

## CASO DE USO

- O sistema mandara o relatório de consumo sobre economia de energia por e-mail para o usuário, com o cálculo mensal ou anual.
- Cada usuário se inscreve em uma de duas opções: calculo mensal ou anual.
- Antes de cada envio é preciso perceber se o usuário não se descadastrou da lista.

## Primeiro Teste

```
<?php
```

```
class RelatorioEmailTest extends \PHPUnit_Framework_TestCase
{
    public function testDeveriaRetornarGruposVaziosSeRecebeuNenhumEmail()
    {
        $expected = [
            'mensal' => [],
            'anual' => []
        ];
        $emails = array();

        $service = new Relatorio;

        $groups = $service->sendAll($emails);

        $this->assertEquals($expected, $groups);
    }
}
```

**Fatal error: Class 'Relatorio' not found in /meudiretorio/news.php on line 13**

```
<?php
```

```
class Relatorio
{
}
}
```

**Fatal error: Call to undefined method Relatorio::sendAll() in /meudiretorio/news.php on line 20**

```
<?php
```

```
class Relatorio
{
    public function sendAll($users)
    {
        $groups = array(
            'mensal' => [],
            'Anual' => []
        );

        return $groups;
    }
}
```

**OK (1 test, 1 assertion)**

**Segundo Teste**

```
<?php
```

```
class RelatorioEmailTest extends \PHPUnit_Framework_TestCase
{
    // primeiro teste
    // ...

    public function testDeveriaTerUmEmailNoGrupoDeMensal()
    {
        // relatório do tipo mensal
        $email = "mcezar27@gmail.com";
        $user = new User;
        $user->setEmail($email);
        $user->setNews(User::NEWS_TECHNOLOGY);

        // valor esperado diz que um email de relatório mensal foi enviado para
        mcezar27@gmail.com

        $expected = [
            'mensal' => [$email],
            'anual' => []
        ];

        // passaremos uma coleção com um usuário
        $emails = array($user);

        $service = new relatorio;

        $groups = $service->sendAll($emails);
        $this->assertEquals($expected, $groups);
    }
}
```

## **FAILURES!**

**Tests: 2, Assertions: 2, Failures: 1.**

```
class Relatorio
```

```

{
    public function sendAll($users)
    {
        $groups = array(
            'mensal' => [],
            'anual' => []
        );
        foreach ($users as $user) {
            $groups['mensal'][] = $user->getEmail();
        }

        return $groups;
    }
}

```

**OK (2 tests, 2 assertions)**

## Refatorando código

```

class Relatorio
{
    public function sendAll($users)
    {
        $groups = array(
            'mensal' => [],
            'anual' => []
        );

        foreach ($users as $user) {
            // apenas alteramos a forma como se descobre em que tipo de relatório o
            // usuário está.

            $key = ($user->getRelato() == User::NEWS_RELATORIOS) ? 'mensal' :
            'anual';

            $groups[$key][] = $user->getEmail();
        }

        return $groups;
    }
}

```

**OK (2 tests, 2 assertions)**