Validacoes.c: Nesse código, e onde ocorre todos os procedimentos que vão auxiliar o programa, tais como: Busca, Ordenação, Digitação, Verificação, e outros subprogramas auxiliares.

Int busca: Nesse procedimento, pensamos em uma forma mais genérica de utiliza a busca, de tal forma que fosse possível utilizá-la em todo caso de busca para os Usuários, uma busca mais simples porem que englobe boa parte dos tipos de buscas desejados, em vez utilizar a busca passando o arquivo e o parâmetro de busca, utilizamos somente o Arquivo e a Opção de busca. Utilizando de um switch para trabalhar com a opção de busca desejada, com base na sua decisão mandando o usuário digitar o que deve ser procurado, nome, código, telefone ou login. Retornando a posição pós-leitura.

Int CateBusca: Da mesma forma que fizemos uma busca genérica para o usuário pensamos em utilizar da mesma maneira para as categorias, onde o usuário digita a opção e o que será buscado, código ou nome. Retornando a posição pós-leitura.

Int ProdutoBusca: Também utilizamos da mesma maneira que o busca e a CateBusca, criamos esta busca somente para os produtos, onde o usuário digitara como deseja procurar e o que será buscado, Código, Título, Categoria e Descrição. Retornando a posição pós-leitura.

int VerificarUsuario: Uma busca normal, que possui o arquivo de usuários e a string do login, para verificar se já existe ou não. Retornando a sua posição em caso de existência e -1 caso contrário.

int VerificarCodUsuario: Através de uma busca, verifica se existe ou não um Usuário, com o código que é lhe passado como referência.

void digitarPreco: Procedimento utilizado para a escrita do preço do produto, utilizando de duas strings, uma para os valores inteiros antes da virgula e a outra as casas decimais do preço. Para identificar quando o usuário for digitar o ‘.’ para o numero decimais do preço, automaticamente depois de dois dígitos após o ponto, dar o resultado do preço digitado. Usando a função Atoi, em ambas strings, e somando as duas, obtendo o resultado do preço digitado.

int verificarAvaliacao: Função desenvolvida com o propósito de verificar se o usuário , já efetuou alguma avaliação no produto. O flag utilizado nesta função funciona ‘1’, para verificar se o produto a ser avaliado já foi avaliado pelo mesmo usuário que está avaliando. E ‘2’ para obter o código do produto através do código do usuário.

float CalcularMedia: Através do arquivo de avaliações e código do produto, esta função calcula a média de cada produto avaliado, verificando se o avaliador comprou o produto ou não, atribuindo pesos para a nota dada, baseando-se na compra; Sendo peso 2 para quem já adquiriu o produto e peso 1 para quem não possui o produto, também sempre somando os pesos a cada avaliação lida. Após somar todas as notas com seus devidos pesos, será dividida pela soma de todos os pesos dos avaliadores, retornando a média dos votos do produto. Se as somas dos pesos forem iguais a 0 significa não houve avalições para este produto, assim retornando -1.

void ordenar: Este ordenar , criado para a listagem dos produtos, teria que organizar de duas maneiras, 1º pela categoria e 2º pela escolha do usuário, podendo ser: Titulo, Preço ou Melhor Avaliação. Decidimos criar vetores que facilitaram a ordenação do nosso arquivo, sem alterar o arquivo. Passando como parâmetros os arquivos de produtos e de avalições, a quantidade de registros diferentes em dentro do arquivo de produto, obtido através do (fteel do arquivo do produto) dividido por (a quantidade de bytes da struct dos produtos); e a opção de ordenação desejada pelo usuário. O Método de ordenação utilizado foi o de Inserção Direta, por ser mais fácil de visualizar o como estava ocorrendo as trocas, e mais simples de implementar da maneira como será ordenada. Através de um switch podendo controlar como será ordenado o arquivo dos produtos, T – Titulo , P – Preço, A – melhor avaliação;