

## Documentação

### Decisão tomada para Implementação da lista encadeada :

Para a implementação da lista encadeada decidi , criar um vetor de 2000 posições para armazenar o conteúdo das linhas que começavam com a palavra ATOM no arquivo , depois de armazenado o conteúdo da linha , foi alocado na memória a struct que armazena os dados de cada átomo , usei a função sscanf para pegar alguns dos dados do vetor e armazená-los na struct alocada , depois foi feito o apontamento da struct para que se fosse criada uma lista encadeada .

### Testes realizados :

Testei o código com todos os arquivos pdb é a saída do programa correspondeu com saída esperada .

### Soluções para problemas encontrados :

Tive um grande problema em mostrar quantos átomos e aminoácidos de cada tipo tinham no arquivo .

Depois de muitos erros cheguei em lógica que funcionou .

**Solução:** Foi criado um vetor de structs , depois fui percorrendo a lista encadeada testando se o átomo ou aminoácido que estava comparando na lista já existia no vetor de struct , se ele já existisse eu somava mais um no contador da struct , senão eu adicionava o átomo ou aminoácido no vetor de structs .

### Estruturas Usadas :

1)

```
typedef struct propriedades
```

```
{  
    int num;  
    char nome[6];  
    char aminoacido[6];  
    float x;  
    float y;  
    float z;    struct propriedades *prox;  
} ATOM;
```

Essa struct foi criada para armazenar dados de cada átomo .

2)

```
typedef struct aux{  
    char atomo[6];  
    int quantos;  
}quantidade;
```

Struct criada para armazenar a quantidade de átomos de cada tipo.

3)

```
typedef struct aminoacidos{  
    char aminoacido[6];  
    int quantidade;  
}quant;
```

Struct criada para armazenar a quantidade de aminoácidos de cada tipo.