

Black Rainbow

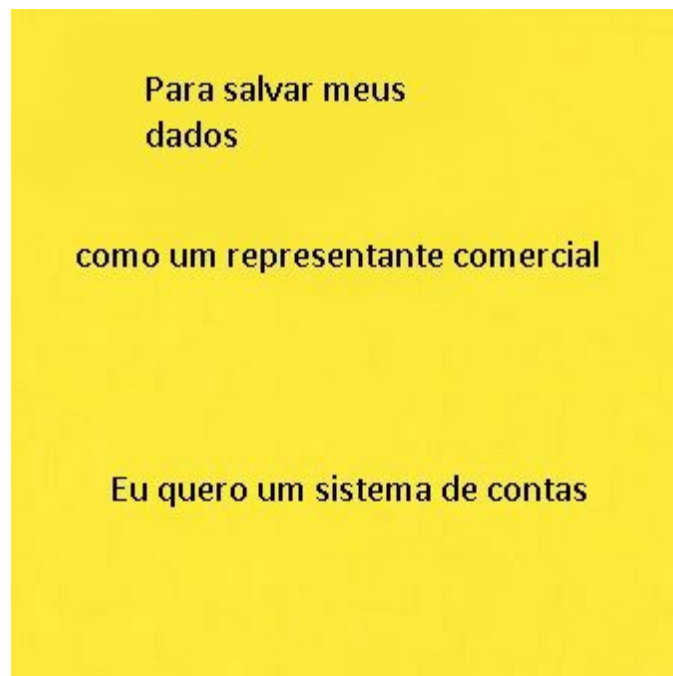


Postmortem -Iteração 2

Nesta segunda iteração a equipe de desenvolvimento se uniu em prol do desenvolvimento de um sistema de contas e de um sistema de pedidos, que foram requisitados pelo cliente. Ilustramos cada visão sobre nosso sistema com imagens e textos.

Sistema de Contas

Historia do usuário:



Implementação:

O processo de implementação seguiu a seguinte linha
Sistema de registrar > Aplicação SSL e uso do Puma webserv > Sistema de Log in >
Sistema de Log out > Sistema de Sessão > Sistema de lembrar do usuário

O parte de registrar foi desenvolvida primeiro através de modelar o usuário em questão de banco e model, dando suas restrições e requisitos de como por exemplo quem vai ser `primary_key`, os máximos e mínimos de tamanho de strings em outros campos, além de garantir que certos campos não serão nulos, logo após a lógica de iteração de view e controller foram feitas. Foi usado o bcrypt para criptografar as senhas.

Foi aplicado o uso de SSL para garantir segurança no tráfego de dados entre o usuário e a aplicação do webserver puma para melhor desempenho.

Logo após foi feito o sistema de log in e log out, e ao mesmo tempo o de sessão, fazendo grande uso dos helpers para definir funções para auxiliar os controladores.

Então com o sistema pronto foi feito um sistema adicional de lembrar usuários caso eles queiram, funciona como um checkbox no momento do log in, com o uso de cookies e tokens, tudo de modo criptografado para garantir o máximo de segurança possível.

TDD:

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/controllers/sessions_controller_test.rb
Started

 1/1: [=====] 100% Time: 00:00:00, Time: 00:00:00

Finished in 0.15660s
1 tests, 1 assertions, 0 failures, 0 errors, 0 skips
imlbf:~/workspace (TestSet) $
```

Teste unitário do Controlador de sessions, o qual garante o mantimento de uma sessão de um usuário.

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/models/user_test.rb
Started

 12/12: [=====] 100% Time: 00:00:00, Time: 00:00:00

Finished in 0.63622s
12 tests, 20 assertions, 0 failures, 0 errors, 0 skips
imlbf:~/workspace (TestSet) $
```

Teste unitário do Model dos usuários, Test o qual garante que os requisitos serão seguidos, tais como tamanho máximos e mínimos de campos a serem salvos no banco de dados, que não pode ter valores em branco e quem deve ter valores únicos.

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/controllers/users_controller_test.rb
Started

1/1: [=====] 100% Time: 00:00:00, Time: 00:00:00
Finished in 0.18184s
1 tests, 1 assertions, 0 failures, 0 errors, 0 skips
imlbf:~/workspace (TestSet) $
```

Teste unitário do controlador dos usuários, que garante o que será enviado para a view e o que será recebido da view, tudo da maneira certa.

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/integration/users_signup_test.rb
Started

2/2: [=====] 100% Time: 00:00:00, Time: 00:00:00
Finished in 0.26839s
2 tests, 4 assertions, 0 failures, 0 errors, 0 skips
imlbf:~/workspace (TestSet) $
```

Teste de integração de registro de usuários, garante que os campos de registros façam a coisa certa com as informações certas e redirecione para o local certo.

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/integration/users_login_test.rb
Started

4/4: [=====] 100% Time: 00:00:00, Time: 00:00:00
Finished in 0.24945s
4 tests, 18 assertions, 0 failures, 0 errors, 0 skips
imlbf:~/workspace (TestSet) $
```

Teste de integração de login de usuário, garante que os campos estejam da maneira certa.

BDD:

```
imlbf:~/workspace (master) $ cucumber features/login.feature
Using the default profile...
Feature: Have a personal account

  Scenario: Login in Nobalk site # features/login.feature:3
    Given I am in the nobalk's sign in page # features/step_definitions/login_steps.rb:1
    Then I fill the email camp # features/step_definitions/login_steps.rb:5
    And I fill the senha camp # features/step_definitions/login_steps.rb:8
    When I press entrar # features/step_definitions/login_steps.rb:11
    Then I should see my homepage on the site # features/step_definitions/login_steps.rb:14

1 scenario (1 passed)
5 steps (5 passed)
0m0.108s
imlbf:~/workspace (master) $
```

Foi definido que um usuário que se encontra na página home do nobalk ao entrar com seus dados, fosse direcionado a sua página inicial.

Sistema de Pedidos

Historia do usuário:

Para organizar meus pedidos

como um representante comercial

Eu quero um sistema de pedidos

Implementação:

Foi desenvolvido um CRUD para pedidos, com seus atributos mínimos, sendo eles uma Quantidade de itens inteira e um valor total do pedido, sendo float, ele foi estruturado para que em iterações futuras se integre com outras funcionalidades, por exemplo, a de produtos.

Com o uso de scaffold criamos o create, read, update e delete da aplicação, logo depois estruturamos layout, views com uso do bootstrap, e depois adaptações no controller e model, para seguir a linha certa.

TDD:

```
imlbf:~/workspace (TestSet) $ rake test TEST=test/controllers/pedidos_controller_test.rb
Started

 7/7: [=====] 100% Time: 00:00:00, Time: 00:00:00

Finished in 0.22870s
7 tests, 13 assertions, 0 failures, 0 errors, 0 skips

imlbf:~/workspace (TestSet) $
```

Teste unitário do controlador dos pedidos, que garante o que sera enviado para a view e o que sera recebido da view, da maneira certa.

BDD:

```
i■lbf:~/workspace (master) $ cucumber features/pedidos.feature
Using the default profile...
Feature: Pedidos page

  Scenario:                                     # features/pedidos.feature:3
    Given I m in my homepage on nobalk        # features/step_definitions/pedidos_steps.rb:1
    Then I press the button pedidos            # features/step_definitions/pedidos_steps.rb:5
    Then I should see my pedidos list          # features/step_definitions/pedidos_steps.rb:10

1 scenario (1 passed)
3 steps (3 passed)
0m0.062s
i■lbf:~/workspace (master) $
```

Foi definido que um usuário que se encontrasse na sua pagina inicial ao clicar no boto pedidos fosse redirecionado a pagina com seus respectivos pedidos.

Avaliação dos membros:

Ian Fireman:Foi mais ativo durante o desenvolvimento.

Danilo Matheus:Auxiliou durante o desenvolvimento e durante o planejamento e gerenciamento.

Abhner Araujo: Foi mais ativo durante o planejamento e gerenciamento

Conclusões

- **A iteração serviu de aprendizado para entender como funcionam os teste no rails, Sendo testes unitários do rails, Cucumber e Capybara.**
- **A iteração também serviu para tornar os conhecimentos da equipe mais profundos em relação ao funcionamento de branches e forks**
- **Durante a iteração os integrantes tornaram-se mais entrosados e comprometidos.**
- **As metas foram alcançadas com sucesso,com um melhor gerenciamento da equipe,diferentemente da outras iterações.**