Código limpo guiado a testes

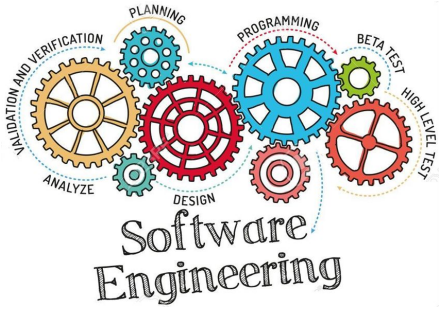
Como ser um(a) desenvolvedor(a) profissional de verdade

Durval Pereira

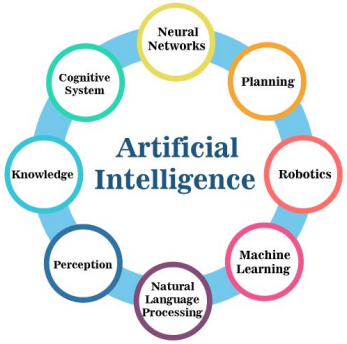
@durvalpcn
@durvalpereira

Hoje é sobre vocês, então sobre mim serei rápido:

+

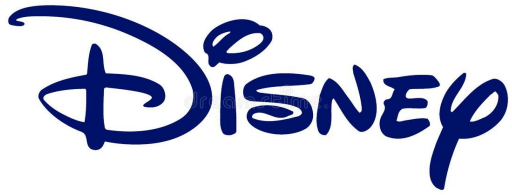


+



=


DUNKIN'
DONUTS


Disney


DANONE
ONE PLANET. ONE HEALTH


BlackBerry


CERVEJARIA
ambev


Itaú


TOYOTA


SAMSUNG


EPSON
EXCEED YOUR VISION


NFL


bradesco


BANCO
DO BRASIL


PEUGEOT

~100 devs mentorados

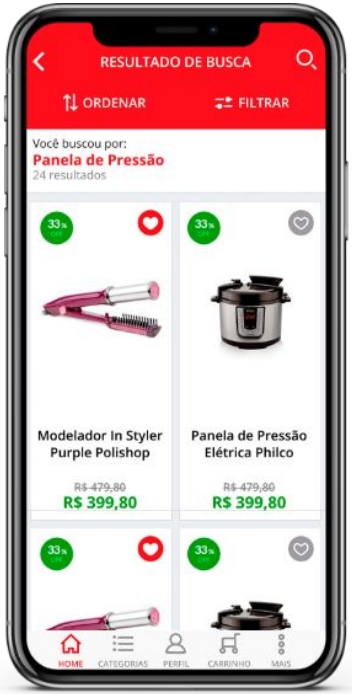
noumena

+

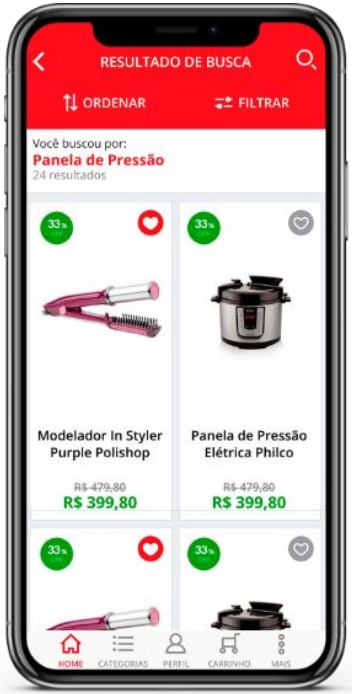
WORKANA

O que é ser um profissional?

Imagine o seguinte cenário...

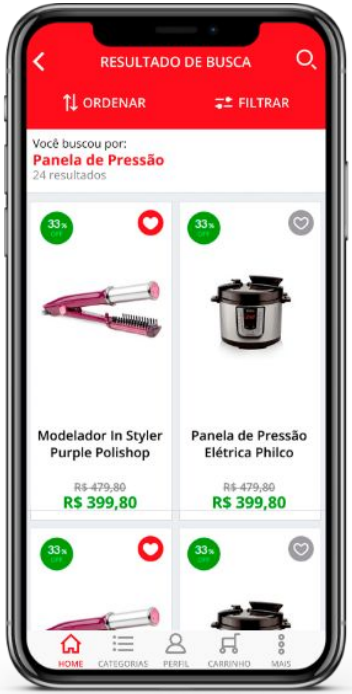


Imagine o seguinte cenário...



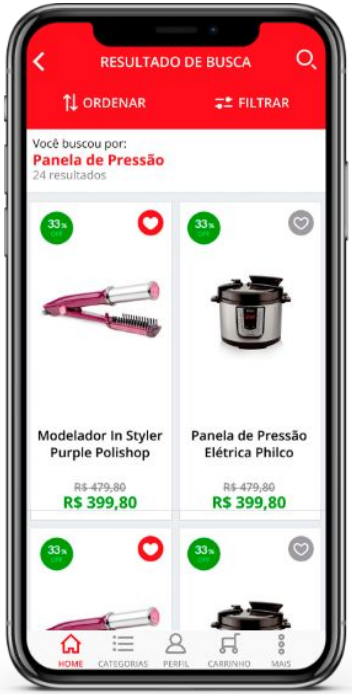
1. Cliente requisita sistema de cupons

Imagine o seguinte cenário...



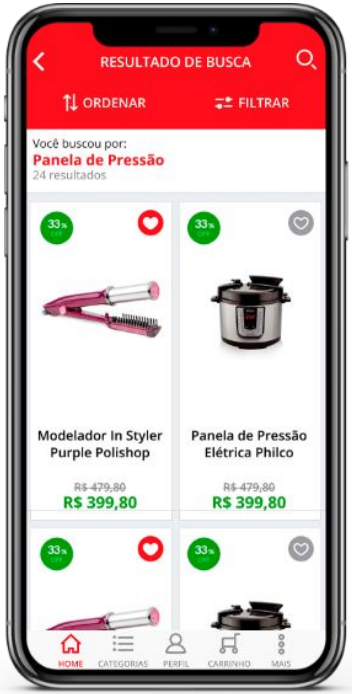
1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.000,00

Imagine o seguinte cenário...



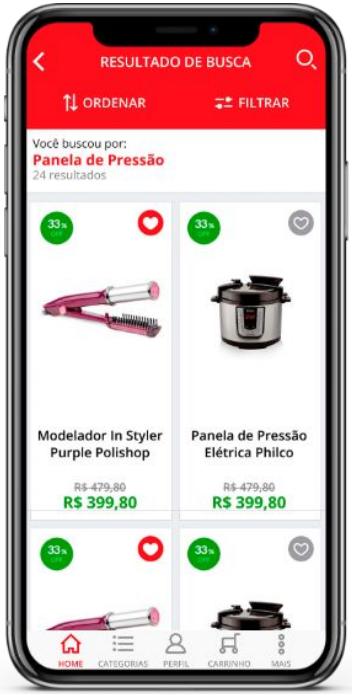
1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.000,00
3. Cliente aprova (app já está no ar)

Imagine o seguinte cenário...



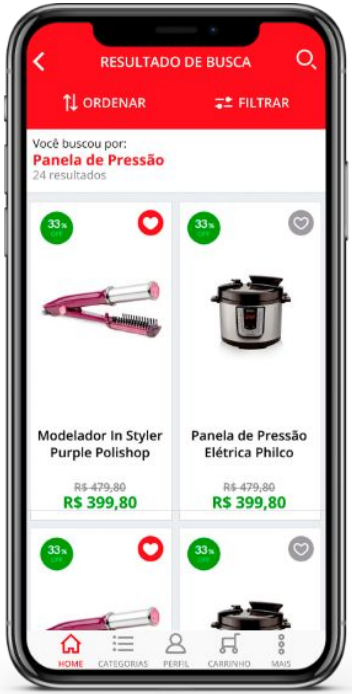
1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.000,00
3. Cliente aprova (app já está no ar)
4. Ao finalizar, você lança a atualização

Imagine o seguinte cenário...



1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.000,00
3. Cliente aprova (app já está no ar)
4. Ao finalizar, você lança a atualização
5. Algo passou despercebido, 12h OFF

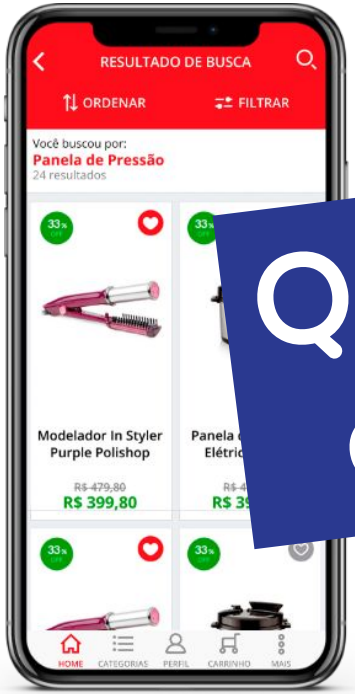
Imagine o seguinte cenário...



1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.000,00
3. Cliente aprova (app já está no ar)
4. Ao finalizar, você lança a atualização
5. Algo passou despercebido, 12h OFF

CLIENTE PERDEU R$ 1.200,00 EM VENDAS (!)

Imagine o seguinte cenário...



1. Cliente requisita sistema de cupons
2. Você cobra R$ 4.0[illegible]
3. [illegible]

Quanto um profissional deve receber ao fim?

...ão

...FF

...ERDEU R$ 1.200,00 EM VENDAS (!)

R$ 2.800,00

O **profissional é responsável** pelos seus erros.

PRINCIPALMENTE SE FOREM ERROS "IDIOTAS" :-)

REGRA #1

PARA SE TORNAR UM
CODIFICADOR LIMPO

Não cause danos ao funcionamento.

"À medida que amadurece em sua profissão, **sua taxa de erro deve diminuir** rapidamente em direção à assíntota de zero.

Ela **jamais chegará a zero**, mas é sua responsabilidade chegar ao mais próximo possível."


$\frac{1}{x}$
5
4
3
2
1
0
-1
-2
-3
-4
-5
-5
-4
-3
-2
-1
0
1
2
3
4
5

Voltemos então para o app do cliente e vamos codar a feature de cupom no back-end

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cart extends Model
{
    use HasFactory;

    /**
     * The attributes that can be mass assigned
     */
    protected $fillable = [
        'total_value',
        'discount',
    ];

    public function applyCoupon(Coupon $coupon): bool
    {
        $this->discount = $this->total_value * $coupon->percentage / 100;
        return true;
    }
}
```

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cart extends Model
{
    use HasFactory;

    /**
     * The attributes that can be mass assigned
     */
    protected $fillable = [
        'total_value',
        'discount',
    ];

    /**
     * Apply a given Coupon to a Cart, generating Discount
     */
    public function applyCoupon(Coupon $coupon): bool
    {
        $this->discount = $this->total_value * $coupon->percentage / 100;
        return true;
    }
}
```

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cart extends Model
{
    use HasFactory;

    /**
     * The attributes that can be mass assigned
     */
    protected $fillable = [
        'total_value',
        'discount',
    ];

    /**
     * Apply a given Coupon to a Cart, generating Discount
     *
     * @param Coupon $coupon A coupon object to be applied
     * @return bool
     */
    public function applyCoupon(Coupon $coupon): bool
    {
        $this->discount = $this->total_value * $coupon->percentage / 100;
        return true;
    }
}
```

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Cart extends Model
{
    use HasFactory;

    /**
     * The attributes that can be mass assigned
     */
    protected $fillable = [
        'total_value',
        'discount',
        'payment_value',
    ];

    /**
     * Apply a given Coupon to a Cart, generating Discount
     *
     * @param Coupon $coupon A coupon object to be applied
     * @return bool
     */
    public function applyCoupon(Coupon $coupon): bool
    {
        $this->discount = $this->total_value * $coupon->percentage / 100;
        $this->payment_value = $this->total_value - $this->discount;
        return true;
    }
}
```

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

TESTEI NA MÃO E FOI:

$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => 20]);

vai salvar um desconto de R$ 40,00, perfeito.

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

TESTEI NA MÃO E FOI:

$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => 20]);

vai salvar um desconto de R$ 40,00, perfeito.

TESTEI NA MÃO E FOI [2X]

$cart = new Cart([total_value => 300]);
$coupon = new Coupon([percentage => 35]);

vai salvar um desconto de R$ 105,00, show de bola!

Lorota do Dev 001.mp3

"O cliente/chefe tá no meu pé, **já testei o bastante**, vou lançar isso para ficar livre logo e mover para outras coisas que tenho que fazer."

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

O QUE O CLIENTE CONSEGUIU (E O APP PERMITIU) FAZER:
GERAR UM CUPOM COM 110% DE DESCONTO

$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => 110]);

quanto cliente vai pagar numa compra de 200$ com cupom?

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

O QUE O CLIENTE CONSEGUIU (E O APP PERMITIU) FAZER:
GERAR UM CUPOM COM 110% DE DESCONTO

$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => 110]);

quanto cliente vai pagar numa compra de 200$ com cupom?

-20$

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

-200$

O QUE O CLIENTE CONSEGUIU (E O APP PERMITIU) FAZER:
GERAR UM CUPOM COM 200% DE DESCONTO

$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => 200]);

quanto cliente vai pagar numa compra de 200$ com cupom?

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->percentage > 100) {
        return false;
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->percentage > 100) {
        return false;
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

240$

O QUE O CLIENTE CONSEGUIU (E O APP PERMITIU) FAZER:
GERAR UM CUPOM COM -20% DE DESCONTO

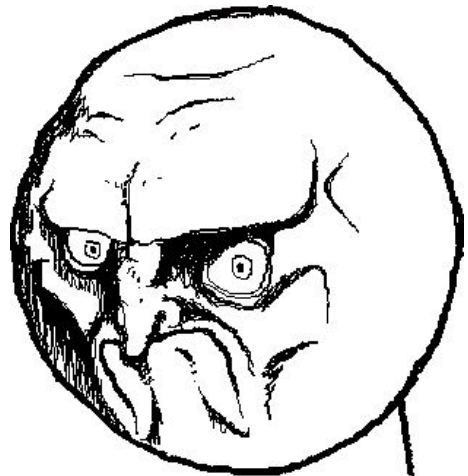
$cart = new Cart([total_value => 200]);
$coupon = new Coupon([percentage => -20]);

quanto cliente vai pagar numa compra de 200$ com cupom?

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        return false;
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        return false;
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        return false;
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

QUEM CODA NO FRONT



```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

```php
public function applyCoupon(Coupon $coupon): bool
{
    $this->discount = $this->total_value * $coupon->percentage / 100;
    return true;
}
```

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

Ser um profissional é bem mais satisfatório.

E pode ter certeza.. os "minutinhos" a mais vão lhe economizar horas - **e tempo é dinheiro.**

```
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
```

7 linhas de código que vão mudar sua vida lá na frente.

Ou a vida de outro programador.

```php
if ($coupon->is_expired) {
    throw new Exception("this coupon " . $coupon->code . " is expired.");
}

if ($coupon->percentage < 0 || $coupon->percentage > 100) {
    throw new Exception("this coupon doesn't have a valid value.");
}
```

7 linhas de código que vão lhe evitar dores de cabeça.

E em seu cliente/chefe também.

Lorota do Dev 002.mp3

"Preciso produzir.
Testes eu faço lá na frente - ou, se tiver equipe de QA, eles fazem e me dizem se achar algo."

REGRA #2

PARA SE TORNAR UM
CODIFICADOR LIMPO

QA não deve encontrar nada.


WORKED FINE IN LOCAL
QA PROBLEM NOW
memegenerator.net


BRACE YOURSELF
THE TESTERS ARE TESTING
makeameme.org


YOU ARE A THIEF
AND YOU KNOW IT
memegenerator.net

Quanto mais erros o QA achar, menor a confiança da empresa na equipe de desenvolvimento.

Não diminua a confiança propositalmente.

Se você não tem certeza, provavelmente não funciona.

REGRA #3

PARA SE TORNAR UM
CODIFICADOR LIMPO

Você precisa SABER que funciona.

Mas como vou saber?

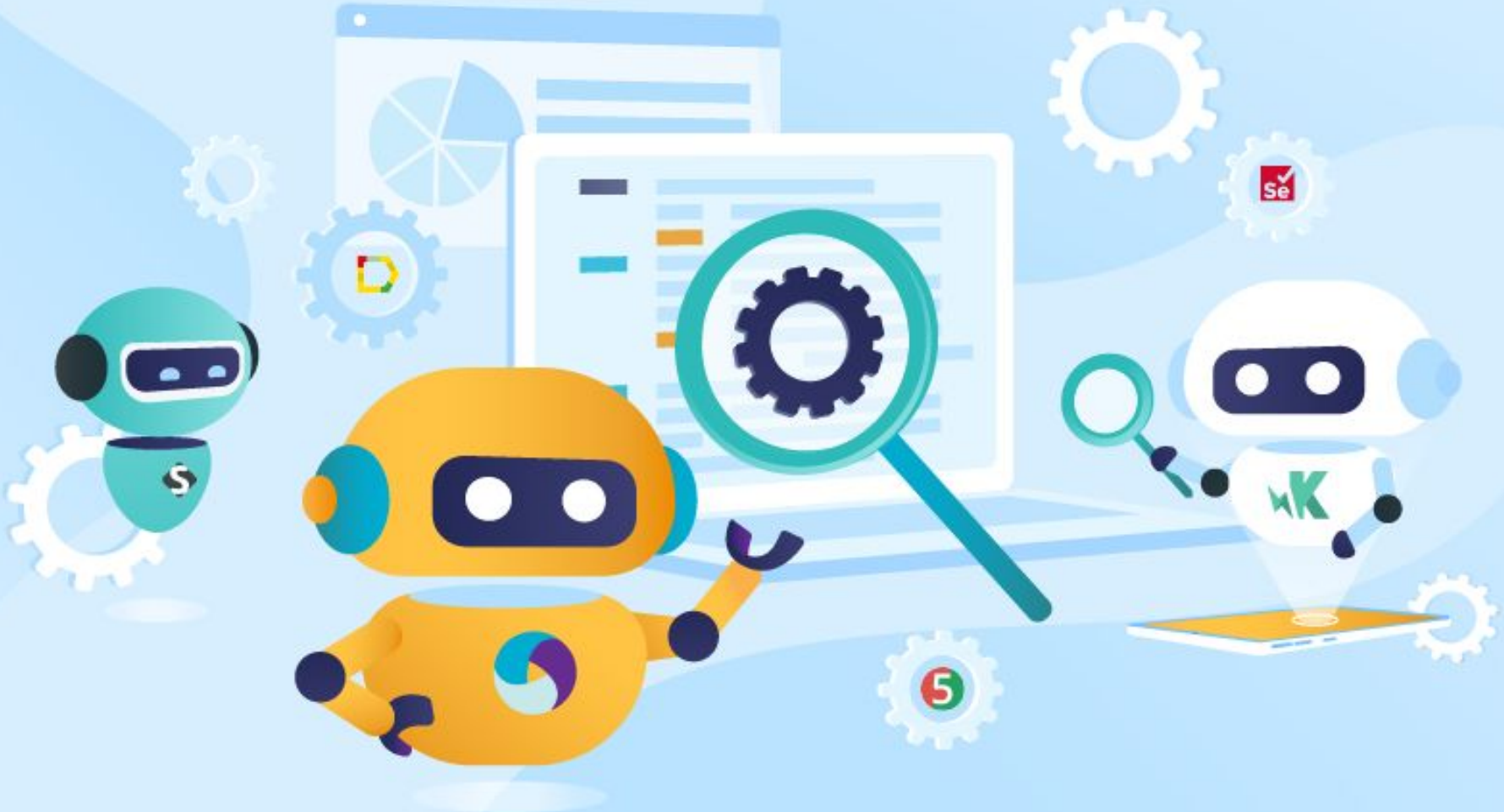
Mas como vou saber?

Teste o código. Teste-o de novo.
Teste novamente. Mais uma vez.

Mas como vou saber?

Teste o código. Teste-o de novo.
Teste novamente. Mais uma vez.

Está entendendo para onde estamos indo?



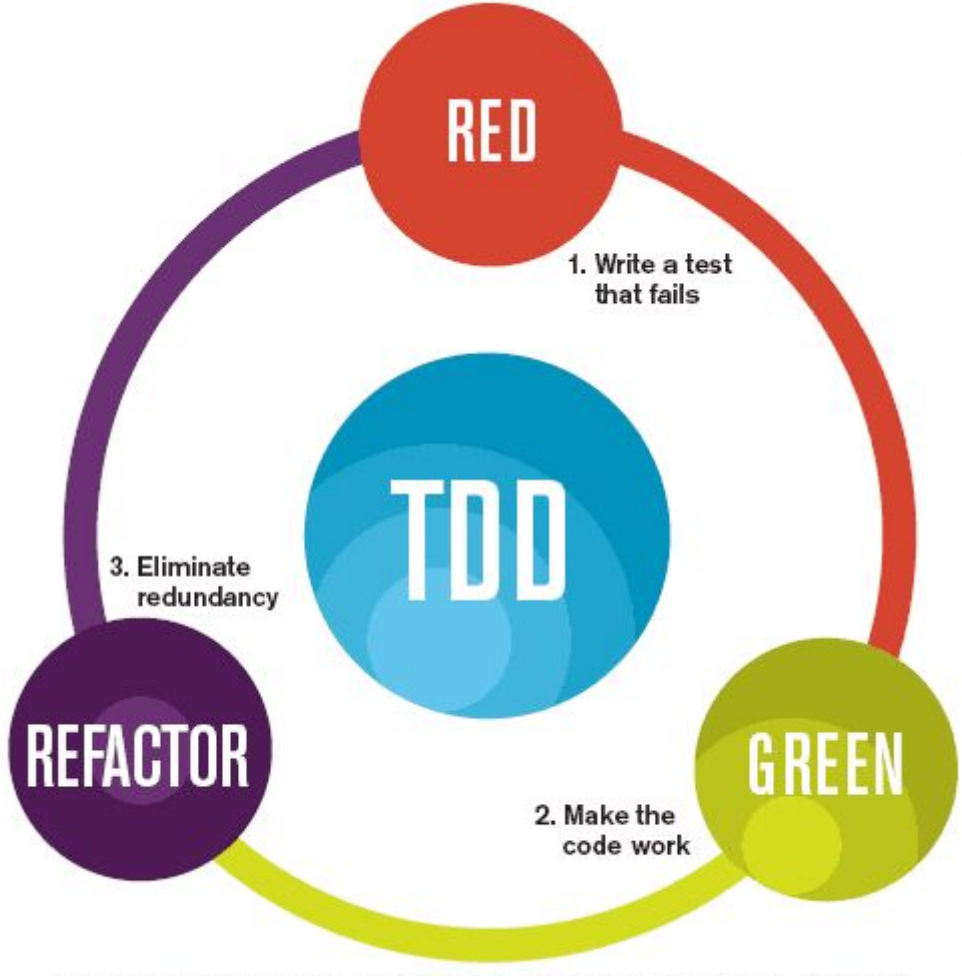
REGRA #4

PARA SE TORNAR UM
CODIFICADOR LIMPO

Utilize Testes Automatizados.

Automation Testing

- Fast
- Reliable
- Reusable
- Improves Accuracy
- Saves time and money
- Reduces Human-generated error
- Supports the execution of repeated test cases



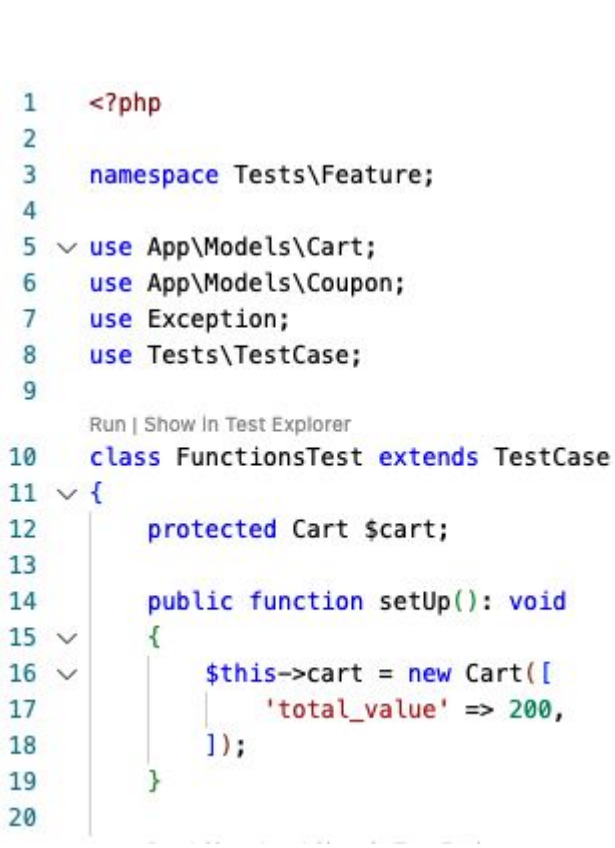
The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Voltemos **de novo** para o app do cliente e vejamos como desenvolver **guiando-se a testes**

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

```php
<?php

namespace Tests\Feature;

use App\Models\Cart;
use App\Models\Coupon;
use Exception;
use Tests\TestCase;

Run | Show In Test Explorer
class FunctionsTest extends TestCase
{
    protected Cart $cart;

    public function setUp(): void
    {
        $this->cart = new Cart([
            'total_value' => 200,
        ]);
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_works_properly_for_valid_coupons(): void
    {
        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => 10,
        ]);

        $this->assertTrue($this->cart->applyCoupon($coupon));
        $this->assertEquals(
            $this->cart->discount,
            ($this->cart->total_value * ($coupon->percentage / 100)),
        );
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_throws_exception_when_coupon_has_invalid_value(): void
    {
        $this->expectException(Exception::class);

        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => -10,
        ]);

        $this->cart->applyCoupon($coupon);
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_throws_exception_when_coupon_is_expired(): void
    {
        $this->expectException(Exception::class);

        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => 20,
            'is_expired' => true,
        ]);

        $this->cart->applyCoupon($coupon);
    }
}
```

```php
<?php

namespace Tests\Feature;

use App\Models\Cart;
use App\Models\Coupon;
use Exception;
use Tests\TestCase;

Run | Show In Test Explorer
class FunctionsTest extends TestCase
{
    protected Cart $cart;

    public function setUp(): void
    {
        $this->cart = new Cart([
            'total_value' => 200,
        ]);
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_works_properly_for_valid_coupons(): void
    {
        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => 10,
        ]);

        $this->assertTrue($this->cart->applyCoupon($coupon));
        $this->assertEquals(
            $this->cart->discount,
            ($this->cart->total_value * ($coupon->percentage / 100)),
        );
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_throws_exception_when_coupon_has_invalid_value(): void
    {
        $this->expectException(Exception::class);

        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => -10,
        ]);

        $this->cart->applyCoupon($coupon);
    }

    Run | Show Log | Show In Test Explorer
    public function test_applyCoupon_throws_exception_when_coupon_is_expired(): void
    {
        $this->expectException(Exception::class);

        $coupon = new Coupon([
            'code' => 'TEST10%OFF',
            'percentage' => 20,
            'is_expired' => true,
        ]);

        $this->cart->applyCoupon($coupon);
    }
}
```



```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_works_properly_for_valid_coupons(): void
{
    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 10,
    ]);

    $this->assertTrue($this->cart->applyCoupon($coupon));
    $this->assertEquals(
        $this->cart->discount,
        ($this->cart->total_value * ($coupon->percentage / 100)),
    );
}
```

```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_works_properly_for_valid_coupons(): void
{
    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 10,
    ]);

    $this->assertTrue($this->cart->applyCoupon($coupon));
    $this->assertEquals(
        $this->cart->discount,
        ($this->cart->total_value * ($coupon->percentage / 100)),
    );
}
```

```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_throws_exception_when_coupon_has_invalid_value(): void
{
    $this->expectException(Exception::class);

    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => -10,
    ]);

    $this->cart->applyCoupon($coupon);
}
```

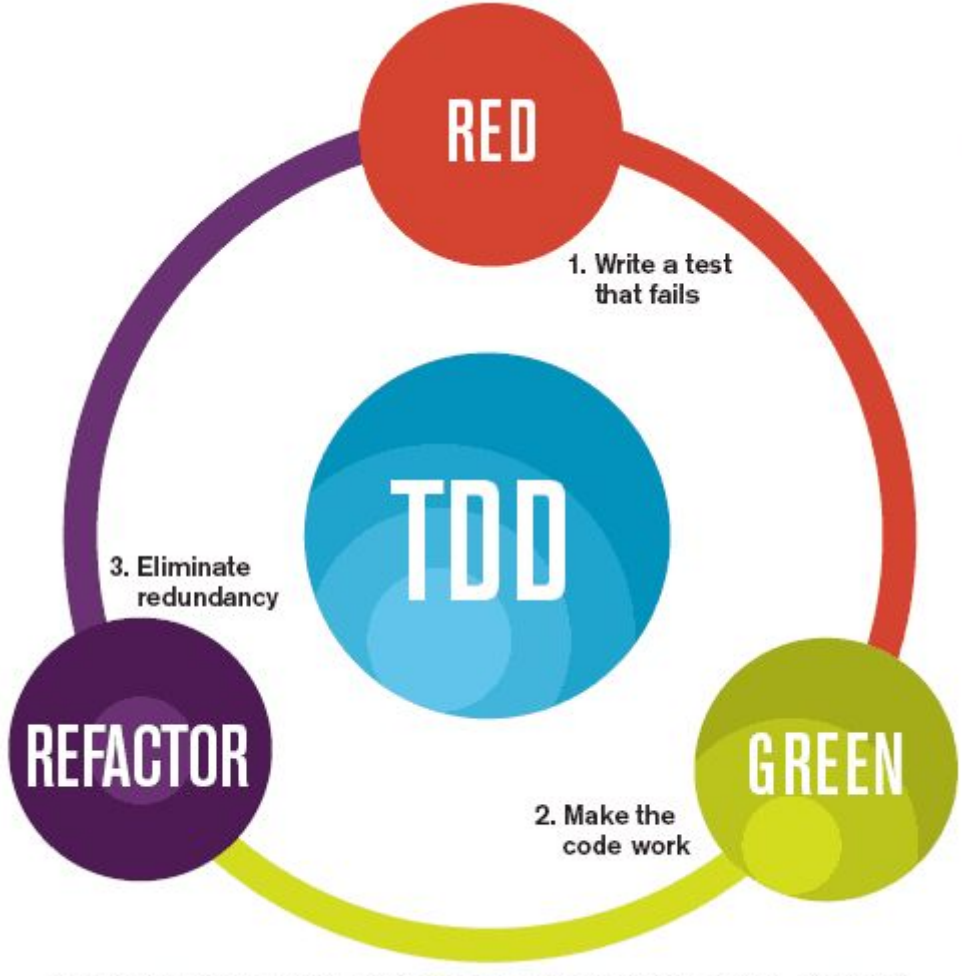
```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_works_properly_for_valid_coupons(): void
{
    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 10,
    ]);

    $this->assertTrue($this->cart->applyCoupon($coupon));
    $this->assertEquals(
        $this->cart->discount,
        ($this->cart->total_value * ($coupon->percentage / 100)),
    );
}
```

```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_throws_exception_when_coupon_is_expired(): void
{
    $this->expectException(Exception::class);

    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 20,
        'is_expired' => true,
    ]);

    $this->cart->applyCoupon($coupon);
}
```

```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_throws_exception_when_coupon_has_invalid_value(): void
{
    $this->expectException(Exception::class);

    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => -10,
    ]);

    $this->cart->applyCoupon($coupon);
}
```

```
durvalpereira@MacBook-Pro codigo-limpo % php artisan test

  PASS  Tests\Feature\FunctionsTest
  ✓ apply coupon works properly for valid coupons
  ✓ apply coupon throws exception when coupon has invalid value
  ✓ apply coupon throws exception when coupon is expired

  Tests:  3 passed
  Time:   0.02s
```

Vamos adicionar uma nova função no app, mas desta vez **vamos utilizar TDD** para desenvolvê-la



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Run | Show Log | Show In Test Explorer

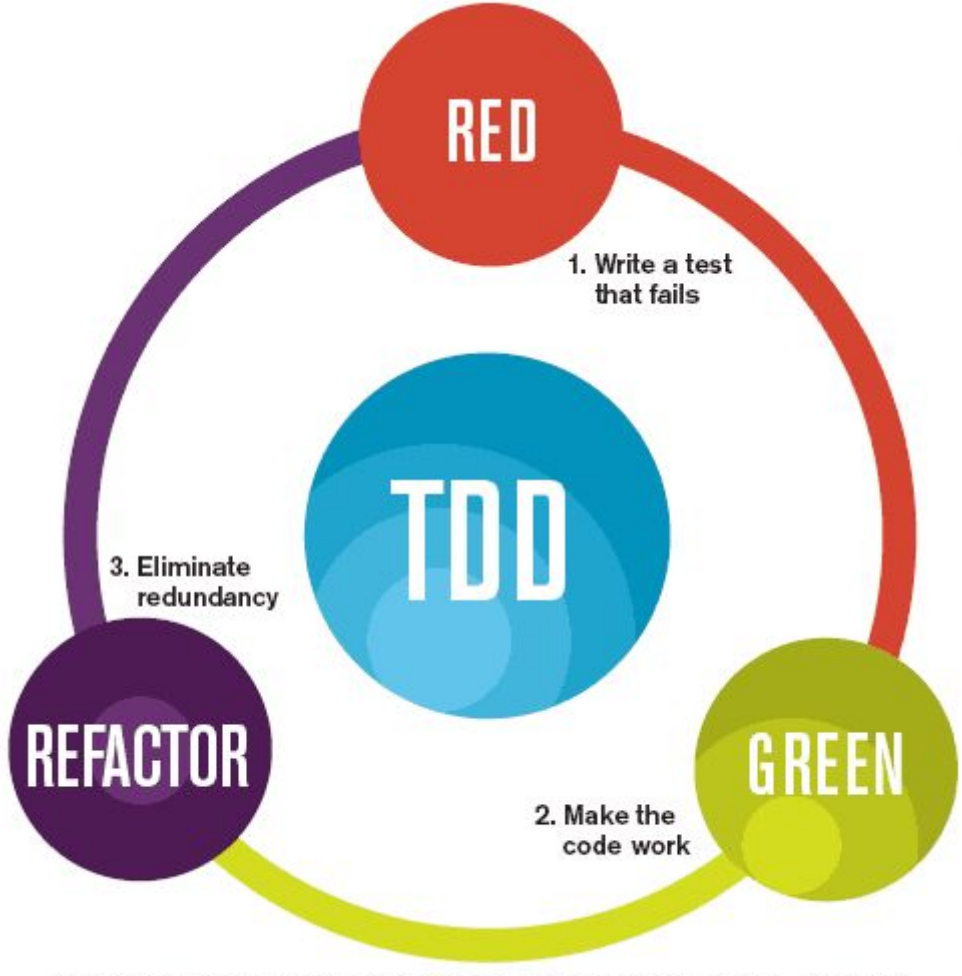
```php
public function test_applyCoupon_throws_exception_when_regions_are_different(): void
{
    $this->expectException(Exception::class);

    $this->cart->region = 'NORDESTE';

    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 20,
        'region' => 'SUL',
    ]);

    $this->cart->applyCoupon($coupon);
}
```

RED...

RED OK!

```
Run | Show Log | Show in Test Explorer
public function test_applyCoupon_throws_exception_when_regions_are_different(): void
{
    $this->expectException(Exception::class);

    $this->cart->region = 'NORDESTE';

    $coupon = new Coupon([
        'code' => 'TEST10%OFF',
        'percentage' => 20,
        'region' => 'SUL',
    ]);

    $this->cart->applyCoupon($coupon);
}
```

```
durvalpereira@MacBook-Pro codigo-limpo % php artisan test

  FAIL  Tests\Feature\FunctionsTest
  ✓ apply coupon works properly for valid coupons
  ✓ apply coupon throws exception when coupon has invalid value
  ✓ apply coupon throws exception when coupon is expired
  × apply coupon throws exception when regions are different

  ---

  • Tests\Feature\FunctionsTest > apply coupon throws exception when regions are different
  Failed asserting that exception of type "Exception" is thrown.

  at vendor/phpunit/phpunit/phpunit:98
     94 | unset($options);
     95 |
     96 | require PHPUNIT_COMPOSER_INSTALL;
     97 |
  →  98 | PHPUnit\TextUI\Command::main();
     99 |


  Tests:  1 failed, 3 passed
  Time:   0.02s
```



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

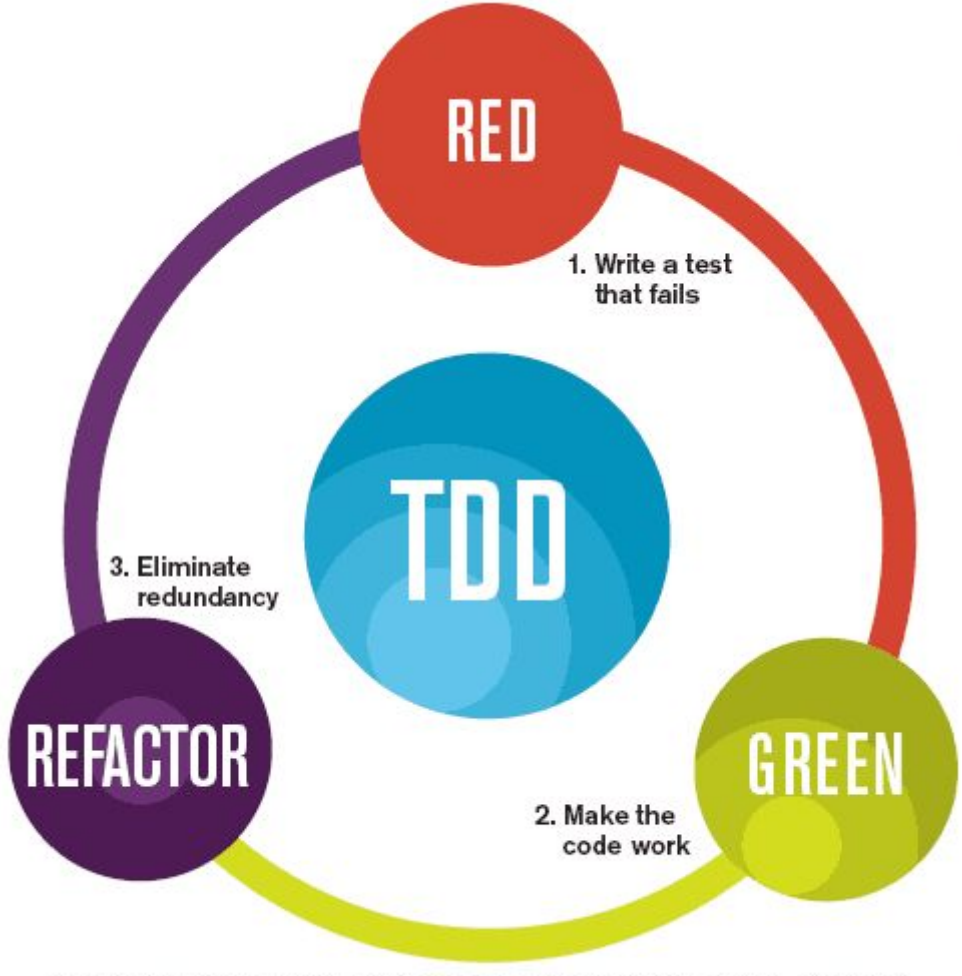
GREEN...

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->region != $this->region) {
        throw new Exception("this coupon is from a different region");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

GREEN...

GREEN OK!

```
durvalpereira@MacBook-Pro codigo-limpo % php artisan test

  PASS  Tests\Feature\FunctionsTest
  ✓ apply coupon works properly for valid coupons
  ✓ apply coupon throws exception when coupon has invalid value
  ✓ apply coupon throws exception when coupon is expired
  ✓ apply coupon throws exception when regions are different

  Tests:  4 passed
  Time:   0.02s
```



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

REGRA #5

PARA SE TORNAR UM
CODIFICADOR LIMPO

Codifique guiando-se pelos testes.

Vantagem de um Codificador Limpo guiado a Testes

Vantagem de um Codificador Limpo guiado a Testes

Quando você é um codificador limpo, as etapas 2 e 3 (GREEN & REFACTOR) do processo de TDD **se mesclam em uma só** nos casos menos complexos.

```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 * @throws Exception
 */
public function applyCoupon(Coupon $coupon): bool
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->region != $this->region) {
        throw new Exception("this coupon is from a different region");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}
```

REFACTOR (?)

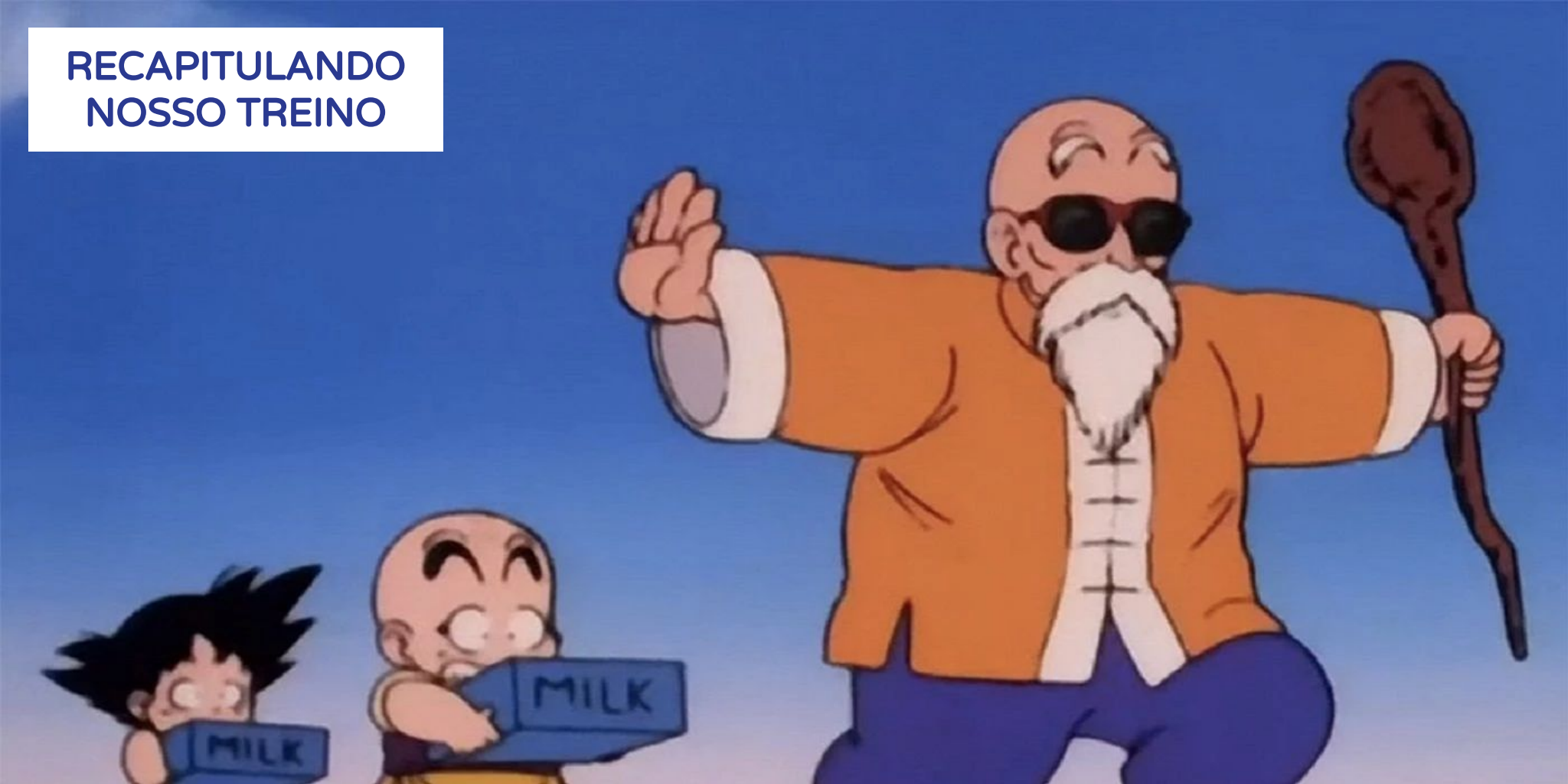
```php
/**
 * Apply a given Coupon to a Cart, generating Discount
 *
 * @param Coupon $coupon A coupon object to be applied
 * @return bool
 */
public function applyCoupon(Coupon $coupon): bool
{
    $this->validateCoupon($coupon);

    $this->discount = $this->total_value * ($coupon->percentage / 100);
    $this->payment_value = $this->total_value - $this->discount;

    return true;
}

/**
 * Checks if coupon is valid for all conditionals
 *
 * @param Coupon $coupon A coupon object to be validated
 * @return void
 * @throws Exception
 */
public function validateCoupon(Coupon $coupon): void
{
    if ($coupon->is_expired) {
        throw new Exception("this coupon " . $coupon->code . " is expired.");
    }

    if ($coupon->region != $this->region) {
        throw new Exception("this coupon is from a different region");
    }

    if ($coupon->percentage < 0 || $coupon->percentage > 100) {
        throw new Exception("this coupon doesn't have a valid value.");
    }
}
```

REFACTOR...

REFACTOR OK!

```
durvalpereira@MacBook-Pro codigo-limpo % php artisan test

  PASS  Tests\Feature\FunctionsTest
  ✓ apply coupon works properly for valid coupons
  ✓ apply coupon throws exception when coupon has invalid value
  ✓ apply coupon throws exception when coupon is expired
  ✓ apply coupon throws exception when regions are different

  Tests:  4 passed
  Time:   0.02s
```

RECAPITULANDO NOSSO TREINO



REGRA #1

PARA SE TORNAR UM
CODIFICADOR LIMPO

Não cause danos ao funcionamento.

REGRA #2

PARA SE TORNAR UM
CODIFICADOR LIMPO

QA não deve encontrar nada.

REGRA #3

PARA SE TORNAR UM
CODIFICADOR LIMPO

Você precisa SABER que funciona.

REGRA #4

PARA SE TORNAR UM
CODIFICADOR LIMPO

Utilize testes automatizados.

REGRA #5

PARA SE TORNAR UM
CODIFICADOR LIMPO

Codifique guiando-se pelos testes.


KAME
HOUSE

Obrigado :)

Espero que tenha aberto seu olho para que você possa se tornar um(a) desenvolvedor(a) profissional de verdade!

Durval Pereira

@durvalpcn
@durvalpereira