

```

int nomeDosIntegrantes();
int criarVetor(int *v);
int exhibirVetor(int *v);
int shellSort(int *v);
int primo(int numero);
int fibonacci(int num, int a = 1, int b = 0);
int funcaoRecursiva(int n);

int atividadeAvaliativa2020()
{
    int opcao, continua, vetor[T], num, n, numero;

    continua = 1;

    do
    {
        system("cls");
        cout << "=== OLÁ VOCÊ ESTÁ DENTRO DA NOSSA ATIVIDADE AVALIATIVA
===\n\n";
        cout << "1 - NOMBES DOS INTEGRANTES DO GRUPO\n";
        cout << "2 - CRIAR VETOR\n";
        cout << "3 - EXIBIR VETOR\n";
        cout << "4 - SHELLSORT\n";
        cout << "5 - NUMEROS PRIMOS\n";
        cout << "6 - FIBONACCI\n";
        cout << "7 - RECURSIVIDADE\n";
        cout << "0 - RETORNAR AO PROGRAMA PRINCIPAL\n\n";
        cout << "SUA OPÇÃO --> ";
        cin >> opcao;
        switch(opcao)
        {
            case 1: nomeDosIntegrantes();
                    break;
            case 2: criarVetor(vetor);
                    break;
            case 3: exhibirVetor(vetor);
                    break;
            case 4: shellSort(vetor);
                    break;
            case 5: printf("DIGITE QUANTOS NUMEROS DESEJA GERAR? ");
                    scanf("%d", &numero);
                    primo(numero);
                    system("pause");
                    break;
            case 6: printf("DIGITE QUANTOS NUMEROS DESEJA GERAR? ");
                    scanf("%d", &num);
                    fibonacci(num);
                    system("pause");
                    break;
            case 7: n = 1;
                    funcaoRecursiva(n);
                    break;
            case 0: continua = 0;
                    break;
            default: cout << "OPCAO INVÁLIDA. TENTE NOVAMENTE\n";
                     system("pause");
                     break;
        }
    }
}

```

```

        }while(continua==1);

        cout << "RETORNANDO....\n";
        system("pause");
    }

int nomeDosIntegrantes()
{
    cout << "===HUGO=FERNANDES=DE=OLIVEIRA=BARROS===\n";
    cout << "=====JOÃO=PEDRO=COSTA=PINHO=====\n";
    cout << "=====HIGOR=LAVIOLA=====\n";
    cout << "=====ANDREY=====\n";
    cout << "=====DANILO=GARCIA=====\n";
    system("pause");
}

int criarVetor(int *v)// CRIANDO O VETOR.
{
    int i;
    srand(time(NULL));
    for(i=0;i<T;i++)
    {
        v[i] = rand()%MAXR;
    }
    cout << "VETORES CRIADOS COM SUCESSO!\n\n";
    system("pause");
    system("cls");
}

int exibirVetor(int *v)// EXIBIR O VETOR DESORDENADO.
{
    int i;
    cout << "\n=== VETOR ATUAL ===\n";
    for(i=0;i<T;i++)
    {
        cout << "Índice: " << i << " -> " << v[i] << "\n";
    }
    cout << "\n === FIM DO VETOR ===\n";
    system("pause");
    system("cls");
}

int shellSort(int *v)// CÓDIGO SHELLSORT, ORDENAÇÃO DO VETOR.
{
    int i , j, h, value, inicio, fim;

    h = 1;

    inicio = clock();
    while(h < T)
    {
        h = 3*h+1;
    }
    while (h > 0)
    {
        for(i = h; i < T; i++)
        {
            value = v[i];

```

```

        j = i;
        while (j > h-1 && value <= v[j - h])
        {
            v[j] = v[j - h];
            j = j - h;
        }
        v[j] = value;
    }
    h = h/3;
}
fim = clock();
cout << " O VETOR FOI ORDENADO EM: " << fim-inicio << "ms\n";
system("pause");
}

int primo(int numero)// RECURSIVIDADE NÚMEROS PRIMOS.
{
    int i, n;
    bool ePrimo;

    n = 2;

    while(numero>0)
    {
        ePrimo = true;
        for(i=2; i<n; i++)
        {
            if(n%i==0)
            {
                ePrimo = false;
            }
        }
        if(ePrimo)
        {
            cout << "[" << numero << "]" O NÚMERO " << n << " É PRIMO\n";
            numero--;
        }
        n++;
    }
}

int fibonacci(int num, int a, int b)// RECURSIVIDADE FIBONACCI.
{
    if(num>0)
    {
        cout << a << " - - ";
        fibonacci(--num,a+b,a);
    }else{
        cout << "SEQUÊNCIA FIBONACCI\n\n";
    }
}

int funcaoRecursiva(int n)// FUNÇÃO RECURSIVA
{
    cout << "O VALOR n É:" << n << "\n";
    cout << "ESTA É A" << n << "ª CHAMADA DA FUNÇÃO\n";
    system("pause");
    if(n>3)

```

```
{
    n - 1;
}else{
    n++;
    funcaoRecursiva(n);
}
cout << "ESTOU NA CHAMADA " << n << "DA FUNÇÃO\n";
system("pause");
}
```