

UNIVERSIDADE FEDERAL FLUMINENSE

Alline Ribeiro

Edson Onildo

Isabela Carvalho

PROGRAMAÇÃO COM ARQUIVOS:

Relatório do Primeiro Trabalho

Rio das Ostras, 11 de abril de 2018

- Código Fonte:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define N 20
```

```
typedef struct elemento {
```

```
    short int num;
```

```
    int quant;
```

```
    struct elemento *esq;
```

```
    struct elemento *dir;
```

```
} Elemento;
```

```
Elemento *cria()
```

```
{
```

```
    return NULL;
```

```
}
```

```
Elemento *insereElemento(Elemento *a, short int num)
```

```
{
```

```
    if (a == NULL)
```

```
    {
```

```

    a = (Elemento *) malloc(sizeof(Elemento));

    a->num = num;

    a->quant = 1;

    a->esq = NULL;

    a->dir = NULL;

}

else if (num < a->num)

{

    a->esq = insereElemento(a->esq, num);

}

else

{

    a->dir = insereElemento(a->dir, num);

}

return a;

}

```

```

Elemento *existe(Elemento *a, int num)

{

```

```

    if (a == NULL || a->num == num)

        return a;

    if (a->num > num)

        return existe(a->esq, num);

    else

        return existe(a->dir, num);

}

void emOrdem(FILE *saida, Elemento *a)

{

    if (a != NULL)

    {

        emOrdem(saida, a->esq);

        fprintf(saida, "%hu,%d\n", a->num, a->quant);

        emOrdem(saida, a->dir);

    }

}

int main(void)

{

    char arqEntrada[100], arqSaida[100], plot[100];

```

```
printf("Digite o nome do arquivo de entrada:\n");
```

```
scanf("%s", arqEntrada);
```

```
int i;
```

```
FILE *entrada;
```

```
Elemento *lista;
```

```
Elemento *aux;
```

```
FILE *saida;
```

```
short int numero;
```

```
for (i = 0; i < 2; i++)
```

```
{
```

```
    if (i == 0)
```

```
    {
```

```
        entrada = fopen(arqEntrada, "rb");
```

```
    }
```

```
    else
```

```
    {
```

```
        strcat(arqEntrada, ".zip");
```

```
        entrada = fopen(arqEntrada, "rb");
```

```
}
```

```
lista = cria();
```

```
aux = NULL;
```

```
while (!feof(entrada))
```

```
{
```

```
    fread(&numero, 1, sizeof(short int), entrada);
```

```
    aux = existe(lista, numero);
```

```
    if (aux!=NULL)
```

```
    {
```

```
        aux->quant++;
```

```
    }
```

```
    else
```

```
    {
```

```
        lista = insereElemento(lista, numero);
```

```
    }
```

```
}
```

```
fclose(entrada);
```

```
strcpy(arqSaida, arqEntrada);
```

```
saida = fopen(strcat(arqSaida, ".csv"), "w");
```

```
fprintf(saida, "bits,frequencia\n");
```

```
emOrdem(saida, lista);
```

```
fclose(saida);
```

```
printf("Arquivo gerado: %s\n", arqSaida);
```

```
strcpy(plot, "./plot ");
```

```
strcat(plot, arqSaida);
```

```
system(plot);
```

```
}
```

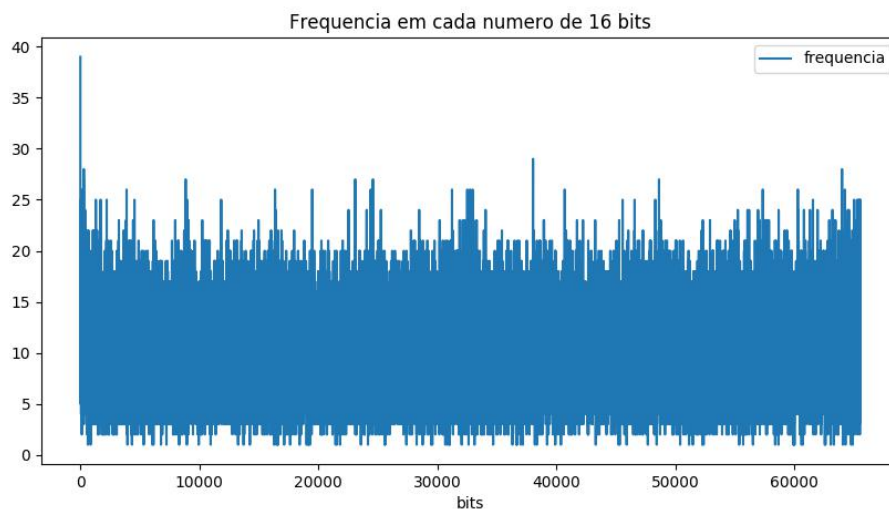
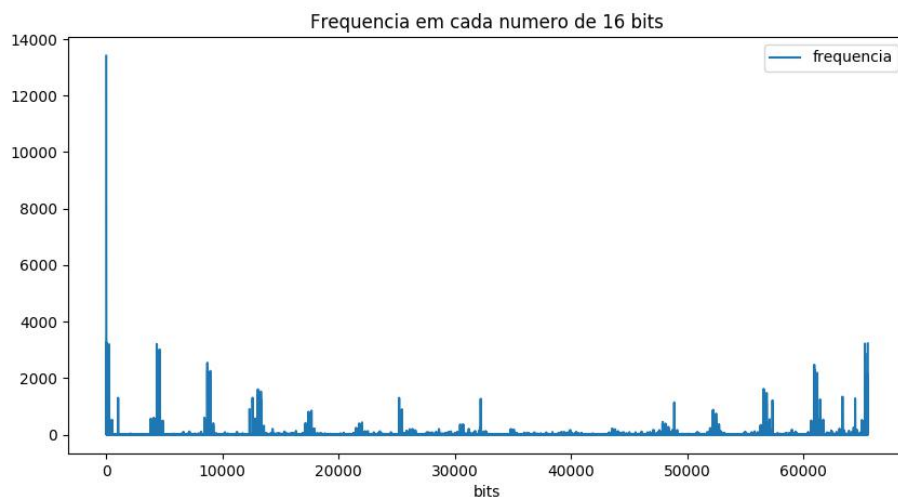
```
return 0;
```

```
}
```

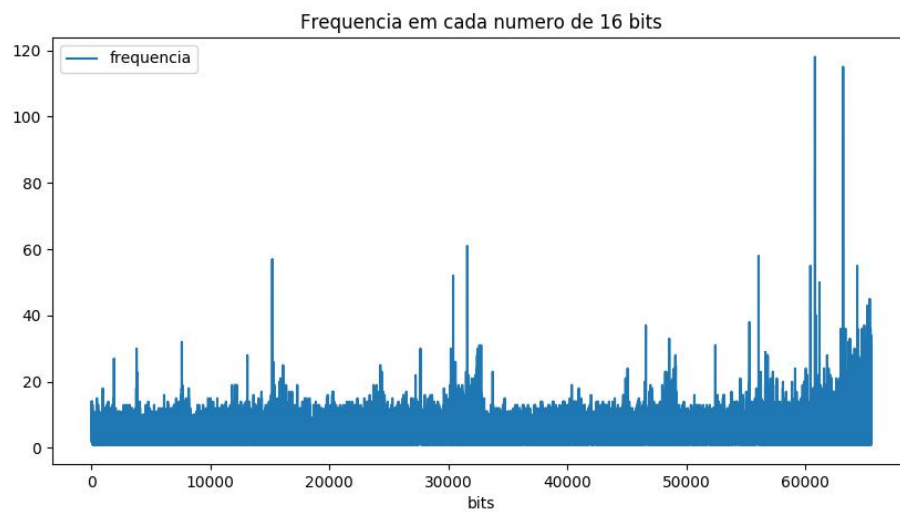
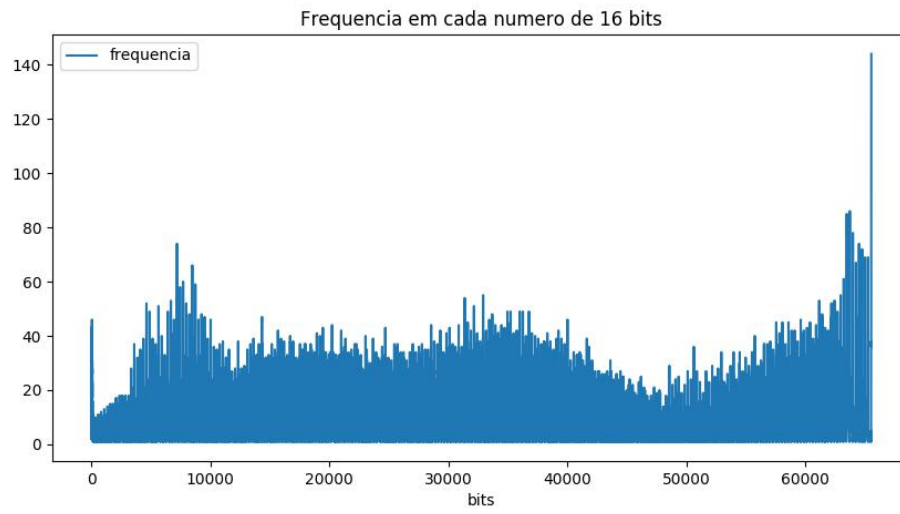
- **Análise dos gráficos:**

Os gráficos a seguir denotam a frequência com que um número de 16 bits se repete em arquivos de diferentes formatos. Os primeiros gráficos de cada formato mostram esta frequência para os arquivos originais, isto é, antes de passarem pelo algoritmo de compressão. Enquanto que os gráficos seguintes, mostram a frequência depois dos arquivos terem sido comprimidos. Note que a frequência dos números inteiros cai consideravelmente.

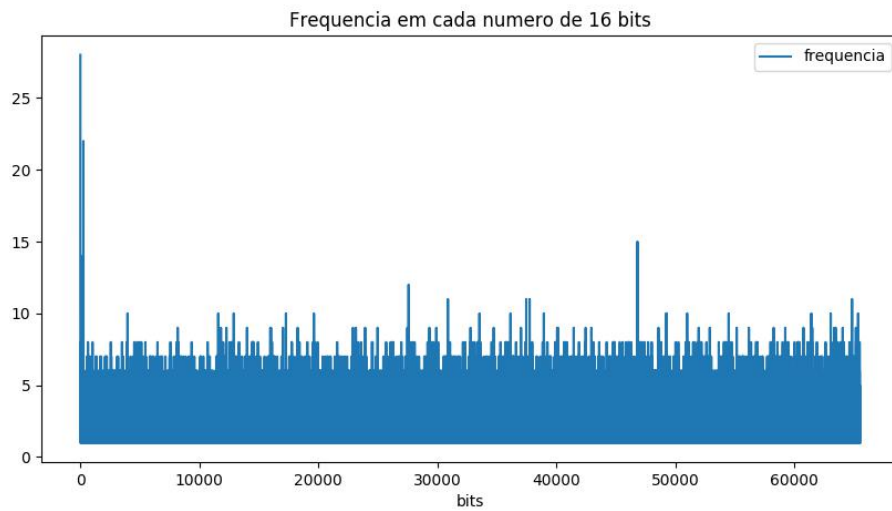
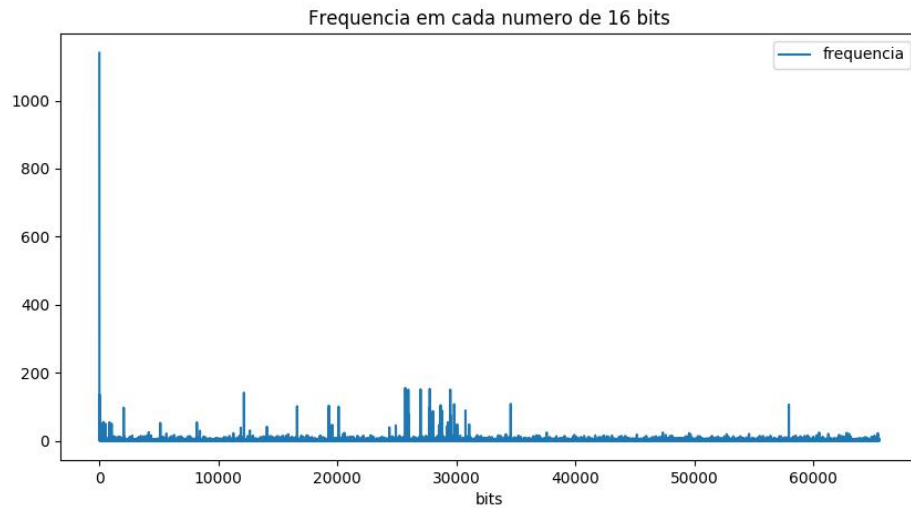
1. AVI



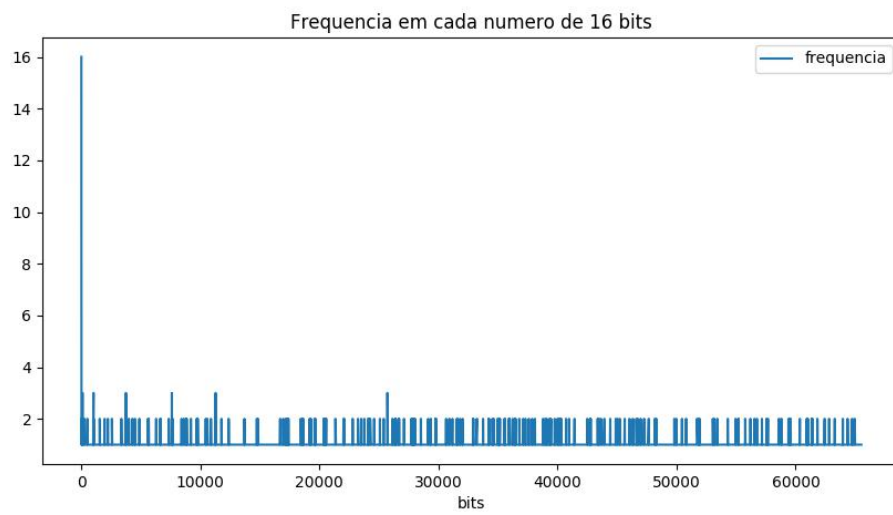
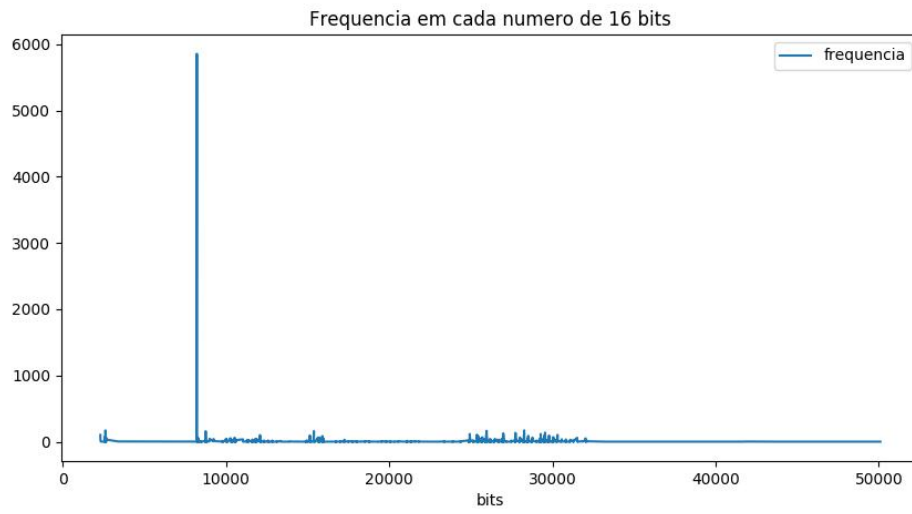
2. Bitmap



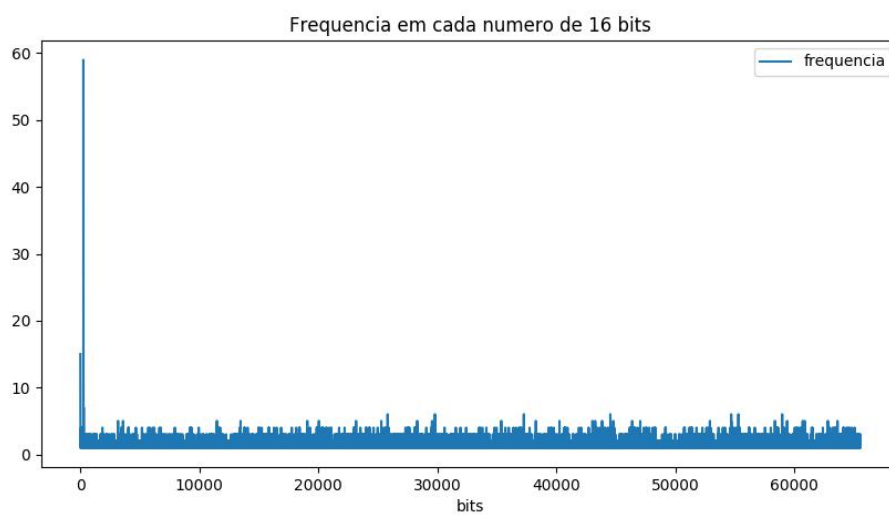
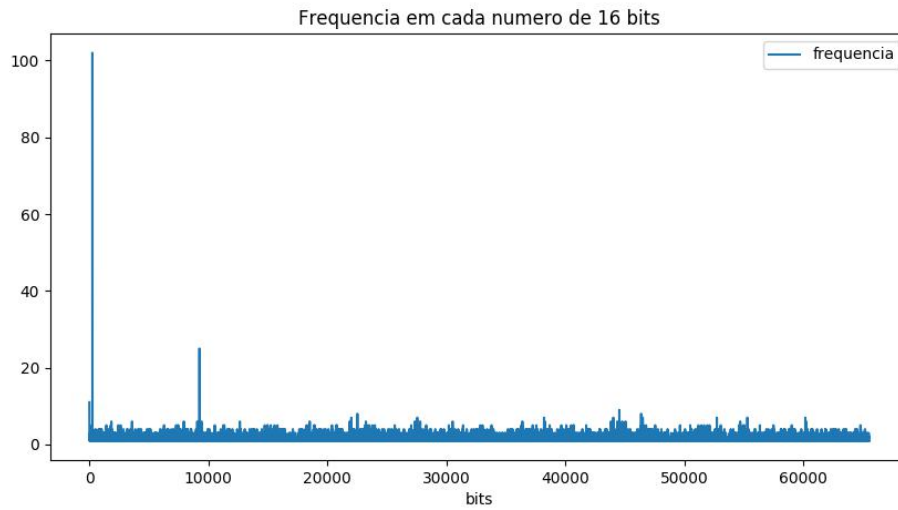
3. DOC



4. HTML



5. JPG



6. MP3

