**CENTRO UNIVERSITÁRIO CAMPO REAL**

**ENGENHARIA DE SOFTWARE**

JOÃO VINICIUS YUJI NAKANO

BORIS PHILIPP WEBER DA LUZ

EMANUEL DE CRISTO SANTOS

**SEGUNDO PROJETO BIMESTRAL**

PROVA BIMESTRAL II

GUARAPUAVA

2023

Documentação do Projeto Bimestral 2

O programa inicia seu processo com a definição de sucessivas variáveis que perduram desde os caracteres até os vetores.

Em seguida, um laço Do-While executa. Todo o código é executado ao menos uma vez, quando, posteriormente, no menu, for selecionado o Voltar ao Login, o código reiniciará por esta primeira linha.

As variáveis relatorio\_menu, add\_menu e edicao\_menu são configuradas para o valor padrão: FALSE. Não havendo, portanto, conflito com valores antigos.

A primeira tela do programa é um pedido de RA do usuário que, posteriormente, é definida na variável ra\_user.

Inicia-se um loop For que percorre a Database de RAs (ra\_database), pegando seu tamanho com <vector>.size() como limite. Se o RA inserido for a Senha Super Secreta: "admin", ou também o RA consta na database, o usuário é tido como cadastrado (usuario\_cadastrado = ‘s’) e o loop se encerra. Caso contrário, se o iterador for igual ao tamanho da database e o RA não for encontrado, exibe-se a mensagem "RA não encontrado" e define a variável usuario\_cadastrado para ‘n’; o loop se encerra nas duas possibilidades.

Caso o usuário não seja cadastrado, um If é executado como verificação de um possível cadastro. Dentro do mesmo If, um loop Do-While é executado ao menos uma vez, perguntando se deseja prosseguir com o cadastro até um caractere válido ser inserido usando a estrutura SWITCH CASE. O caractere é armazenado na variável confirma\_cadastro, prosseguindo com o código normalmente.

Uma Estrutura de Decisão If verifica se o RA inserido no começo é válido, mostrando o salário do funcionário registrado; esse If é executado APENAS quando um usuário já é cadastrado no sistema. Um laço For faz o trabalho de iterar a Database de RAs e atribuir a posição do RA posto anteriormente na variável *index*, o qual será responsável por dar a localização exata do salário do usuário. Após isso, salarios\_liquidos.at(index) exibe o salário líquido do funcionário com uma mensagem acolhedora.

Após todos os trâmites, chegou a hora de cadastrar o usuário, isso é feito a partir de um If que verifica se o usuário NÃO É cadastrado E CONFIRMOU o prosseguimento do cadastro. Dentro dessa estrutura, variáveis são redefinidas para o padrão de fábrica, 0, para não dar conflitos com os vários loops em sequência.

O cadastro é feito perguntando o NOME, ENDEREÇO, CPF e DATA DE ADMISSÃO, esse último é gerado após uma sucessiva quantidade de laços While, de modo que a data seja a mais provável e coerente possível, ex: 02/02/2021, e não 99/60/1500.

Após a coleta dos dados principais, os dados de salário tomam espaço no código. Com um laço While, sendo a condição verificador\_tipo\_pessoa = 0, o código irá executar até que o tipo de pessoa inserido seja válido e encerre esse loop. O usuário escolherá entre os 3 possíveis casos: CONTRATADO (Efetivado), TEMPORÁRIO e ESTAGIÁRIO, cada um deles terá perguntas pertinentes a cada tipo de funcionário, como horas extras, dias trabalhados, etc. Após as perguntas serem feitas, toda a informação, já formatada, é transferida para a variável salario\_user, que será a PRINCIPAL variável na hora de calcular os descontos.

Feito isso, uma série de estruturas If são executadas, averiguando em qual classe salarial o funcionário pertence, atribuindo os descontos salariais (Desconto Previdenciário e Imposto de Renda) exclusivos a elas.

De modo subsequente, é tempo de calcular o salário do funcionário e atribuir tudo aos seus vetores próprios. Isso ocorre por meio de estruturas de decisão If e Else If, que verificam aquele mesmo tipo de funcionário posto em outros momentos (C/T/E), efetuando os devidos descontos salariais para cada funcionário de acordo com seu tipo inserido, armazenando, com a função vetorial push\_back, essas informações em vetores, como salario\_liquido\_(C/T/E)\_database e tipo\_funcionario\_(C/T/E)\_database.

Feitos os descontos devidos dentro dos Ifs, todas as outras informações, como SALÁRIOS LÍQUIDOS, RA, NOME, ENDEREÇO, CPF e DATA DE ADMISSÃO são colocadas, sem condições especiais, nos vetores de mesmo rótulo, guardando-os no sistema. Exibe-se uma mensagem que mostra o salário líquido do funcionário com base no que ele inseriu. Se ele preferir, pode colocar a Senha Secreta de Segurança para acessar o MENU do ADMINISTRADOR após essa etapa.

Agora, entra um detalhe técnico que é utilizado para efetuar somas CATEGORIZADAS dos salários de cada tipo com laços For. Esse método foi utilizado para exibir nos Relatórios os salários agrupados por classe e também facilitar os cálculos de outras informações, como a média salarial por categoria.

Se a Senha Super Secreta for inserida, que por acaso é “admin”, o MENU é exibido com três opções: RELATÓRIO, EDIÇÃO e retornar ao LOGIN. A escolha é armazenada e verificada com Switch Case, tornando TRUE a variável correspondente à opção e encerrando o Switch.

Se a opção 1 for digitada, a Estrutura de Decisão If com a condição relatorio\_menu == true é executada, exibindo o relatório completo com o Total de Funcionários, Total de Funcionários por Categoria, Média Salarial por Categoria, Valor Total por Categoria e Folha de Pagamento Salarial. Após o Relatório ser exibido ao usuário, atribui-se à variável permanecer\_edit\_mode o valor char de 'n', de modo a confirmar uma condição If futura que verifica a pergunta se o usuário deseja retornar ao menu ou sair. O Relatório Menu é também configurado ao seu estado default, FALSE.

Se a opção 2 for digitada, uma Estrutura de Decisão IF com a condição edicao\_menu == true é executada, exibindo um Menu de Edição de Nome, Endereço, CPF e Data de Admissão. Primeiro, pede-se ao usuário o Nome que digitou no Cadastro, colocando-o na variável string nome\_user, que será responsável por LOCALIZAR as INFORMAÇÕES do usuário, funcionando como COORDENADAS. Um laço For é inicializado que itera o tamanho da Database de Nomes, nome\_database, e se, em uma estrutura If, o Nome do Usuário for igual ao nome dentro do Vetor nome\_database, a variável *index* passa a ter o valor do iterador inteiro (index = i), a dita coordenada. Atribui-se, inclusive, à variável permanecer\_edit\_mode o char 's', para confirmar que o nome foi de fato encontrado, caso contrário, o código menciona que o nome não foi encontrado, retornando ao menu.

Após as verificações, um Laço While, usando a condição permanecer\_edit\_mode == ‘s’, com o Menu e as opções: (1), (2), (3) e (4), é executado, exibindo o NOME, ENDEREÇO, CPF e DATA DE ADMISSÃO para o usuário. Dentro desse loop, um Laço Do-While é executado para pegar o input dessas opções, ele será executado sempre que a opção escolhida estiver fora das 4 opções, e verificá-lo com Switch Case, exibindo a formatação adequada e o assunto da edição, ex: “Insira seu novo NOME, (exemplo: Maria Bethânia)”.

Após o Laço Do-While ser encerrado com uma opção correta, pergunta-se ao usuário se deseja permanecer na aba de Edição, para que não precise ficar retornando toda vez para alterar mais de uma informação, usando do Switch Case para atribuir o caractere devido a cada circunstância, SIM ou NÃO, à variável permanecer\_edit\_mode, retornando ou não até a primeira linha do Menu de Edição.

Uma estrutura If é executada após o encerramento do Modo Edição, com a condição add\_menu == false && permanecer\_edit\_mode == 'n'. Se a variável add\_menu tiver o valor booleano FALSE, que é o valor padrão é obtido quando se seleciona a opção Relatórios ou Edição, e se a variável permanecer\_edit\_mode tiver char 'n', que é também padrão, executa-se uma tela que pergunta do retorno ao Menu, mencionado em situações anteriores. Essa tela de pergunta, retorno\_menu, é feita através de um Laço Do-While que executa sempre enquanto a resposta for diferente de SIM, prevenindo que o usuário coloque um caractere errôneo e impossibilite a execução do código.

Se a variável retorno\_menu for ‘s’, então todo o código partindo do Menu Principal é executado novamente através do Laço Do-While com a condição retorno\_menu == ‘s’.

Da mesma forma, se, mesmo assim, retorno\_menu for igual a ‘n’, um outro Laço Do-While que abriga os Laços anteriores, verifica se a opção restante, add\_menu, é verdadeira. Se sim, então o código REINICIA desde a PRIMEIRA LINHA, senão, ele apenas encerra seu processo.