

PostMortem

1. Projeto

Descrição

1.0 Nome do Projeto

Zilla

1.1 Equipe

-Jhonattas Ferreira(jfs4@cin.ufpe.br)

-Matheus Marconi(mmfr@cin.ufpe.br)

1.2 Iteração

-Deploy

-2 Histórias de Usuário

1.3 Data de Inicio

25/04/2016

1.4 Data Final

09/05/2016

2. Performance

Principais realizações

2.0 O que deu certo?

Deploy

O deploy foi bem sucedido. A plataforma escolhida foi o Heroku. Precisamos mudar o nosso banco de dados de produção para o **Postgresql**. Não tivemos maiores dificuldades.

Página de Cadastro

A implementação do cadastro de usuário foi bem sucedida. O usuário insere as informações nos campos e caso estejam corretas, os dados são salvos no Banco de Dados. Um hash da senha do usuário é salvo no banco de dados, dessa forma garantimos uma maior segurança ao usuário.

Framework - Bootstrap

A implementação do Bootstrap foi bem sucedida. O layout que pretendemos utilizar já foi definido(mockups) e conseguimos definir algumas páginas.

Geração de Model - User

Geramos o modelo referente ao usuário. A princípio o modelo irá salvar, apenas, o nome, o e-mail e um hash da senha do usuário no banco de dados. Colocamos alguma restrições referente ao e-mail, ao nome e a senha. Todos os atributos não podem estar em branco. Todos os atributos tem restrições quanto ao seu tamanho. O e-mail é único e formatado.

Geração de Controller- User

Geramos o Controlador referente ao usuário. Definimos algumas ações do controlador afim de mostrar e cadastrar os usuários.

Geração do Controller - StaticPages

Geramos o controlador referente as páginas estáticas da nossa aplicação. A HomePage foi definida.

Testes Unitários

Realizamos alguns testes unitários referentes ao Modelo e aos Controladores do usuário.

2.1 O que funcionou bem?

Página do Usuário

A página do usuário foi preenchida com as informações do usuário em questão. Ficamos impedidos de preencher a página com outras informações, pois ainda não implementamos a funcionalidade de comprar novos tickets.

2.2 Que processos do projeto não funcionou bem?

API Login Google

Tentamos utilizar a API de login do google, porém não obtivemos sucesso, decidimos então criar nossa própria página de cadastro de usuários e, abandonamos a API de Login do google. As devidas mudanças foram realizadas nas documentações.

3. As principais lições aprendidas

3.0 Lições Aprendidas

Heroku

Aprendemos como utilizar o Heroku, bem como fazer o deploy da aplicação. Também aprendemos como funciona o ambiente de desenvolvimento e o de produção, visto que estamos utilizando o Sqlite no desenvolvimento e o Postgresql na produção.

API Google(Login)

Aprendemos alguns aspectos sobre a API do google.

Bootstrap

Aprendemos como implementar e utilizar a ferramenta Bootstrap nas nossas views. Também tivemos que buscar informações sobre o CSS e como funciona o Asset Pipeline do rails.

MVC

Obtivemos uma maior compreensão sobre como funciona a arquitetura MVC.

Hash

Aprendemos como utilizar a encriptação em senhas para oferecer uma maior segurança ao usuário

Arquitetura REST

Obtivemos uma maior compreensão de como as URIs e seus métodos HTTP são mapeados para as ações nos controladores, visto que tivemos que implementar o cadastro de usuários.

Restrições nos Models

Aprendemos como realizar algumas restrições nos models para que as informações não sejam inconsistentes.

Teste Unitários

Aprendemos a fazer testes unitários.

3.1 Considerações futuras

- Seguir o que foi proposto no Syllabus.
- Implementar o Log-in e o Log-out do usuário
- Criar o modelo para os Tickets
- Associar o usuário com os tickets