

Documentação: TPs 2 e 3 – Calendário

1. Introdução

A necessidade da contagem e organização do tempo, o registro de datas e eventos importantes e a identificação de ciclos astronômicos periódicos passíveis de serem acompanhados, dão origem ao estabelecimento de calendários. Os primeiros calendários produzidos remontam às civilizações e povos antigos, sendo que, o primeiro registro encontrado é do ano de 2700 a.C. dos sumérios (povo da Mesopotâmia).

Há vários tipos de calendário (solar, lunar, misto e etc.) que variam de acordo com sua referência astronômica e com seu grau de exatidão matemática. O calendário usado atualmente no ocidente é o calendário gregoriano, promulgado pelo Papa Gregório XIII em 24 de Fevereiro de 1582. Nesse calendário, o ano tem 365 dias, o que quase coincide com o ano solar/trópico (tempo exato que a Terra leva para completar uma volta completa ao redor do Sol) que possui 365 dias, 5 horas, 49 minutos e 27 segundos; devido a essa diferença, estabeleceu-se a necessidade da criação do ano bissexto. Se não houvesse anos bissextos, a cada quatro anos, o calendário estaria defasado em um dia em relação ao ano trópico, o que significa que, os equinócios e os solstícios (marcam as mudanças de estações e possuem datas fixas exatas no calendário) ocorreriam em datas diferentes a cada quatro anos.

Um ano é bissexto quando possui 366 dias, ou seja, há o acréscimo de um dia em relação à anos não bissextos. Em anos bissextos o mês de Fevereiro possui 29 dias, enquanto em anos não bissextos, possui apenas 28. Abaixo estão expressas as regras para a determinação dos anos bissextos. O calendário gregoriano possui 12 meses organizados em 28, 29, 30 ou 31 dias. As semanas possuem 7 dias, então um ano possui 52 semanas inteiras e uma quantidade variável de dias a mais (dependendo se o ano é bissexto ou não). A *Figura 01* mostra um exemplo da organização de um ano (2019) no calendário gregoriano.

A estrutura e organização de um calendário gregoriano são as seguintes:

Meses	Total de dias
Janeiro	31
Fevereiro	28 ou 29
Março	31
Abril	30
Maio	31
Junho	30
Julho	31
Agosto	31
Setembro	30
Outubro	31
Novembro	30
Dezembro	31

(a)

Número	Dias da Semana
1	Domingo
2	Segunda-feira
3	Terça-feira
4	Quarta-feira
5	Quinta-feira
6	Sexta-feira
7	Sábado

(b)

Tabela 01: (a) Meses sequenciais do calendário gregoriano em relação ao número total de dias
(b) Ordem dos dias em relação a cada dia da semana

* 29 dias em anos bissextos e 28 dias em anos não bissextos.

A regra para determinar se um ano é bissexto ou não é:

- São bissextos todos os anos múltiplos de 400;
- São bissextos todos os múltiplos de 4, exceto se for múltiplo de 100 mas não de 400;
- Não são bissextos todos os demais anos;

A maioria dos países do mundo adota o calendário gregoriano, no ocidente é o principal calendário utilizado. Apesar disso, muitas críticas foram feitas a sua estrutura e organização ao longo dos séculos. As principais delas são em relação à irregularidade dos dias totais de cada mês (que variam de 28 a 31) e à falta de sincronismo entre os meses e semanas, já que, os meses não possuem uma quantidade exata e inteira de semanas (o que dificulta a organização prática da passagem de tempo).

Janeiro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
1		1	2	3	4	5	6
2	7	8	9	10	11	12	13
3	14	15	16	17	18	19	20
4	21	22	23	24	25	26	27
5	28	29	30	31			

Fevereiro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
5					1	2	3
6	4	5	6	7	8	9	10
7	11	12	13	14	15	16	17
8	18	19	20	21	22	23	24
9	25	26	27	28			

Março							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
9					1	2	3
10	4	5	6	7	8	9	10
11	11	12	13	14	15	16	17
12	18	19	20	21	22	23	24
13	25	26	27	28	29	30	31

| | | | | | | | |

Abril							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
14	1	2	3	4	5	6	7
15	8	9	10	11	12	13	14
16	15	16	17	18	19	20	21
17	22	23	24	25	26	27	28
18	29	30					

Maio							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
18			1	2	3	4	5
19	6	7	8	9	10	11	12
20	13	14	15	16	17	18	19
21	20	21	22	23	24	25	26
22	27	28	29	30	31		

Junho							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
22						1	2
23	3	4	5	6	7	8	9
24	10	11	12	13	14	15	16
25	17	18	19	20	21	22	23
26	24	25	26	27	28	29	30

| | | | | | | | |

Julho							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
27	1	2	3	4	5	6	7
28	8	9	10	11	12	13	14
29	15	16	17	18	19	20	21
30	22	23	24	25	26	27	28
31	29	30	31				

Agosto							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
31				1	2	3	4
32	5	6	7	8	9	10	11
33	12	13	14	15	16	17	18
34	19	20	21	22	23	24	25
35	26	27	28	29	30	31	

Setembro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
35							1
36	2	3	4	5	6	7	8
37	9	10	11	12	13	14	15
38	16	17	18	19	20	21	22
39	23	24	25	26	27	28	29
40	30						

| | | | | | | | |

Outubro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
40		1	2	3	4	5	6
41	7	8	9	10	11	12	13
42	14	15	16	17	18	19	20
43	21	22	23	24	25	26	27
44	28	29	30	31			

Novembro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
44					1	2	3
45	4	5	6	7	8	9	10
46	11	12	13	14	15	16	17
47	18	19	20	21	22	23	24
48	25	26	27	28	29	30	

Dezembro							
Semana	G	A	B	C	D	E	F
	Sg	Tr	Qa	Qi	Sx	Sb	Dm
48							1
49	2	3	4	5	6	7	8
50	9	10	11	12	13	14	15
51	16	17	18	19	20	21	22
52	23	24	25	26	27	28	29
1	30	31					

Figura 01: Calendário gregoriano do ano de 2019

2. Objetivos e Funcionalidades

O objetivo deste trabalho prático (TP) é projetar e implementar uma aplicação de calendário. Esse trabalho pode ser dividido em duas partes (TP II e TP III), as funcionalidades a serem executadas por cada uma dessas partes estão resumidamente expressas na tabela abaixo:

Funcionalidade	TP2	TP3
Interface com o usuário		
Mês e Ano	•	•
Idioma	•	•
Eventos a serem impressos no calendário	•	•
Detalhes de impressão do calendário	•	•
Impressão do calendário como uma lista	•	•
Impressão do calendário como um arquivo pdf		•
Arquivo de eventos	•	•
Consulta eventos		•

Tabela 02: Tabela resumida das funcionalidades presentes em cada um dos TPs

Como é possível observar na tabela acima, o TP 3 é uma continuação do TP 2. Então, o TP 3 tem todas as funcionalidades do TP 2, porém com o acréscimo da impressão do calendário como um arquivo pdf (*sub-item 2.4*) e da consulta de eventos (*sub-item 2.5*). Abaixo, estão expressos os detalhes de cada uma das funcionalidades que deverão ser implementadas nos TPs 2 e 3 (funcionalidades presentes na *Tabela 01*).

2.1. Interface com o usuário – TP 2 e 3

Deve tratar os seguintes aspectos a serem fornecidos pelo usuário pela entrada padrão:

- Mês e Ano: Mês deve ser um número inteiro entre 1 e 12, sendo 1 o mês de Janeiro e 12 o de Dezembro. O ano pode ser qualquer valor entre 1900 e 2999;
- Idioma: Pode ser Português (0) ou Inglês (1). Isso significa que os dias da semana e o mês devem ser impressos no idioma escolhido;
- Eventos a serem impressos no calendário: deve-se definir entre os eventos existentes em um arquivo, cujo nome deverá ser fornecido pelo usuário, o(s) tipo(s) daqueles que serão impressos;
- Detalhes de impressão do calendário: No caso de uma lista, deve-se imprimir na forma de uma lista na saída padrão de dados (monitor) ou um arquivo pdf (TP 3);

2.2. Arquivo de eventos – TP 2 e 3

O arquivo de eventos deve ser criado antes de imprimir o calendário. Ou seja, esse arquivo deve estar disponível antes de gerar o calendário. O formato do registro desse arquivo é o seguinte:

<Evento> <Dia> <Mês> <Ano> <Descrição do evento>
--

→ Onde:

<Evento>

 deve ser uma descrição de até 25 caracteres e pode ser:

- A: Aniversário;
- F: Feriado;
- V: Viagem;
- O: Outros;

<Dia> <Mês> <Ano> deve ser uma data válida. No entanto, se algum desses três campos for 0, isso significa que aquele evento é repetido todo o dia, mês, ano dependendo de onde se encontra o valor zero.

→ Observações importantes:

- Este programa não fará a consistência de cada um desses valores e, assim, será assumindo que não há erro;
- Cada um dos campos do registro é separado por pelo menos um espaço em branco;
- O arquivo de eventos não está ordenado por data;

2.3. Impressão no monitor na forma de uma lista – TP 2 e TP3

Após definir as opções de interface com o usuário e acessar o arquivo de eventos, deve-se gerar uma lista na saída de dados padrão (monitor) com o mês escolhido e com os eventos referentes àquele mês. A *Figura 02* (abaixo) mostra um exemplo de impressão do calendário no formato de uma lista do mês de Fevereiro (mês selecionado) – *Figura 02 (a)* – com alguns eventos (obtidos no arquivo de eventos – entrada do programa) – *Figura 02 (b)*.

(a)	<div><div>Fevereiro 2019</div><div>1 Sexta-feira</div><div>2 Sábado</div><div>3 Domingo</div><div>4 Segunda</div><div>A Zeca Pagodinho</div><div>V São Paulo, SP</div><div>5 Terça-feira</div><div>6 Quarta-feira</div><div>7 Quinta-feira</div><div>8 Sexta-feira</div><div>9 Sábado</div><div>10 Domingo</div><div>11 Segunda</div><div>12 Terça-feira</div><div>13 Quarta-feira</div><div>14 Quinta-feira</div><div>15 Sexta-feira</div><div>16 Sábado</div><div>17 Domingo</div><div>18 Segunda</div><div>19 Terça-feira</div><div>20 Quarta-feira</div><div>21 Quinta-feira</div><div>22 Sexta-feira</div><div>Reunião de câmara</div><div>23 Sábado</div><div>24 Domingo</div><div>25 Segunda</div><div>26 Terça-feira</div><div>27 Quarta-feira</div><div>28 Quinta-feira</div></div> <div><div>0 22 2 2019 Reunião de câmara</div><div>V 4 2 2019 São Paulo, SP</div><div>A 4 2 0 Zeca Pagodinho</div></div>	(b)
-----	--	-----

Figura 02: (a) Impressão do mês de Fevereiro com os respectivos eventos associados
(b) Eventos do mês de Fevereiro provenientes de um possível arquivo de eventos

2.4. Impressão em arquivo no formato pdf – TP 3

No caso da impressão do calendário no formato pdf, deve-se definir uma interface para as opções do cabeçalho:

- Mês e ano centrado na folha, ajustado à esquerda ou à direita;
- Mês anterior/posterior presente ou não e local no cabeçalho;
- Algum tipo de fundo para sábados e domingos;
- Ícones associados a aniversários, feriados e viagens;

Após definir a interface para as opções do cabeçalho, deve-se gerar um arquivo no formato .pdf cujo o conteúdo seja o calendário impresso do mês escolhido (vide o *item* 2.3).

2.5. Consulta eventos – TP 3

Considerando um determinado arquivo de eventos, deseja-se fazer dois tipos de operações:

- Ordenar e imprimir o arquivo por:
 - Evento usando a seguinte ordem: A (aniversário), F (feriado), V (viagem) e O (outros) considerando a ordem cronológica (da data mais antiga para a mais recente);
 - Ordem cronológica (da data mais antiga para a mais recente) e, em cada dia, por evento usando a seguinte ordem: A (aniversário), F (feriado), V (viagem) e O (outros);
- Pesquisar o arquivo por:
 - Tipo de evento: A (aniversário), F (feriado), V (viagem) e O (outros);
 - Uma data específica;

A ordenação, impressão e o resultado da pesquisa do arquivo de eventos devem ser exibidos na saída padrão de dados (monitor).

Ao realizar esse trabalho, espera-se elaborar um programa que, a partir da entrada de um conjunto de eventos e da seleção de uma série de opções, gere como saídas: (1) uma lista de dias e eventos associados de um determinado mês (TP 2 e 3 – vide *Figura 02*); (2) um arquivo no formato .pdf cujo o conteúdo seja um calendário com o mês escolhido, com interface de cabeçalho previamente definida (TP 3); (3) o arquivo de eventos ordenado e impresso na saída de dados padrão (monitor) de acordo com as opções escolhidas e um campo exibindo possibilidades de pesquisa nos dados do arquivo (TP 3).

As funcionalidades implementadas neste trabalho vão de encontro com o conteúdo das aulas ministradas na disciplina de Algoritmos e Estruturas de Dados II (AEDs II), à medida em que, exploram o conhecimento em algoritmos de pesquisa e ordenação, o uso de listas, a divisão de um problema em problemas menores e a criação de funções para a solução de cada um destes problemas (conceito de TAD e abstração). Além disso, esse trabalho prático explora os conceitos estudados em análise de complexidade de um algoritmo e estratégias de resolução de um problema, já que, a escolha da implementação de um algoritmo em detrimento de outro para a resolução de um problema está diretamente relacionada com o custo computacional que cada algoritmo possui. Ou seja, com a execução deste trabalho foi possível aplicar todos os conhecimentos aprendidos teoricamente na disciplina de AEDs II.

3. Implementação

3.1. Metodologia

TP 2

A primeira etapa para a execução do programa foi o recebimento dos dados de entrada e a leitura do arquivo de eventos. Para usar os dados do arquivo de eventos, foram usadas as funções `AbreArquivo` para abrir o arquivo de eventos e a função `CriaVetor` para ler os dados do arquivo e armazená-los em uma lista encadeada (essas funções e todas as outras utilizadas estão devidamente detalhadas e explicadas no *sub-item: 3.2- Funções*). As funções contidas no programa foram usadas no arquivo `main.c`, implementadas no `tp2.c` e têm seus cabeçalhos no arquivo `tp2.h`. De acordo com o que foi descrito no *item: 2. Objetivos e Funcionalidades*, era necessário implementar a funcionalidade da interface com o usuário (*sub-item 2.1*). Para isso, foi necessário montar uma estrutura que recebesse esses dados, de modo a checar as preferências para a impressão do usuário. A *Figura 03* abaixo mostra essa etapa:

```
15
16     printf("Digite o nome do arquivo: ");
17     scanf("%s", nome_arquivo);
18     CriaVetor(&T);
19     // Montagem do arquivo de eventos:
20     AbreArquivo(&T, nome_arquivo);
21     //ImprimeArquivo(&T);
22
23     // Entrada de Dados:
24     printf("Digite o mes do Calendario[1 a 12]: ");
25     scanf("%d", &mes_opcao);
26     printf("Digite o ano do calendario[1900 a 2999]: ");
27     scanf("%d", &ano_opcao);
28     printf("Escolha o idioma[0 -> Pt | 1 -> In]: ");
29     scanf("%d", &idioma_opcao);
30     TrataDadosEntrada(idioma_opcao, ano_opcao, qtdias_mes);
31     printf("\n\n");
32
```

Figura 03: Aquisição dos dados de entrada e leitura do arquivo de eventos

Após a aquisição das preferências do usuário e a leitura e armazenamento dos dados contidos no arquivo de eventos, foi necessário fazer o tratamento dos dados de entrada, para isso, utilizou-se a função: `TrataDadosEntrada`. Essa função confere se o ano é bissexto ou não, caso o ano seja bissexto, ela altera a quantidade de dias do segundo mês (Fevereiro) para 29.

Após a aquisição dos dados de entrada e o tratamento destes dados, fez-se necessário a exibição da saída de dados. Então, o próximo passo foi implementar a funcionalidade da impressão no monitor uma lista dos dias do mês em relação aos dias da semana de acordo com as opções escolhidas pelo usuário (escolha do mês, do ano e do idioma de visualização - conforme descrito nos *sub-ítems 1 e 2*). Para isso, foi necessário descobrir a relação entre o dia do mês e o respectivo dia da semana, ou seja, o primeiro dia do mês de um determinado ano corresponde a um dia da semana

(por exemplo, o primeiro dia do mês de Fevereiro/2019 foi uma sexta-feira). Essa relação entre o dia referente ao mês e o seu respectivo dia referente à semana foi determinada a partir da função: DiaSemana. Para que os eventos do mês escolhido pelo usuário contidos no arquivo de eventos apareçam na lista a ser impressa, foi necessário usar a função: ProcuraEvento. Assim, por meio das funções: DiaSemana e ProcuraEvento, foi possível imprimir no monitor uma lista com os dias de um mês e seus respectivos dias da semana com os eventos associados a cada dia (no formato mostrado na *Figura 02*) de acordo com as escolhas de interface do usuário. Essa implementação é mostrada na *Figura 04* abaixo.

```

33 // Saida de Dados: Portugues
34 if(idioma_opcao == 0){
35     printf("%s %d\n", mes[mes_opcao - 1], ano_opcao);
36     printf("Quantidade de dias: %d\n", qtdias_mes[mes_opcao - 1]);
37     printf("\n");
38     for(i = 1; i <= qtdias_mes[mes_opcao - 1]; i++){
39         resultado = DiaSemana(i, mes_opcao, ano_opcao);
40         printf("%d %s\n", i, semana[resultado]);
41         ProcuraEvento(&T, i, mes_opcao, ano_opcao);
42     }
43 }
44 // Saida de Dados: Ingles
45 else{
46     printf("%s %d\n", month[mes_opcao - 1], ano_opcao);
47     printf("Number of days: %d\n", qtdias_mes[mes_opcao - 1]);
48     printf("\n");
49     for(i = 1; i <= qtdias_mes[mes_opcao - 1]; i++){
50         resultado = DiaSemana(i, mes_opcao, ano_opcao);
51         printf("%d %s\n", i, week[resultado]);
52         ProcuraEvento(&T, i, mes_opcao, ano_opcao);
53     }
54 }
55
56 }

```

Figura 04: Saída de dados no monitor em formato de uma lista

TP 3

Conforme dito no item **2. Objetivos e Funcionalidades**, o TP 3 é uma extensão do TP 2, porém com duas funcionalidades a mais (*sub-ítems 2.4 e 2.5*). A funcionalidade da impressão do calendário em um arquivo com formato .pdf não pôde ser implementada. Entre as diversas dificuldades encontradas, as principais foram: problemas relacionados ao uso da biblioteca Haru e incompatibilidades inerentes a diferenças de ambientes de programação encontrados em materiais da internet e os ambientes de computação que estavam ao nosso alcance.

A funcionalidade: Consulta Eventos (sub-item 2.5) foi implementada mediante o uso das funções: GuardaEventos, TestaEvento, Atribui, Ordena, ImprimeOrdena e PesquisaArquivo. Utilizando essas funções (que serão devidamente explicadas no item 3.2), implementou-se a funcionalidade descrita de forma satisfatória.

3.2. Funções e análise de complexidade

As funções citadas no tópico 3.1. Metodologia, serão explicadas e detalhadas nessa seção, assim como outras funções que não foram citadas. É necessário destacar que as funções foram declaradas em um arquivo de cabeçalho chamado tp2.h, foram implementadas em um arquivo chamado tp2.c e foram chamadas (usadas) pela função main{} em um arquivo chamado main.c.

As funções implementadas no arquivo tp2.c são:

TP 2

- CriaVetor(Tabela *T):

```
6 void CriaVetor(Tabela *T){
7     int i;
8     T->dias = malloc(32 * sizeof(Evento));
9     //Implementa as celulas cabeca:
10    for(i = 0; i < 32; i++)
11        T->dias[i].prox = (Apontador)malloc(sizeof(Evento));
12 }
```

Figura 05: Implementação da função CriaVetor

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

- TrataDadosEntrada(int op_idioma, int op_ano, int *qtdias_mes):

```
14 void TrataDadosEntrada(int op_idioma, int op_ano, int *qtdias_mes){
15     // Checo se o ano e bissexto:
16     if( (op_ano % 4 == 0 && op_ano % 100 != 0) ||
17         (op_ano % 4 == 0 && op_ano % 100 == 0 && op_ano % 400 == 0) ){
18         qtdias_mes[1] = 29;
19     }
20 }
```

Figura 06: Implementação da função TrataDadosEntrada

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(1)$.

- DiaSemana(int dia, int mes, int ano):

```
22 int DiaSemana(int dia, int mes, int ano){
23     int k, resultado;
24     if(mes == 1){
25         mes = 13;
26         ano--;
27     } else if(mes == 2){
28         mes = 14;
29         ano--;
30     }
31     k = dia + (2 * mes) + (3*(mes + 1) / 5) + ano + (ano / 4) - (ano / 100) + (ano / 400) + 2;
32     resultado = k % 7;
33     return resultado;
34 }
```

Figura 07: Implementação da função DiaSemana

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(1)$.

- `InsererTabela(Tabela *T, char evento, int dia, int mes, int ano, char *descr_evento):`

```
37 void InsererTabela(Tabela *T, char evento, int dia, int mes, int ano, char *descr_evento){
38     T->Primeiro = T->dias[dia].prox;
39     T->Ultimo = T->Primeiro;
40     while(T->Ultimo->prox != NULL)
41         T->Ultimo = T->Ultimo->prox;
42     // Criacao da Celula e insercao dos dados:
43     T->Ultimo->prox= (Apontador)malloc(sizeof(Evento));
44     T->Ultimo->evento = evento;
45     T->Ultimo->dia = dia;
46     T->Ultimo->mes = mes;
47     T->Ultimo->ano = ano;
48     strcpy(T->Ultimo->descr_evento, descr_evento);
49 }
```

Figura 08: Implementação da função InsererTabela

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

- `AbreArquivo(Tabela *T, char *nome_arq):`

```
51 void AbreArquivo(Tabela *T, char *nome_arq){
52     FILE *p_arq;
53     int dia, mes, ano;
54     char evento, descr[25];
55     p_arq = fopen(nome_arq, "r+");
56     //Arquivo nao encontrado:
57     if(p_arq == NULL){
58         printf("Arquivo nao encontrado. \n");
59         exit(1);
60     }
61     //Arquivo encontrado:
62     else{
63         // MUDAR LINHA DO WHILE!!!
64         while(!feof(p_arq)){
65             fscanf(p_arq, "%c %d %d %d", &evento, &dia, &mes, &ano);
66             fgets(descr, 40, p_arq);
67             InsererTabela(T, evento, dia, mes, ano, descr);
68         }
69     }
70 }
```

Figura 09: Implementação da função AbreArquivo

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

- ImprimeArquivo(Tabela *T):

```

72 void ImprimeArquivo(Tabela *T){
73     Apontador aux;
74     int i;
75     printf("Começou a imprimir: \n");
76     for(i = 0; i < 32; i++){
77         T->Primeiro = T->dias[i].prox;
78         T->Ultimo = T->Primeiro;
79         while(T->Ultimo->prox != NULL){
80             printf("%c %d %d %d %s", T->Ultimo->evento, T->Ultimo->dia, T->Ultimo->mes, T->Ultimo->ano, T->Ultimo->descr_evento);
81             T->Ultimo = T->Ultimo->prox;
82         }
83     }
84     printf("\n");
85 }

```

Figura 10: Implementação da função ImprimeArquivo

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n^2)$.

- ProcuraEvento(Tabela *T, int dia, int mes, int ano):

```

87 void ProcuraEvento(Tabela *T, int dia, int mes, int ano){
88     Apontador aux1, aux2;
89     // Procura na lista encadeada os eventos salvos com o dia correspondente:
90     aux1 = T->dias[dia].prox;
91     while(aux1->prox != NULL){
92         // Nenhum zero:
93         if(aux1->mes == mes && aux1->ano == ano){
94             if(aux1->evento == '0')
95                 printf("\t%s", aux1->descr_evento);
96             else
97                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
98         }
99         // Dados com ano igual a zero:
100         else if(aux1->mes == mes && aux1->ano == 0){
101             if(aux1->evento == '0')
102                 printf("\t%s", aux1->descr_evento);
103             else
104                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
105         }
106         // Dados com mes igual a zero:
107         else if(aux1->mes == 0 && aux1->ano == ano){
108             if(aux1->evento == '0')
109                 printf("\t%s", aux1->descr_evento);
110             else
111                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
112         }
113         aux1 = aux1->prox;
114     }
115     // Procura na lista encadeada os eventos salvos com dia zero:
116     aux2 = T->dias[0].prox;
117     while(aux2->prox != NULL){
118         if(aux2->mes == mes && aux2->ano == ano){
119             if(aux2->evento == '0')
120                 printf("\t%s", aux2->descr_evento);
121             else
122                 printf("\t%c %s", aux2->evento, aux2->descr_evento);
123         }
124         aux2 = aux2->prox;
125     }
126 }

```

Figura 11: Implementação da função ProcuraEvento

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

TP 3

- *GuardaEventos(char *eventos);*

```
58 void GuardaEventos(char *eventos){
59     int n = 0, i = 0;
60     printf("Digite quantos eventos deseja guardar[1 a 4]: ");
61     scanf("%d", &n);
62     printf("Digite os eventos que voce deseja guardar de maneira espacada(Ex: A O F): ");
63     for(i = 0; i < n; i++)
64         scanf(" %c", &eventos[i]);
65     printf("Eventos digitados: ");
66     for(i = 0; i < n; i++)
67         printf("%c ", eventos[i]);
68     printf("\n");
69 }
70
```

Figura 11: Implementação da função GuardaEventos

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

- *TestaEvento(char *eventos, char evento):*

```
73 int TestaEvento(char *eventos, char evento){
74     int i;
75     for(i = 0; i < 5; i++){
76         if(evento == eventos[i]){
77             return 1;
78             break;
79         }
80     }
81     return 0;
82 }
```

Figura 12: Implementação da função TestaEvento

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

- *Atribui(Apontador a, Apontador b):*

```
174 void Atribui(Apontador a, Apontador b) {  
175     //rintf("A: %c B: %c", a->evento, b->evento);  
176     a->evento = b->evento;  
177     a->dia = b->dia;  
178     a->mes = b->mes;  
179     a->ano = b->ano;  
180     strcpy(a->descr_evento, b->descr_evento);  
181 }
```

Figura 13: Implementação da função Atribui

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(1)$.

- *Ordena(Tabela *O, int opcao):*

```

185 void Ordena(Tabela *O, int opcao){
186     Apontador i;
187     Apontador novo;
188     Apontador auxl;
189     // Ordenacao por Eventos:
190     if(opcao == 1){
191         novo = (Apontador)malloc(sizeof(Evento));
192         i = O->Primeiro->prox;
193         auxl = i;
194         while(i->prox != NULL){
195             auxl = i;
196             Atribui(novo, auxl);
197             while(auxl->prox != NULL){
198                 if(auxl->evento < i->evento){
199                     Atribui(novo, auxl);
200                     Atribui(auxl, i);
201                     Atribui(i, novo);
202                 }
203                 else if(auxl->evento == i->evento){
204                     if(auxl->ano < i->ano){
205                         Atribui(novo, auxl);
206                         Atribui(auxl, i);
207                         Atribui(i, novo);
208                     }
209                     else if(auxl->ano == i->ano){
210                         if(auxl->mes < i->mes){
211                             Atribui(novo, auxl);
212                             Atribui(auxl, i);
213                             Atribui(i, novo);
214                         }
215                         else if(auxl->mes == i->mes){
216                             if(auxl->dia < i->dia){
217                                 Atribui(novo, auxl);
218                                 Atribui(auxl, i);
219                                 Atribui(i, novo);
220                             }
221                         }
222                     }
223                 }
224                 auxl = auxl->prox;
225             }
226             i = i->prox;
227         }
228     }
229     // Ordenacao por Data cronologicas:
230     else if(opcao == 2){
231         novo = (Apontador)malloc(sizeof(Evento));
232         i = O->Primeiro->prox;
233         auxl = i;
234         while(i->prox != NULL){
235             auxl = i;
236             Atribui(novo, auxl);
237             while(auxl->prox != NULL){
238                 if(auxl->ano < i->ano){
239                     Atribui(novo, auxl);
240                     Atribui(auxl, i);
241                     Atribui(i, novo);
242                 }
243                 else if(auxl->ano == i->ano){
244                     if(auxl->mes < i->mes){
245                         Atribui(novo, auxl);
246                         Atribui(auxl, i);
247                         Atribui(i, novo);
248                     }
249                     else if(auxl->mes == i->mes){
250                         if(auxl->dia < i->dia){
251                             Atribui(novo, auxl);
252                             Atribui(auxl, i);
253                             Atribui(i, novo);
254                         }
255                     }
256                 }
257                 auxl = auxl->prox;
258             }
259             i = i->prox;
260         }
261     }
262 }

```

Figura 14: Implementação da função Ordena

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n^2)$.

- *ImprimeOrdena(Tabela *T, Tabela *O, int opcao)*

```
265 void ImprimeOrdena(Tabela *T, Tabela *O, int opcao){
266     Apontador novo;
267     Apontador aux;
268     int i;
269     // Testa opcao digitada:
270     if(opcao != 1 && opcao != 2){
271         printf("Opcao nao disponivel.\n");
272         exit(1);
273     }
274     // Criacao da nova lista encadeada:
275     O->Primeiro = (Apontador)malloc(sizeof(Evento));
276     O->Ultimo = O->Primeiro;
277     O->Primeiro->prox = NULL;
278     // Ordena por eventos:
279     for(i = 0; i < 32; i++){
280         T->Primeiro = T->dias[i].prox;
281         T->Ultimo = T->Primeiro;
282         while(T->Ultimo->prox != NULL){
283             novo = (Apontador)malloc(sizeof(Evento));
284             novo->evento = T->Ultimo->evento;
285             novo->dia = T->Ultimo->dia;
286             novo->mes = T->Ultimo->mes;
287             novo->ano = T->Ultimo->ano;
288             strcpy(novo->descr_evento, T->Ultimo->descr_evento);
289             O->Ultimo->prox = novo;
290             O->Ultimo = O->Ultimo->prox;
291             T->Ultimo = T->Ultimo->prox;
292         }
293     }
294     Ordena(O, opcao);
295     aux = O->Primeiro->prox;
296     while(aux->prox != NULL){
297         printf("\t%c %d %d %d \t%s", aux->evento, aux->dia, aux->mes, aux->ano, aux->descr_evento);
298         aux = aux->prox;
299     }
300     printf("\n");
301 }
302 }
```

Figura 15: Implementação da função ImprimeOrdena

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n^2)$.

- *PesquisaArquivo(Tabela *T, int opcao)*

```

305 void PesquisaArquivo(Tabela *T, int opcao){
306     int i;
307     char evento, evento2;
308     int dia, mes, ano;
309     if(opcao == 1){
310         printf("Por favor, digite o tipo de evento buscado(A, F, V, O): ");
311         scanf(" %c", &evento);
312         printf("Evento digitado: %c\n\n", evento);
313         for(i = 0; i < 32; i++){
314             T->Primeiro = T->dias[i].prox;
315             T->Ultimo = T->Primeiro;
316             while(T->Ultimo->prox != NULL){
317                 if(evento == T->Ultimo->evento)
318                     printf("\t%c %s", T->Ultimo->evento, T->Ultimo->descr_evento);
319                 T->Ultimo = T->Ultimo->prox;
320             }
321         }
322         printf("\n");
323     }
324     else if(opcao == 2){
325         printf("Por favor, digite a data buscada(dd/mm/aaaa): ");
326         scanf("%d/%d/%d", &dia, &mes, &ano);
327         printf("Data digitada: %d/%d/%d\n\n", dia, mes, ano);
328         ProcuraEvento(T, dia, mes, ano);
329         printf("\n");
330     }
331     else{
332         printf("Opcao nao disponivel.\n");
333         exit(1);
334     }
335 }

```

Figura 16: Implementação da função PesquisaArquivo

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n^2)$.

- *ProcuraEvento(Tabela *T, int dia, int mes, int ano):*

```

131 void ProcuraEvento(Tabela *T, int dia, int mes, int ano){
132     Apontador aux1, aux2;
133     // Procura na lista encadeada os eventos salvos com o dia correspondente:
134     aux1 = T->dias[dia].prox;
135     while(aux1->prox != NULL){
136         // Nenhum zero:
137         if(aux1->mes == mes && aux1->ano == ano){
138             if(aux1->evento == 'O')
139                 printf("\t%s", aux1->descr_evento);
140             else
141                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
142         }
143         // Dados com ano igual a zero:
144         else if(aux1->mes == mes && aux1->ano == 0){
145             if(aux1->evento == 'O')
146                 printf("\t%s", aux1->descr_evento);
147             else
148                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
149         }
150         // Dados com mes igual a zero:
151         else if(aux1->mes == 0 && aux1->ano == ano){
152             if(aux1->evento == 'O')
153                 printf("\t%s", aux1->descr_evento);
154             else
155                 printf("\t%c %s", aux1->evento, aux1->descr_evento);
156         }
157         aux1 = aux1->prox;
158     }
159     // Procura na lista encadeada os eventos salvos com dia zero:
160     aux2 = T->dias[0].prox;
161     while(aux2->prox != NULL){
162         if(aux2->mes == mes && aux2->ano == ano){
163             if(aux2->evento == 'O')
164                 printf("\t%s", aux2->descr_evento);
165             else
166                 printf("\t%c %s", aux2->evento, aux2->descr_evento);
167         }
168         aux2 = aux2->prox;
169     }

```

Figura 17: Implementação da função ProcuraEvento

A ordem de complexidade desta função, ou seja, o comportamento assintótico desta função pode ser representado por: $O(n)$.

Vou dar uma olhada em tudo P

4. Testes

- Teste de leitura de arquivo e montagem de calendário de acordo com as escolhas do TP2:

```
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP2
Arquivo Editar Ver Pesquisar Terminal Ajuda

giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP2$ gcc main.c tp2.c -o exe
giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP2$ ./exe
Digite o nome do arquivo: teste2.txt
Digite quantos eventos deseja guardar[1 a 4]: 4
Digite os eventos que voce deseja guardar de maneira espacada(Ex: A O F): A O F V
Eventos digitados: A O F V
Digite o mes do Calendario[1 a 12]: 12
Digite o ano do calendario[1900 a 2999]: 2019
Escolha o idioma[0 -> Pt | 1 -> In]: 1

December 2019
Number of days: 31

1 Sunday
2 Monday
3 Tuesday
4 Wednesday
5 Thursday
6 Friday
    Fim do semestre
7 Saturday
8 Sunday
    A V
9 Monday
10 Tuesday
11 Wednesday
12 Thursday
13 Friday
14 Saturday
15 Sunday
16 Monday
    V Cuiaba, MS
17 Tuesday
18 Wednesday
19 Thursday
20 Friday
21 Saturday
22 Sunday
23 Monday
24 Tuesday
25 Wednesday
    F Natal
    A Escola de Engenharia
26 Thursday
27 Friday
28 Saturday
29 Sunday
30 Monday
31 Tuesday
giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP2$
```

teste2.txt
~/Dropbox/Program...

Abrir Salvar

O 30 6 2019 Entrega dos TPs 2 e 3
A 15 05 2019 João Vitor
O 6 7 2019 Encerramento do 2 sem
V 0 10 2019 Paris, França
F 20 6 2019 Corpus Christi
F 25 12 2019 Natal
A 5 8 2019 Bruno de Oliveira
A 11 7 2019 David
A 30 07 2019 Diego de Oliveira
F 1 5 2019 Dia do Trabalho
V 0 1 2020 Porto Seguro, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Verão
O 30 10 2030 Caixinha
O 08 06 2020 Filho Pródigo (COLTEC)
A 24 05 2020 Audrey
F 12 10 2019 Dia das Crianças
A 08 12 2019 V
A 25 12 2019 Escola de Engenharia
V 16 12 2019 Cuiaba, MS
O 06 12 2019 Fim do semestre
O 30 6 2015 Reunião Administrativa
A 15 10 2024 Thor
O 6 10 2019 Seminário
V 22 10 2050 Lile, França
A 28 4 2019 Goku
F 25 12 2030 Natal
O 5 03 2019 Apresentação Oral
A 12 7 2020 Paula
A 02 2 2019 Alice
V 15 01 2026 Lisboa, Portugal
V 0 1 2029 Natal, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Verão
O 30 10 2036 Copa do Mundo
O 08 06 2019 Pelada
A 28 02 2019 Anna Victoria
F 15 10 2019 Dia da árvore
A 25 09 2019 Tia Geny
tação Largura da tabulação: 8 Lin 21, Col 29 INS

- Ordenação por tipos de eventos no TP3, não foi possível mostrar de todo o arquivo. Foram selecionados três tipos de eventos: A, F, O:

```
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3
Arquivo Editar Ver Pesquisar Terminal Ajuda

Voce escolheu ordenar e imprimir arquivo. Temos as seguintes operacoes:
1)- Ordem de eventos(A, F, V, O).
2)- Ordem cronologica.
Digite sua opcao: 1

A 2 2 2019 Alice
A 28 2 2019 Anna Victoria
A 1 4 2019 Tia Selma
A 28 4 2019 Goku
A 2 5 2019 Edna
A 15 5 2019 Joao Vitor
A 11 7 2019 David
A 30 7 2019 Diego de Oliveira
A 5 8 2019 Bruno de Oliveira
A 9 9 2019 Zeca
A 25 9 2019 Tia Geny
A 17 11 2019 Escola de Musica
A 22 11 2019 FAFICH
A 8 12 2019 Vó
A 25 12 2019 Escola de Engenharia
A 24 5 2020 Audrey
A 12 7 2020 Paula
A 12 8 2021 Lucas
A 15 10 2024 Thor
A 15 11 2025 Pingo
A 5 8 2028 Bruno de Oliveira Bastos
A 21 1 2031 Aline
A 8 7 2033 Tia Juliana
A 30 7 2033 Sofia
F 3 3 2019 Carnaval
F 5 3 2019 Cinzas
F 1 5 2019 Dia do Trabalho
F 20 6 2019 Corpus Chriti
F 12 10 2019 Dia das Crianas
F 15 10 2019 Dia da rvore
F 25 12 2019 Natal
F 25 9 2025 Aniversario de BH
F 25 12 2030 Natal
F 30 10 2034 Dia da Consciencia Negra
F 25 12 2041 Natal
F 29 6 2050 Dia do ndio
O 0 1 0 Veró
O 0 1 0 Veró
O 0 1 0 Veró
O 0 6 0 Inverno
O 0 6 0 Inverno
O 0 6 0 Inverno
O 30 6 2015 Reuniao Administrativa
O 7 12 2018 Seminarinho
O 3 2 2019 Ensaio para o Carnaval

O 30 6 2019 Entrega dos TPs 2 e 3
A 15 05 2019 Joao Vitor
O 6 7 2019 Encerramento do 2 sem
V 0 10 2019 Paris, Franca
F 20 6 2019 Corpus Chriti
F 25 12 2019 Natal
A 5 8 2019 Bruno de Oliveira
A 11 7 2019 David
A 30 07 2019 Diego de Oliveira
F 1 5 2019 Dia do Trabalho
V 0 1 2020 Porto Seguro, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Veró
O 30 10 2030 Caixinha
O 08 06 2020 Filho Prodigio (COLTEC)
A 24 05 2020 Audrey
F 12 10 2019 Dia das Crianas
A 08 12 2019 Vó
A 25 12 2019 Escola de Engenharia
V 16 12 2019 Cuiaba, MS
O 06 12 2019 Fim do semestre
O 30 6 2015 Reuniao Administrativa
A 15 10 2024 Thor
O 6 10 2019 Seminario
V 22 10 2050 Lille, Franca
A 28 4 2019 Goku
F 25 12 2030 Natal
O 5 03 2019 Apresentacao Oral
A 12 7 2020 Paula
A 02 2 2019 Alice
V 15 01 2026 Lisboa, Portugal
V 0 1 2029 Natal, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Veró
O 30 10 2036 Copa do Mundo
O 08 06 2019 Pelada
A 28 02 2019 Anna Victoria
F 15 10 2019 Dia da rvore
A 25 09 2019 Tia Geny
```


- Ordenação de eventos por ordem cronológica do TP3, dessa vez, foram escolhidos apenas dois tipos de eventos: O e F. Note que alguns eventos aparecem repetidos, mas não se tratam de erros, e sim, pelo fato de aparecer repetidamente no arquivo lido.

```

giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3
Arquivo Editar Ver Pesquisar Terminal Ajuda
Operacoes disponiveis:
1)- Imprimir calendario como lista.
2)- Imprimir calendario como arquivo pdf.
3)- Consulta de eventos.
Digite a sua opcao: 3

Bem vindo a consulta de arquivos!
Operacoes disponiveis:
1)- Ordenar e imprimir arquivo.
2)- Pesquisar arquivo.
Digite a sua opcao: 1

Voce escolheu ordenar e imprimir arquivo. Temos as seguintes opcoes:
1)- Ordem de eventos(A, F, V, O).
2)- Ordem cronologica.
Digite sua opcao: 2

O 0 1 0 Verão
O 0 1 0 Verão
O 0 1 0 Verão
O 0 6 0 Inverno
O 0 6 0 Inverno
O 0 6 0 Inverno
O 30 6 2015 Reunião Administrativa
O 7 12 2018 Seminarinho
O 3 2 2019 Ensaio para o Carnaval
F 3 3 2019 Carnaval
F 5 3 2019 Cinzas
O 5 3 2019 Apresentaão Oral
F 1 5 2019 Dia do Trabalho
O 8 6 2019 Pelada
F 20 6 2019 Corpus Chriti
O 29 6 2019 Curso de Inglês
O 30 6 2019 Entrega dos TPs 2 e 3
O 6 7 2019 Encerramento do 2 sem
O 6 10 2019 Seminario
F 12 10 2019 Dia das Crianas
F 15 10 2019 Dia da árvore
F 25 12 2019 Natal
O 8 6 2020 Filho Pródigo (COLTEC)
O 6 7 2022 Copa do Mundo: Catar
F 25 9 2025 Aniversário de BH
O 30 10 2030 Caixinha
F 25 12 2030 Natal
F 30 10 2034 Dia da Consciência Negra
O 30 10 2036 Copa do Mundo
O 21 2 2037 Entrega das Pizzas
O 30 10 2041 Campeão do Mundo
F 25 12 2041 Natal
F 29 6 2050 Dia do Índio

O 30 6 2019 Entrega dos TPs 2 e 3
A 15 05 2019 João Vitor
O 6 7 2019 Encerramento do 2 sem
V 0 10 2019 Paris, França
F 20 6 2019 Corpus Chriti
F 25 12 2019 Natal
A 5 8 2019 Bruno de Oliveira
A 11 7 2019 David
A 30 07 2019 Diego de Oliveira
F 1 5 2019 Dia do Trabalho
V 0 1 2020 Porto Seguro, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Verão
O 30 10 2030 Caixinha
O 08 06 2020 Filho Pródigo (COLTEC)
A 24 05 2020 Audrey
F 12 10 2019 Dia das Crianas
A 08 12 2019 Vó
A 25 12 2019 Escola de Engenharia
V 16 12 2019 Cuiaba, MS
O 06 12 2019 Fim do semestre
O 30 6 2015 Reunião Administrativa
A 15 10 2024 Thor
O 6 10 2019 Seminario
V 22 10 2050 Lile, França
A 28 4 2019 Goku
F 25 12 2030 Natal
O 5 03 2019 Apresentaão Oral
A 12 7 2020 Paula
A 02 2 2019 Alice
V 15 01 2026 Lisboa, Portugal
V 0 1 2029 Natal, BA, Brasil
O 0 6 0 Inverno
O 0 1 0 Verão
O 30 10 2036 Copa do Mundo
O 08 06 2019 Pelada
A 28 02 2019 Anna Victoria
F 15 10 2019 Dia da árvore
A 25 09 2019 Tia Geny

Largura da tabulação: 8 Lin 8, Col 19 INS

```

- Pesquisa de arquivo de acordo com o tipo de evento, nesse caso, V:

```

giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3
Arquivo Editar Ver Pesquisar Terminal Ajuda
giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ ./exe
Digite o nome do arquivo: teste2.txt
Digite quantos eventos deseja guardar[1 a 4]: 4
Digite os eventos que voce deseja guardar de maneira espacada(Ex: A O F): O F V A
Eventos digitados: O F V A
Operacoes disponiveis:
1)- Imprimir calendario como lista.
2)- Imprimir calendario como arquivo pdf.
3)- Consulta de eventos.
Digite a sua opcao: 3

Bem vindo a consulta de arquivos!
Operacoes disponiveis:
1)- Ordenar e imprimir arquivo.
2)- Pesquisar arquivo.
Digite a sua opcao: 2

Voce escolheu pesquisar arquivo. Temos as seguintes operacoes:
1)- Tipo de evento(A, F, V, O).
2)- Data especifica.
Digite sua opcao: 1
Por favor, digite o tipo de evento buscado(A, F, V, O): V
Evento digitado: V

    V Paris, Franca
    V Porto Seguro, BA, Brasil
    V Natal, BA, Brasil
    V Xangai, China
    V Duque de Caxias,RS,Brasil
    V Lisboa, Portugal
    V Londres, Inglaterra
    V Lille, Franca
    V Roma,Italia

giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$

```

- Pesquisa de eventos com a data referente a 25/09/2025. Perceba que foram selecionados apenas 2 tipos de eventos(F e V) no início. Logo, o evento com a data correspondente do tipo A não entra.

```

giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3
Arquivo Editar Ver Pesquisar Terminal Ajuda
giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ gcc main.c tp3.c -o exe
giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ ./exe
Digite o nome do arquivo: teste2.txt
Digite quantos eventos deseja guardar[1 a 4]: 2
Digite os eventos que voce deseja guardar de maneira espacada(Ex: A O F): F V
Eventos digitados: F V
Operacoes disponiveis:
1)- Imprimir calendario como lista.
2)- Imprimir calendario como arquivo pdf.
3)- Consulta de eventos.
Digite a sua opcao: 3

Bem vindo a consulta de arquivos!
Operacoes disponiveis:
1)- Ordenar e imprimir arquivo.
2)- Pesquisar arquivo.
Digite a sua opcao: 2

Voce escolheu pesquisar arquivo. Temos as seguintes operacoes:
1)- Tipo de evento(A, F, V, O).
2)- Data especifica.
Digite sua opcao: 2
Por favor, digite a data buscada(dd/mm/aaaa): 25/09/2025
Data digitada: 25/9/2025

F Aniversario de BH

giovanni@giovanni-Inspiron-7520:~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$

```

Evento	Tipo	Data	Evento	Tipo	Data
V	0	1	2029	Natal, BA, Brasil	
O	0	6	0	Inverno	
O	0	1	0	Verão	
O	30	10	2036	Copa do Mundo	
O	08	06	2019	Pelada	
A	28	02	2019	Anna Victoria	
F	15	10	2019	Dia da árvore	
A	25	09	2019	Tia Geny	
A	17	11	2019	Escola de Música	
O	21	02	2037	Entrega das Pizzas	
A	21	01	2031	Aline	
O	6	7	2022	Copa do Mundo: Catar	
V	0	10	2023	Xangai, China	
F	29	6	2050	Dia do Índio	
A	25	09	2025	Fernando Dias	
F	25	09	2025	Aniversário de BH	
A	5	8	2028	Bruno de Oliveira Bastos	
A	09	9	2019	Zeca	
A	30	07	2033	Sofia	
F	30	10	2034	Dia da Consciência Negra	
V	0	1	2020	Duque de Caxias, RS, Brasil	
O	0	6	0	Inverno	
O	0	1	0	Verão	
O	30	10	2041	Campeão do Mundo	
O	03	02	2019	Ensaio para o Carnaval	
F	03	3	2019	Carnaval	
F	5	3	2019	Cinzas	
A	08	07	2033	Tia Juliana	
A	22	11	2019	FAFICH	
O	30	8	2010	Reunião de Condomínio	
A	15	11	2025	Pingo	
O	07	12	2018	Seminarinho	
V	22	10	2050	Roma, Itália	
A	01	4	2019	Tia Selma	
F	25	12	2041	Natal	
O	29	6	2019	Curso de Inglês	
A	12	08	2021	Lucas	
A	02	5	2019	Edna	
V	19	01	2046	Londres, Inglaterra	

5. Conclusões

- Foi bastante interessante poder exercitar nossos conhecimentos aprendidos durante a disciplina no desenvolvimentos desses 2 trabalhos práticos. Pudemos desenvolver com folga todas as funcionalidades pedidas, com exceção da impressão do calendário em arquivo PDF.
- Tivemos vários contratempos (explicados logo a seguir), que dificultaram o desenvolvimento da impressão do calendário em arquivo pdf. Foi sugerido a utilização da biblioteca chamada “Haru”. Na documentação presente no repositório do GitHub, existem vários exemplos de aplicações, mas não fica clara como a biblioteca poderia seria utilizada.
- Baixamos a versão mais recente disponibilizada no repositório, e tentamos encontrar os arquivos que julgávamos úteis para a própria montagem (hpdf.c e hpdf.h). Todavia, quando fazíamos o include desses arquivos, e a eventual compilação do TP, sempre havia a necessidade de importar outros arquivos, como um efeito em cascata.


```
Atividades * - Terminal ▾ sáb, 29 de jun, 21:1  
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ gcc main.c tp3.c -o exe  
In file included from main.c:4:0:  
hpdf.h:20:10: fatal error: hpdf_config.h: Arquivo ou diretório inexistente  
#include "hpdf_config.h"  
~~~~~  
compilation terminated.  
In file included from tp3.c:5:0:  
hpdf.h:20:10: fatal error: hpdf_config.h: Arquivo ou diretório inexistente  
#include "hpdf_config.h"  
~~~~~  
compilation terminated.  
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ gcc main.c tp3.c -o exe  
In file included from main.c:4:0:  
hpdf.h:21:10: fatal error: hpdf_version.h: Arquivo ou diretório inexistente  
#include "hpdf_version.h"  
~~~~~  
compilation terminated.  
In file included from tp3.c:5:0:  
hpdf.h:21:10: fatal error: hpdf_version.h: Arquivo ou diretório inexistente  
#include "hpdf_version.h"  
~~~~~  
compilation terminated.  
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ gcc main.c tp3.c -o exe  
In file included from main.c:4:0:  
hpdf.h:87:10: fatal error: hpdf_consts.h: Arquivo ou diretório inexistente  
#include "hpdf_consts.h"  
~~~~~  
compilation terminated.  
In file included from tp3.c:5:0:  
hpdf.h:87:10: fatal error: hpdf_consts.h: Arquivo ou diretório inexistente  
#include "hpdf_consts.h"  
~~~~~  
compilation terminated.  
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$ gcc main.c tp3.c hpdf.c -o exe  
In file included from main.c:4:0:  
hpdf.h:87:10: fatal error: hpdf_consts.h: Arquivo ou diretório inexistente  
#include "hpdf_consts.h"  
~~~~~  
compilation terminated.  
In file included from tp3.c:5:0:  
hpdf.h:87:10: fatal error: hpdf_consts.h: Arquivo ou diretório inexistente  
#include "hpdf_consts.h"  
~~~~~  
compilation terminated.  
hpdf.c:18:10: fatal error: ruby.h: Arquivo ou diretório inexistente  
#include "ruby.h"  
~~~~~  
compilation terminated.  
giovanni@giovanni-Inspiron-7520: ~/Dropbox/Programming/Aeds/Aeds_02_2019_1/TP3$
```

- Além do problema associado a impressão do arquivo PDF, o aluno com o maior domínio (Giovanni), adoeceu no início do mês de julho devido a uma dengue, o que nos tirou quase 10 dias de tempo do trabalho prático (ainda possui o atestado, caso seja requerido). Tempo este que poderia ter usado utilizado para terminar os trabalhos da maneira ideal em tempo hábil.
- Diante de todos esses problemas e desafios, pedimos que valorizem o nosso esforço, e façam uma correção justa e coerente com todo o trabalho feito.

6. Referências Bibliográficas

https://pt.wikipedia.org/wiki/Ano_bissexto

<https://www.todamateria.com.br/historia-e-origem-do-calendario/>

<https://pt.wikipedia.org/wiki/Calend%C3%A1rio>

<https://www.calendarioano.com.br/calendario-gregoriano/>

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKFwjI-N7o34_jAhUHEbkGHaSXBnwQjhx6BAgBEAM&url=https%3A%2F%2Fpt.wikipedia.org%2Fwiki%2F2019&sig=AOyVaw3nUONW3lB05xdlz6ReCekW&ust=1561934023703627

https://www.youtube.com/watch?v=u7NAy_kDZ3A