

EC – Versionador de Arquivos

Para o estudo de caso, implementaremos um mecanismo de versionamento de arquivos de texto em linguagem C. Este projeto será desenvolvido e avaliado em múltiplas etapas e tem como inspiração a lógica do git.

ETAPA 1

Nesta etapa, deverá ser criado um arquivo executável, capaz de ser utilizado para adicionar arquivos de texto ao versionador de código. O versionador deverá ser capaz de:

- Selecionar arquivos que serão versionados e solicitar ao usuário digitar um texto vinculado àquele versionamento. Após isso, gerar um snapshot da versão dos arquivos selecionados;
- Ao gerar versões, através de leitura e escrita de arquivos, vocês devem ser capazes de registrar essas versões, a ordem das versões além de salvar os arquivos correspondentes a cada uma das versões;
- Exibir todos os snapshots feitos, com os textos gerados na ordem inversa em que foram registrados (do mais recente para o mais antigo) e os arquivos que foram versionados naquelas versões (não é necessário mostrar o conteúdo dos arquivos de texto, apenas os nomes dos arquivos que foram versionados – similar ao git log –stat).

Uso (via command line):

versionador.exe iniciar

Comando que deve criar todos os arquivos necessários para começar a acompanhar os arquivos do folder em que foi executado. Nesse comando, você criará uma pasta oculta chamada .versionador, e dentro dessa pasta oculta estarão todos os arquivos necessários para que as funcionalidades do versionador sejam executadas.

versionador.exe adiciona <arquivo_1>, <arquivo_2>, ..., <arquivo_n>

Ao utilizar o comando adiciona, você deverá ir marcando arquivos que serão adicionados ao próximo snapshot de versão. O comando adiciona pode ser utilizado múltiplas vezes para adicionar múltiplos arquivos manualmente.

versionador.exe registra “Texto”

O comando registra deve criar o snapshot de todos os arquivos marcados com o comando adiciona e adicionar esse snapshot ao banco de dados de versões.

versionador.exe log

O comando log deve listar todos os snapshots já feitos, com os respectivos textos.

Requisitos:

- Para se basearem no comportamento dessa versão simplificada de comandos, os alunos deverão se basear na lógica de funcionamento básico do git, com relação aos comandos git add, git log, git commit e git init. Além da forma como o git gerencia esses arquivos. Há um livro do git, o [Git Pro](#), que é um livro de leitura muito fácil. Para esta etapa do estudo de caso, basta ler os capítulos 1 e 2.
- O código vai ter que estar modularizado: cada funcionalidade tem que estar implementada em uma função separada e funções só realizam o que elas foram designadas para realizar. Por exemplo, uma função de leitura de vetor, só deve ler o vetor. Uma função de cálculo da média de um vetor deve calcular a média de um vetor que já foi lido, e assim sucessivamente.
- Os nomes das variáveis terão de ser significativos do papel delas em cada uma das funções.
- O código deverá ser capaz de receber entradas via command line. Para isso será necessário utilizar o argc e o argv da função main do código.
- Para a implementação, será necessário utilizar um array dinâmico de estruturas que representam os snapshots do código. O código deverá ser capaz de ler informações armazenadas em arquivos sobre o versionamento realizado. Então esse arquivo vai guardar toda a informação a respeito dos snapshots realizados. Seu código deve ser capaz de retirar informações dos snapshots feitos a partir da leitura do arquivo.
- O banco de dados do versionador será composto por dois arquivos que compõe o banco de dados. Um arquivo de snapshot que guardará informações sobre o snapshot e os arquivos versionados (o texto do snapshot, todos os arquivos que foram versionados). Haverá outro arquivo que guardará o conteúdo dos arquivos versionados. Os snapshots devem guardar informação a respeito de como ler os dados dos arquivos correspondentes àquele snapshot.