EC – Versionador de Arquivos

Para o estudo de caso, implementaremos um mecanismo de versionamento de arquivos de texto em linguagem C. A lógica de inspiração para esse projeto é o versionador geral de arquivos 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 executado pela command line e deverá ter os seguintes comandos:

versionador.exe iniciar

Comando que deve criar todos os arquivos necessários para começar um banco de dados de versionamento 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. Ao criar uma versão, você deve atribuir um número único (um identificador) para esse snapshot dos arquivos.

versionador.exe log

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

versionador.exe log --conteudo

O comando log deve listar todos os snapshots já feitos, com os respectivos textos e, para cada snapshot apresentado, o conteúdo dos arquivos registrados naquela versão deve ser exibido.

versionador.exe mostrar <identificador>

O comando deve exibir o texto registrado para aquela versão e todos os textos de todos os arquivos versionados naquela versão.

versionador.exe mudar <identificador>

Esse comando deve passar a exibir os arquivos como estavam na versão <identificador>. Para evitar perder arquivos atuais, você deve salvar um backup dos arquivos que seriam sobrescritos pela mudança de versão em uma pasta temporária (pois haverá comandos para voltar para o estado atual dos arquivos).

versionador.exe mudar –atual

Reverte os arquivos de uma versão específica para a versão atual dos arquivos. Estejam eles registrados em uma última versão ou não.

Requisitos:

- Para compreender a lógica desse versionador, é necessário entender o git. Para tal, faça um estudo do livro do git, o Git Pro, que é um livro de leitura muito fácil. Aprenda a usar o versionador com base nos tutoriais dos capítulos 1, 2, 3 e 7.

- O código vai ter que estar totalmente modularizado 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. Eventualmente pode ser necessário criar interfaces de funções, caso o código fique muito extenso.

- O código deve estar todo documentando: cada função deve ter uma descrição breve (como comentário) do que ela realiza além de uma descrição do que cada argumento da função significa. Seguir o padrão doxigen de documentação.

- O código deve ser legível: os nomes das variáveis terão de ser significativos do papel delas em cada uma das funções e os nomes das funções devem representar bem o que as funções realizam.

- O banco de dados do versionador será composto por dois arquivos que compõe o banco de dados. Um arquivo de versões que guardará informações sobre a versão e quais os arquivos que foram registrados naquela versão. O outro arquivo que guardará o conteúdo dos arquivos versionados. O arquivo de versões deve guardar informação a respeito de como ler os dados do conteúdo dos arquivos correspondentes àquele versão.

- Para exercitar o conceito de listas, para a implementação, será necessário utilizar uma lista encadeada da estrutura que representa cada versão, com ponteiros para o conteúdo dos arquivos (dentro do arquivo de versões). Assim, ao realizar cada comando, antes de qualquer coisa, será necessário criar uma lista encadeada das versões, com os ponteiros para o conteúdo dos arquivos (no arquivo de versões). Ao terminar de executar o comando, você deve desalocar a lista. Ou seja, todas as operações serão feitas com base na análise dessa lista encadeada de versões com ponteiros para os conteúdos dos arquivos.

- O projeto deverá ser versionado em um repositório git no Github (colocar o repositório privado). Cada commit deverá conter uma versão funcionando do seu projeto. Para facilitar a análise, cada commit deverá conter uma versão funcionando com uma nova funcionalidade da command line. Por exemplo, o primeiro commit do seu repositório de código deverá implementar uma versão do “versionador” que executa corretamente o comando “iniciar”. O próximo commit deverá ter uma versão funcionando do seu “versionador” com o comando “adiciona”. E assim sucessivamente. Para entrega do projeto, será necessário entregar esse código versionado no github (compartilhar com o professor).