

### algoritmo

```
defina X 18
defina SMINIMO 112
declare funcionario[X] registro (nome literal, horastrabalhadas numerico, turno literal, categoria literal,
valor numerico)
declare escolha, i, j, cFuncionarios, horasmes, achou, valor, sInicial, aAlimentacao numerico
declare funcionario, turno, categoria literal

{ tabelaCategorias será no seguinte formato: }
{ [x][0] Categoria; [x][1] Turno; [x][2] valor }
declare tabelaC registro[6] (categoria literal, turno literal, valor numerico)
tabelaC[0] = {{G},{N},{18/100}}
tabelaC[1] = {{G},{M},{15/100}}
tabelaC[2] = {{G},{V},{15/100}}
tabelaC[3] = {{O},{N},{13/100}}
tabelaC[4] = {{O},{M},{10/100}}
tabelaC[5] = {{O},{V},{10/100}}

{ tabelaSalario será no seguinte formato }
{ x[0] salario base, x[1] salario teto, x[2] alimentacao }
declare tabelaS[3][2] numerico
tabelaS[0] = {{0},{300},{20/100}}
tabelaS[1] = {{300},{600},{15/100}}
tabelaS[2] = {{600},{999999},{5/100}}

escolha ← 0
enquanto (escolha != 3) faça {Menu de opções}
    imprima "1 – Cadastrar funcionario"
    imprima "2 – Mostre folha de pagamento"
    imprima "3 – Finalizar"
    leia escolha
    achou ← 0
    caso escolha {
        1: se cFuncionarios > X então
            imprima "Limite de funcionarios excedido" {Caso programa já atingiu limite de
cadastramento}
        senão
            imprima "Insira o nome do funcionario" {Lendo dados do funcionario}
            leia funcionario
            imprima "Insira o numero de horas trabalhadas no mes"
            leia horastrabalhadas
            imprima "Insira o turno de trabalho (M, V ou N)"
            leia turno
            enquanto achou == 0 faça {Força ao usuario digitar corretamente}
                se (turno == M || turno == V) || turno == N então
                    imprima "Insira a categoria (O ou G)"
                    leia categoria
                    achou ← 1 {"achou" é sinalizador}
                    se categoria == O || categoria == G então
```

```

                                achou ← 1
senão
    imprima "Insira uma categoria válida"
                                achou ← 0
fim-se
senão
    imprima "Insira um turno válido"
                                achou ← 0
fim-se
fim-enquanto {Salvado informações no registro}
funcionario[cFuncionarios].nome ← funcionario
funcionario[cFuncionarios].horastrabalhadas ← horastrabalhadas
funcionario[cFuncionarios].turno ← turno
funcionario[cFuncionarios].categoria ← categoria
contador ← contador+1
fim-1

2: para i de 0 até contador-1 faça {fixando usuario}
    para j de 0 até 6 faça {fixando tabela salario para comparação}
        se (funcionario[i].categoria = tabelaC[j].categoria) && (funcionario[i].turno ==
tabelaC[j].turno)
            funcionario[i].valor ← tabelaC[j][2]*SMINIMO
            sBase ← SMINIMO*tabelaC[j][2]* funcionario[i].horastrabalhadas
        fim-se
        para j de 0 até 3 faça {fixando tabela alimentação para comparação}
            se (sBase > tabelaS[j][0] && sBase <= tabela[j][1])
                aAlimentacao ← sBase * tabela[j][2]
            fim-se
        fim-para
        imprima funcionario[i].nome, funcionario[i].horastrabalhadas, funcionario[i].valor,
sBase, aAlimentacao, aAlimentacao+sBase {imprimindo resultados}
    fim-2
fim-caso
fim-enquanto
fim-algoritmo

```

## Algoritmo em "C"

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define X 18
#define SMINIMO 112

struct Funcionario {
    char nome[255], turno[2], categoria[2];
    int horastrabalhadas;
    float valor;
};

void main() {
    struct Funcionario funcionario[X];
    char cat[6][2][2] = {{"G","N"}, {"G","M"}, {"G","V"}, {"O","N"}, {"O","M"}, {"O","V"}};
    float por[6][1] = {{18*0.01}, {15*0.01}, {15*0.01}, {13*0.01}, {10*0.01}, {10*0.01}};
    float ali[3][3] = {{0,300,20*0.01}, {300,600,15*0.01}, {600,9999999,5*0.01}};
    system("clear");
}

```

```

char nome[255], turno[2], categoria[2];
int escolha, i, j, cFuncionarios, horasmes, achou;
float valor, sBase, aAlimentacao;
cFuncionarios = 0;
escolha = 0;
while (escolha != 3) {
    printf("*****\n");
    printf("1 - Cadastrar funcionario\n");
    printf("2 - Mostre folha de pagamento\n");
    printf("3 - Finalizar\n");
    printf("*****\n\n");
    scanf("%i", &escolha);
    achou = 0;
    switch (escolha) {
        case 1: system("clear");
            if (cFuncionarios >= X)
                printf("Limite de funcionarios excedido\n\n");
            else
                printf("Insira o nome do funcionario: ");
                scanf("%s", nome);
                printf("\nInsira o numero de horas trabalhadas no mes: ");
                scanf("%i", &horasmes);
                while (achou == 0) {
                    printf("\nInsira o turno de trabalho (M, V ou N): ");
                    scanf("%s", turno);
                    if (strcmp(turno, "M") == 0 || strcmp(turno, "V") == 0 || strcmp(turno, "N")
== 0) {
                        while (achou == 0) {
                            printf("\nInsira a categoria (O ou G): ");
                            scanf("%s", categoria);
                            achou = 1;
                            if (strcmp(categoria, "O") == 0 || strcmp(categoria, "G") == 0)
                                achou = 1;
                            else {
                                printf("\nInsira uma categoria valida\n\n");
                                achou = 0;
                            }
                        }
                    }
                    else {
                        printf("\nInsira um turno valido\n\n");
                        achou = 0;
                    }
                }
                strcpy(funcionario[cFuncionarios].nome, nome);
                funcionario[cFuncionarios].horastrabalhadas = horasmes;
                strcpy(funcionario[cFuncionarios].turno, turno);
                strcpy(funcionario[cFuncionarios].categoria, categoria);
                cFuncionarios = cFuncionarios+1;
                system("clear");
                printf("\n\nFuncionario cadastrado com sucesso\n\n");

                break;

```

```

    case 2: system("clear");
    printf("Nome\tHoras No Mes\tValor Hora\tAuxilio Alimentacao\tSalario
Bruto\tSalario Liquido\n");
    for ( i = 0; i <= cFuncionarios-1; i++) {
        for (j = 0; j < 6; j++) {
            if ((strcmp(funcionario[i].categoria, cat[j][0]) == 0) &&
(strcmp(funcionario[i].turno, cat[j][1]) == 0)) {
                funcionario[i].valor = por[j][0]*SMINIMO;
                sBase = funcionario[i].horastrabalhadas*por[j][0]*SMINIMO;
            }
        }
        for (j = 0; j < 3; j++){
            if (sBase > ali[j][0] && sBase <= ali[j][1])
                aAlimentacao = sBase*ali[j][2];
        }

        printf("%s\t%i\tR$%.2f\tR$%.2f\tR$%.2f\tR$%.2f\n",funcionario[i].nome,
funcionario[i].horastrabalhadas, funcionario[i].valor, aAlimentacao, sBase, (sBase+aAlimentacao));
    }
    printf("\n\n");
}
}
}

```