da conta, mesmo não sendo o objetivo do sistema, o mesmo foi feito afim de melhorar a qualidade dos testes e garantir um resultado mais confiável.

Requisitos funcionais:

Fluxo Principal: Calcular Estimativa de Conta

- 1.O sistema exibe interface para o preenchimento do Preço da Conta ou consumo de KW .
- 2.O Usuário preenche os campos e clica em calcular.
- 3.O sistema calcula de acordo com a preferencia do usuário.
4. O caso de uso é encerrado.

IMPLEMENTACAO DE CASOS DE TESTE

É realizado o caso de teste afim de documentar os testes realizados no caso de uso para auxiliar nos testes futuros e assim como em possíveis mudanças no sistema. Foi realizado o TDD no caso de uso Estimativa de Conta, da classe Conta.php.

Objetivos

Os objetivos da implementação dos casos de testes são:

- Realizar o TDD
- Verificar se o sistema faz o solicitado
- Implementação das classes
- Documentar o sistema.

Casos de testes.

Foram realizados no caso de uso Estimar Conta o TDD, para isso foi feito o teste da seguinte forma:

1. Teste da implemtacao antes de escrever a função;
2. Funcao somente para executar o teste;
3. Realizacao do teste;
- 4.

O teste feito foi o teste unitário e o teste de aceitação:

O teste unitário testa apenas um caso de uso do sistema e o teste de aceitação testa uma funcionalidade do sistema.

Conclusao:

Ao realizar o TDD temos a certeza da qualidade do código a ser entregue ao cliente. Ele ajuda a corrigir e prevenir erros. O teste realizado no caso de uso citado acima, foi sucedido com sucesso.

CODIGO”:

```
<?php
require_once 'Conta.php';
class TestConta extends PHPUnit {
$PrecoKW Calculo = new $PrecoKW(0.92)
$Calculo.Consumo(100.00)
assertEquals(92.00, $Calculo.PrecoConta);
$Calculo.Consumo(218.00)
assertEquals(200.56, $Calculo.PrecoConta);
}
?>
//Classe construtora
<?php
require_once 'Conta.php';
```

```

class TestConta extends PHPUnit {
    $PrecoKW Calculo = new $PrecoKW(0.92)
    $Calculo.Consumo(100.00)
    assertEquals(92.00, $Calculo.PrecoConta);
    $Calculo.Consumo(218.00)
    assertEquals(200.56, $Calculo.PrecoConta);
    function __construct($PrecoKW,$PrecoConta,$ConsumoFinal)
    {
        this.$PrecoKW = $PrecoKW;
        this.$PrecoConta = $PrecoConta;
        this.$ConsumoFinal=$ConsumoFinal;
    }
}

```

?>

//refatora e torna generico

```

<?php
require_once 'Conta.php';
class TestConta extends PHPUnit {
    $PrecoKW Calculo = new $PrecoKW(0.92)
    $Calculo.Consumo(100.00)
    assertEquals(92.00, $Calculo.PrecoConta);
    $Calculo.Consumo(218.00)
    assertEquals(200.56, $Calculo.PrecoConta);

    public function Calculo($PrecoKW,$PrecoConta,$ConsumoFinal){
        $PrecoConta = $Consumo * $PrecoKW;
        return $PrecoConta;
    }
}

```

?>

// fim do primeiro ciclo

```

<?php
require_once 'Conta.php';
class TestConta extends PHPUnit {
    $PrecoKW Calculo = new $PrecoKW(0.92)
    $Calculo.PrecoConta(200.56)
    assertEquals(218.00, $Calculo.Consumo);
}
?>

```

//Classe construtora

```

<?php
require_once 'Conta.php';
class TestConta extends PHPUnit {
    $PrecoKW Calculo = new $PrecoKW(0.92)
    $Calculo.PrecoConta(200.56)
    assertEquals(218.00, $Calculo.Consumo);
}

```

```
function __construct($PrecoKW,$PrecoConta,$ConsumoFinal)
{
    this.$PrecoKW = $PrecoKW;
    this.$PrecoConta = $PrecoConta;
    this.$ConsumoFinal=$ConsumoFinal;
}
}
```

?>

//refatora e torna genérico

```
<?php
require_once 'Conta.php';
class TestConta extends PHPUnit {
$PrecoKW Calculo = new $PrecoKW(0.92)
$Calculo.PrecoConta(200.56)
assertEquals(218.00, $Calculo.Consumo);

public function Calculo($PrecoKW,$PrecoConta,$ConsumoFinal){
    $ConsumoFinal = $PrecoConta / $PrecoKW;
    return $ConsumoFinal;
}
}
```

?>