Relatório

Trabalho de Linguagens de Programação
Parte 2: Código em Perl

25 de Novembro de 2016 – Turma 2016.2

Integrantes do Grupo: Diogo Nocera Magalhães / Bruno Machado Afonso
Professor: Miguel Elias Mitre Campista

1 – Introdução aos pricipais objetivos do código

De acordo com o relatório da 1ª parte do trabalho, este projeto consiste na criação de um Fórum. O código em Perl é responsável pelo tratamento geral dos posts e o gerenciamento de usuários, seus principais tópicos são:

- Tratar conteúdo inapropriado em mensagens ou tópicos, caso seja validado, o programa oculta a mensagem e explicita com uma mensagem de aviso que o conteúdo da mensagem é inapropriado.
- Inserção ou Remoção de usuários, tanto MEMBRO quanto ADMIN, de um simples banco de dados composto por um arquivo ".txt"
- Checagem de Login, caso o usuário se identifique ou não, o programa envia o status de login para a interface, que libera a exibição das opções relativas ao tipo de usuário que utiliza o software
- Checagem de Erro em dados de inscrição dos usuário
- Procura de Tópicos de Discussão a partir de uma entrada na pesquisa
- Resposta a usuários em posts
- Procura de posts de SPAM

2 – Descrição das funções

2.1 – userHandler.pm

insertUser [$userName] [$unCryptPassword] [$CPF]
RETURN: Inserção OK (1) / Erro Correspondente (0)

Função responsável por inserir um novo usuário no sistema com os dados recebidos via argumentos, criando um arquivo de texto "$userFileName" dentro da pasta /users/. Esta função támbém possui tratamento para: usuários com nomes iguais, nomes inapropriados e CPF Inválido.

- **deleteUser** [$userName]
  RETURN: Remoção OK (1) / Usuário não encontrado (0)

  Função responsável por remover um usuário no sistema.

- **chkPass** [$pwd]
  RETURN: Senhas Criptografadas (@passwords)

  Função responsável por encriptar uma senha fornecida pelo usuário com todos os "salts" possíveis do programa, com o intuito de checar a senha criptografada salva em arquivo de texto do usuário com a senha que foi fornecida para login.

- **validateLogin** [$userName] [$userPassword]
  RETURN: Validação OK (1) / Erro Correspondente (0)

  Função responsável por realizar o login do usuário.

- **createPost** [$author] [$response] [$message] [$topic]
  RETURN: Post OK (1) / Erro Correspondente (0)

  Função responsável por criar um novo post na forma de um arquivo de texto "post[indice]" (índice é para ordenar os posts na hora de exibição na interface) dentro da pasta /[$topic]/. O arquivo contém o nome do autor do post e a sua mensagem. Caso o post seja uma resposta a outro, o arquivo também conterá o nome do autor e a mensagem do post o qual está sendo respondido.

2.2 – filesHandler.pm

- **getFiles** [$dir]
  RETURN: vetor com todos os diretórios (@directories)

  Função responsável por extrair o nome de todos os diretórios localizados no caminho [$dir]. Primariamente utilizado para extrair os nomes de todos os tópicos dentro do "root".

- **getFilesList** [$dir]
  RETURN: vetor com todos os arquivos de texto (@filesTxt)

  Função responsável por extrair o nome de todos os arquivos de texto (.txt) localizados no caminho [$dir]. Primariamente utilizado para buscar por posts e usuários nos diretórios /[nome_do_topico] e /users/, respectivamente.

- **openTxtFile** [$dir]
  RETURN: NONE

  Função responsável por imprimir na linha de comando o conteúdo de todos os arquivos de texto dentro de um diretório passado via argumento.

- **SPAM_Finder** [$str_post]
  RETURN: Encontrou um post SPAM (1), ou não (0) [$found_SPAM]

  Função responsável por detectar SPAM em posts do Fórum. A função detecta se um texto está completamente em caixa alta.

- **chkTxtFilesSPAM** [$dir]
  RETURN: NONE

  Esta função direciona partes do texto de um post para a função "SPAM_Finder", podendo detectar traços de SPAM em todas as linhas do arquivo de texto.

- **chkTxtFilesCurses** [$dir]
  RETURN: NONE

  Esta função direciona partes do texto de um post para a função "levenshtein" dentro da biblioteca "textReader.pm", para detectar linguagem inapropriada dentro do arquivo de texto (melhor explicação da função se localiza na seção 2.3).

- **runAllDirectories**
  RETURN: NONE

  Função responsável por percorrer todos os diretórios dentro do "root" do programa, em busca de arquivos de texto para posts e autenticação de usuários.

- **getCurses** [$dir]
  RETURN: Array com palavras inapropriadas (@curses)

  Função responsável por extrair e guardar todas as palavras indevidas inseridas no arquivo "Curses.txt" fornecido via argumento da função.

- **chkAllDirectories**
  RETURN: NONE

  Esta função combina a execução das funções "getCurses", "chkTxtFilesCurses" e "chkTxtFilesSPAM" para melhor visualização do código.

2.3 – textReader.pm

- **leveshtein** [$str1] [$str2]
  RETURN: Distância entre duas palavras $dist[@ar1][@ar2]

  Esta função detecta a similaridade entre uma string de referência e uma alvo. A função conta a quantidade de alterações (que incluem remoção, inserção ou substituição de caracteres) que devem ser feitas na string de referência para chegar a ser igual a string alvo.

- **findString** [$bad_str] [$curse_word]
  RETURN: Palavra Indevida Encontrada (1) ou não (0).

  Função responsável por detectar instâncias de uma palavra indevida em uma string.

splitWords [$splitString]
RETURN: Array com as strings separadas (@newSplitWords)

Função responsável por remover pontuações e não-substantivos de uma frase em string. O intuito desta função é separar as palavras principais em uma busca por tópicos no Fórum.

- **calculateDistance** [$searchString] [$topicName]
  RETURN: Pontos de distância entre palavras ($totalPoints)

  Esta função une a execução das funções “leveshtein” e “splitWords”, calculando os pontos de distância entre palavras digitadas na pesquisa de tópico e os tópicos salvos em diretórios.

- **searchTopic** [$searchString]
  RETURN: Diretórios com tópicos próximos encontrados (@foundDirectories)

  Função responsável por detectar tópicos similares ao que foi pesquisado pelo usuário. Ela realiza isso com o auxílio da função “calculateDistance”.

- **validateCPF** [$numberCPF]
  RETURN: CPF Válido (1) / CPF Inválido (0)

  Função que realiza a validação de um CPF através do cálculo do digito verificador.

- **CryptPass** [$pwd]
  RETURN: Senha Criptografada ($crypt_msg)

  Função responsável por criptografar uma senha digitada pelo usuário para ser salva posteriormente no arquivo de texto respectivo. As senhas são criptografadas com 5 “salts” diferentes, escolhidos aleatoriamente.

- **chkName** [$dir] [$name]
  RETURN: Não há um usuário com o nome fornecido (0) / Usuário encontrado com o mesmo nome ($userIndex)

  Função responsável por verificar se há dentro do diretório /users/ algum usuário com o nome igual ao do que está tentando ser inserido.

- **chkCPF** [$dir] [$CPF]
  RETURN: Não há um usuário com o CPF fornecido (0) / Usuário encontrado com o mesmo CPF ($userIndex)

  Função responsável por verificar se há dentro do diretório /users/ algum usuário com o CPF igual ao do que está tentando ser inserido.