

Relatório da Sprint2 para o projeto CodeHub

A princípio, vale lembrar o que foi executado na primeira sprint, pois ela é a base para esta. No caso, foi implementado um banco de dados e um cookie para armazenamento de informações e dos versionamentos. Além disso, foi criada a lógica do repositório que conterá os arquivos versionados e implementado, também, funções de auxílio para orientar o usuário sempre que o mesmo precisar.

Agora, no que se refere a sprint atual, após as dailys realizadas, a equipe informou não haver tido dificuldade para realizar as tarefas, isto é, não houveram impedimentos.

Enfim, dito isso, quanto a implementação, a primeira realização foi a de exibir ao usuário todas as branches que existem, após isso, o usuário pode inserir a Hash da versão escolhida. Então, essa implementação favorece a de salvar a versão atual do usuário e, desse modo, os arquivos que já estão no diretório principal, são deletados. Mas, caso seja deletado com sucesso, os arquivos da branch serão copiados para o diretório atual, após a flag de "Projeto atual" ser alterada, e o projeto é, por fim, encerrado.

Outra implementação realizada foi de, dado a feature adicionar ao container, referente a sprint 2, o que foi implementado trata-se da seleção de arquivos informados pelo usuário, presentes na pasta do projeto, onde se está aplicando o versionamento, assim, alocando-os em uma área denominada de container, para que, a partir desses arquivos adicionados, seja criado uma nova versão do projeto.

Mais ainda, foi adicionado ao sistema o método histórico, que exibe ao usuário uma lista de todas as versões feitas do repositório atual, ordenadas a partir da data mais antiga para a mais recente e também indica, através de uma badge, qual a versão atual para o usuário. Portanto, para desenvolver o método, foi necessário acessar cada versão do repositório no 'banco de dados' e ordená-las conforme as datas do versionamento. Assim, após a ordenação das versões, a lista é exibida com as principais informações: id, comentário, autor e data.

Por fim, responsável por remover do container um arquivo selecionado pelo usuário, a classe `removeDoContainer` foi implementada para selecionar o arquivo `container.json` e manipular os dados através de um array de tipo string que recebe e permite o percurso dos paths existentes para os arquivos. Ademais, é excluído somente o arquivo, o qual o usuário informa, do container.

Concluindo, vale ressaltar que, todas as implementações propostas para a sprint vigente, foram efetuadas com êxito e que a equipe trabalhou com afinco, contornando os obstáculos que surgiram a tempo hábil.