

AGENDA ELETTRONICA

13.10.2022

GRUPPO 16

AUTORI

Elia Valenza
Massimo Manzulli

Analisi

Descrizione del sistema

Il programma, eseguito da un singolo utente, gestisce una agenda elettronica personale, dove è possibile registrare un **evento**, un **appuntamento** o un **promemoria**. Per ipotesi il software sarà attivo fino all'anno 2050.

Lo scopo del software è di mantenere la lista delle registrazioni inserite dall'utente, il quale all'occorrenza potrà visualizzarle, modificarle o eliminarle.

Per ogni registrazione, supponendo che ne esista almeno una, l'utente inserisce in input le seguenti informazioni:

- tipologia di registrazione (evento, appuntamento o promemoria)
- una descrizione generale
- la data (nel formato yyyy\mm\dd)
- l'orario di inizio
- l'orario di fine
- Contrassegno (indica se tale registrazione è considerata dall'utente importante)

1.2 Requisiti Funzionali

Le funzionalità implementate sono le seguenti:

- **AGGIUNTA – MODIFICA – CANCELLAZIONE**

L'utente ha la possibilità di inserire una nuova registrazione, modificare una già esistente o eliminarla.

Per modificare una registrazione il programma chiede all'utente di inserire la data della registrazione, successivamente verranno fatte visualizzare tutte quelle presenti nella data indicata, con i relativi codici univoci. Infine, l'utente inserendo il codice della registrazione di interesse, potrà modificarne tutti i campi; nel caso in cui nella data inserita non esista alcuna registrazione, verrà comunicato che non esiste registrazione nella data indicata.

Per eliminare una registrazione il procedimento è il medesimo, soltanto che una volta inserito il codice della registrazione quest'ultima verrà eliminata, altrimenti se nella data inserita non esiste alcuna registrazione, verrà comunicato che non esiste registrazione.

- **RICERCA REGISTRAZIONE**

L'utente per effettuare la ricerca delle registrazioni ha 3 modalità diverse, ovvero:

- **Ricerca per data** → avviene per matching esatto, così che il programma manda in output tutte le registrazioni presenti nella data ricercata. Ipotizzando che nella data ricercata non siano state inserite registrazioni, il programma comunicherà che non è presente alcuna registrazione in quella data.
- **Ricerca per tipologia** → il programma chiede all'utente quale tipologia di registrazione vuole visualizzare (eventi, promemoria o appuntamenti) mandando in output una lista delle registrazioni richieste. Ipotizzando che per la categoria ricercata non siano presenti registrazioni, il programma comunicherà che non è presente alcuna registrazione di quella tipologia.
- **Ricerca per oggetto** → basata sulla ricerca di parole\caratteri presenti nelle descrizioni delle registrazioni (funziona anche per sottostringa) mandando in output una lista delle registrazioni che presentano tali parole\caratteri nella descrizione, altrimenti un messaggio di ricerca non riuscita.

- **REGISTRAZIONE**

Il programma, quando l'utente richiede di inserire una nuova registrazione, verifica che non ce ne siano altre nell'ora indicata, tenendo conto anche della durata, quindi nel caso in cui la nuova registrazioni vada ad interferire nello svolgimento di un'altra registrazione già esistente, il programma manderà un messaggio in output il quale comunicherà che l'utente è impegnato in quel lasso di tempo (supponendo che sia possibile avere un'unica registrazione nell'ora designata e nel lasso di tempo indicato). In caso contrario il programma consentirà all'utente di inserire una nuova registrazione.

- **STATISTICHE**

Su richiesta dell'utente, il programma calcolerà una media degli appuntamenti riferita ad un certo intervallo di tempo, nel dettaglio inserendo in input una data d'inizio e una data finale, il programma manderà in output in media quanti appuntamenti l'utente ha giornalmente, e conterà quanti appuntamenti sono stati registrati come importanti in quel lasso di tempo. Ipotizzando che nel range di tempo inserito in input non ci siano appuntamenti, il programma manderà in output un messaggio, il quale comunicherà che non ci sono appuntamenti in programma.

- **FUNZIONALITA' DI ORDINAMENTO**

Il programma ordinerà le registrazioni in base alla data e l'ora, ma ipotizzando anche la presenza di registrazioni con un'importanza superiore, l'utente potrà decidere di mettere in primo piano tali registrazioni.

- **GESTIONE FUNZIONALITA' TRAMITE MENU' UTENTE**

Il programma gestisce tutte le funzionalità grazie ad un menù di selezione così composto:

- Gestione registrazioni
- Ricerca registrazione
- Statistiche
- Ordinamento registrazioni

DATI IN INPUT

- Tabella Data:

Anno
Mese
Giorno

-Tabella Orario:

Ora
Minuto

-Tabella Registrazione:

Codice
Data
Orario Inizio
Orario Fine
Tipologia (Appuntamento, Evento o Promemoria)
Descrizione
Contrassegno

- Codice registrazione per modificare o eliminare la registrazione
- Data registrazione da modificare
- Orario inizio registrazione da modificare
- Orario fine registrazione da modificare

- Tipologia registrazione da modificare
- Descrizione registrazione da modificare
- Contrassegno registrazione da modificare
- Data \ Tipologia \ Oggetto per ricercare una registrazione
- Data inizio e fine per calcolo statistica
- Menù per la scelta delle opzioni da utilizzare del programma.

DATI IN OUTPUT

- Messaggio che comunichi all'utente che nell'orario indicato non è possibile inserire una nuova registrazione poiché interferisce con un'altra già registrata.
- Messaggio che comunichi all'utente che nella data inserita non è presente alcuna registrazione da modificare/eliminare.
- Messaggio che comunichi che il codice inserito non appartiene a nessuna registrazione.
- Messaggio che comunichi all'utente che per la data o tipologia o oggetto ricercato non esiste alcuna registrazione nell'agenda elettronica
- Messaggio che comunichi che nel periodo di tempo indicato per la statistica sugli appuntamenti non esiste alcun appuntamento registrato.
- Visualizzazione registrazioni ricercate o ordinate
- Visualizzazione statistica su appuntamenti.

PROGRAMMAZIONE MODULARE

Il programma è stato sviluppato attraverso l'ausilio di un file e di diverse procedure e funzioni, ognuna delle quali ha dei specifici parametri e scopi per garantire l'efficacia del software.

La documentazione è stata prodotta tramite DOXYGEN

Per la realizzazione di questo programma sono stati utilizzati i seguenti algoritmi:

- Ricerca sequenziale di sicuro successo
- Ricerca sequenziale di non sicuro successo
- Ricerca lineare con sentinella
- Ricerca lineare esaustiva
- Selection Sort

2. Progettazione

2.1 Progettazione dei tipi di dato, delle strutture dati

NOME	TIPOLOGIA	DESCRIZIONE	TIPI / CAMPI / VALORI
Data	Struct	Tipo di dato definito per rappresentare le date	-giorno: unsigned short int; -mese: unsigned short int; -anno: unsigned short int
Ora	Struct	Tipo di dato definite per rappresentare gli orari.	-ora: unsigned short int, -minuti: unsigned short int
Registrazione	Struct	Tipo di dato definito per rappresentare le registrazioni	-codice: unsigned short int; -data: Data; - ora_inizio: Ora; - ora_fine: Ora; -tipologia: char[MAX_LUNG_TIPO]; - descrizione: char[MAX_LUNG_DESCR]; - contrassegno: unsigned short int;
MAX_LUNGHEZZA_TIPO	Costante	Costante utilizzata per rappresentare la massima lunghezza della stringa Tipologia	20
MAX_LUNGHEZZA_DESCRIZIONE	Costante	Costante utilizzata per rappresentare la massima lunghezza della stringa Descrizione	200

ANNO_INIZI O_S	Costante	Costante che indica l'inizio di attività del software	2022
ANNO_FINE_ S	Costante	Costante che indica la fine dell'attività del software	2050
GIORNI_ANN O_	Costante	Costante che indica il numero di giorni dell'anno	365
GIORNI_ANN O_B	Costante	Costante che indica i giorni dell'anno bisestile	366
GIORNI_MES E	Costante	Costante che indica i mesi con 31 giorni	31
GIORNI_MES E_2	Costante	Costante che indica i mesi con 30 giorni	30

GIORNI_MES E_F	Costante	Costante che indica i giorni del mese di febbraio	28
GIORNI_MES E_B	Costante	Costante che indica i giorni del mese di febbraio nell'anno bisestile	29
GIORNI_ORA _MINUTI	Costante	Costante che indica l'ora in minuti	60
input	Variabile	Variabile utilizzata per controllare che gli input inseriti rispettino il tipo dichiarato	Int
Scelta	Variabile	Variabile utilizzata per l'implementazione dello switch case	Char[1]
durata	Variabile	Variabile utilizzata per calcolare la durata	Float

importante	Array	Array utilizzato per far visualizzare "SI" se una registrazione è importante, "NO" altrimenti.	Char[3]
codice	Variabile	Variabile utilizzata per rappresentare il codice della registrazione che si vuole modificare o eliminare	Int > 0
registrazioni_file	File	File utilizzato per salvare i record di tipo Registrazione che rappresentano le registrazioni presenti nell'agenda elettronica	
registrazione_t	Struct	Struct temporanea contenente la registrazione utilizzata per le diverse funzionalità del programma	Registrazione { -codice: int; -data: Data; - ora_inizio: Ora; - ora_fine: Ora; -tipologia: char[MAX_LUNG_TIPO]; - descrizione: char[MAX_LUNG_DESCR]; - contrassegno: unsigned short int; }
n_registrazioni	Variabile	Variabile utilizzate per tenere traccia del numero di registrazioni	Int >= 0

controllo	Variabile	Variabile utilizzata per tenere traccia di situazioni verificate o non all'interno del programma	Int >= 0
controllo_codice	Variabile	Variabili utilizzate per controllare che il codice inserito sia valido	Int >= 0
controllo_esistenza_reg	Variabile	Variabile utilizzata per controllare che nella data inserita esista una registrazione	int
OrarioInizio_reg_file_in_min	Variabile	Variabile utilizzata per trasformare l'orario di inizio delle registrazioni presenti nel file in minuti	Unsigned short int
OrarioFine_reg_file_in_min	Variabile	Variabile utilizzata per trasformare l'orario di fine delle registrazioni presenti nel file in minuti	Unsigned short int
OrarioInizio_reg_utente_in_min	Variabile	Variabile utilizzata per trasformare l'orario di inizio della registrazione dell'utente in minuti	Unsigned short int

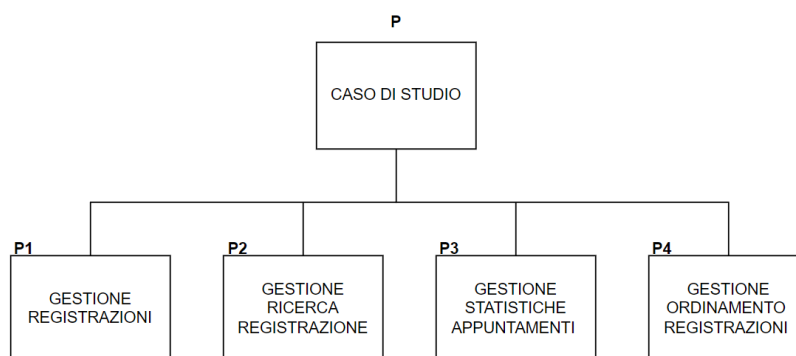
i,j,k	Variabili contatore	Variabili utilizzate come indici contatori	Unsigned short int
-------	------------------------	--	--------------------

2.2 Progettazione della soluzione prodotta

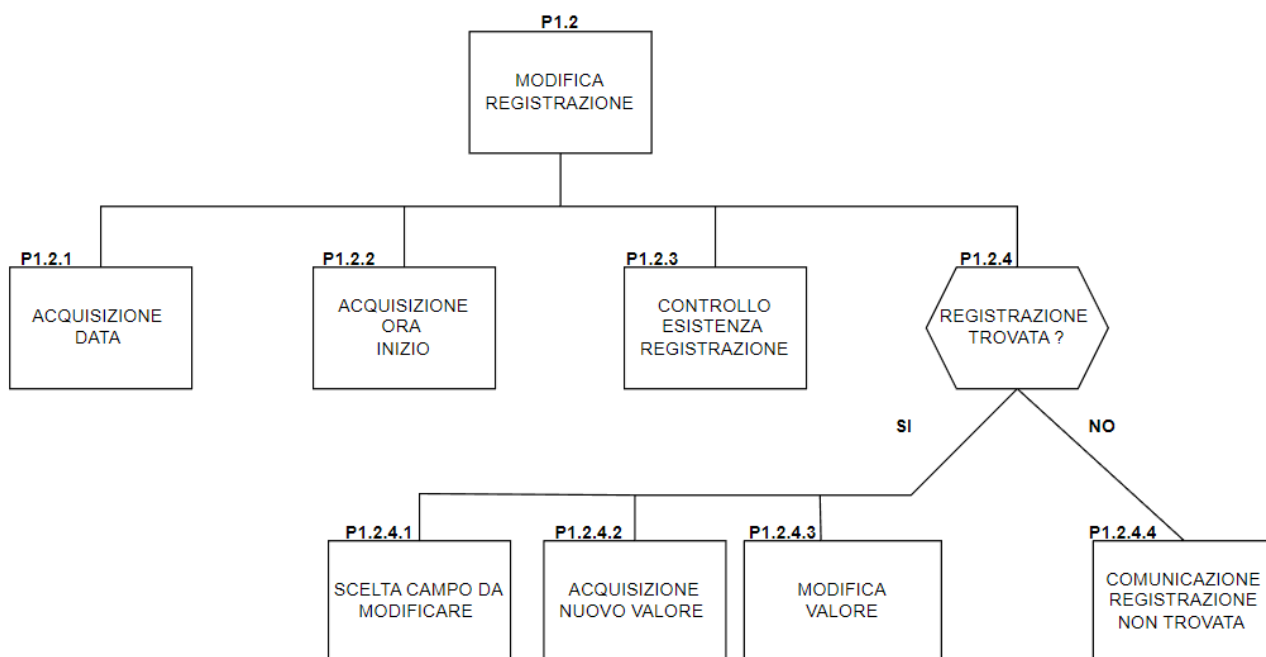
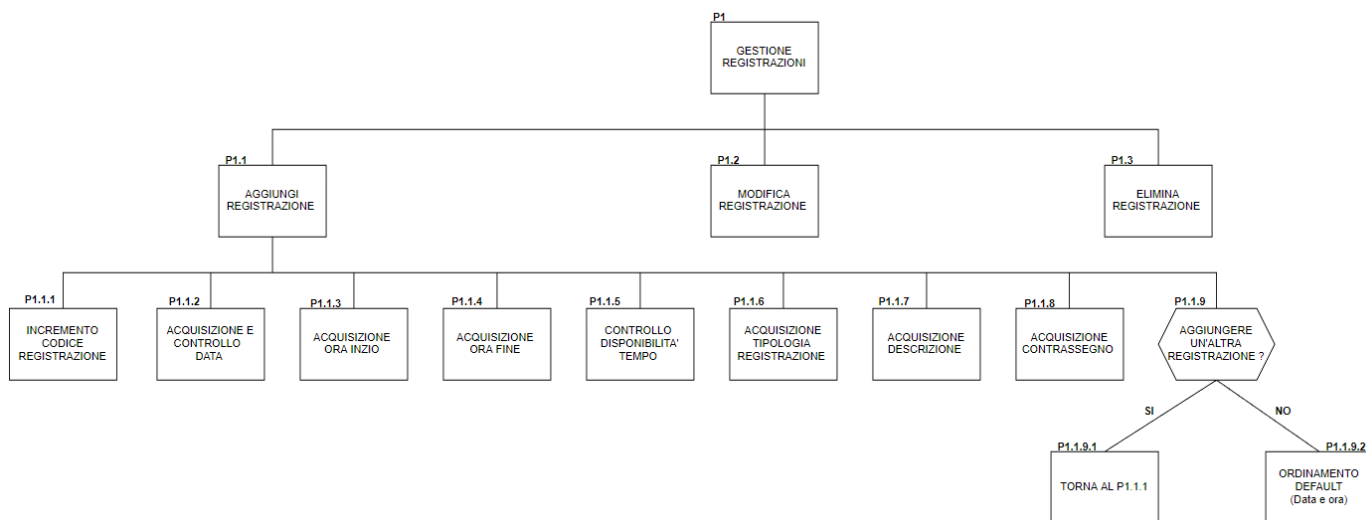
Per tale progettazione si è adottato un approccio **TOP – DOWN**.

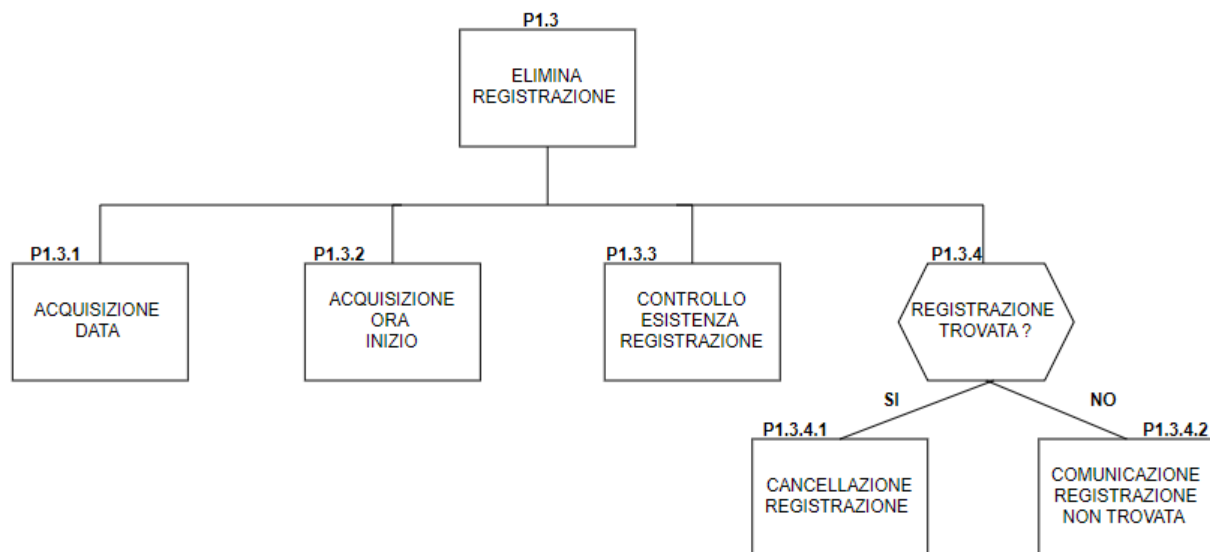
I principali sottoproblemi sono:

- Gestione Registrazione
- Gestione Ricerca Registrazione
- Gestione Statistiche Appuntamenti
- Gestione Ordinamento Registrazioni

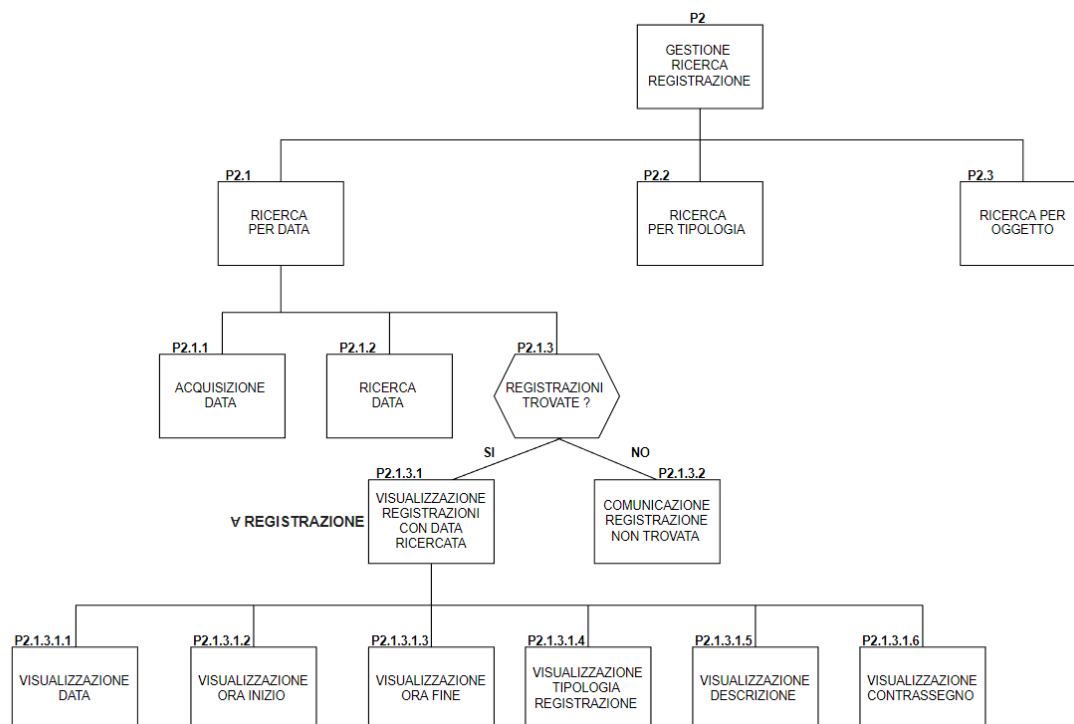


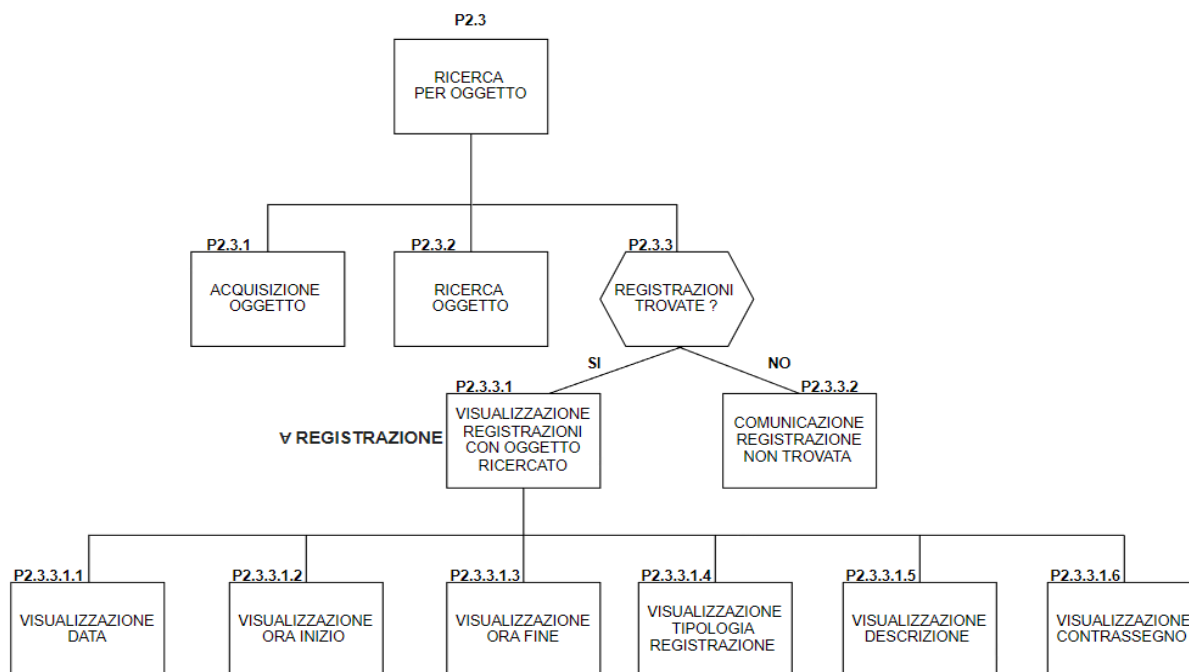
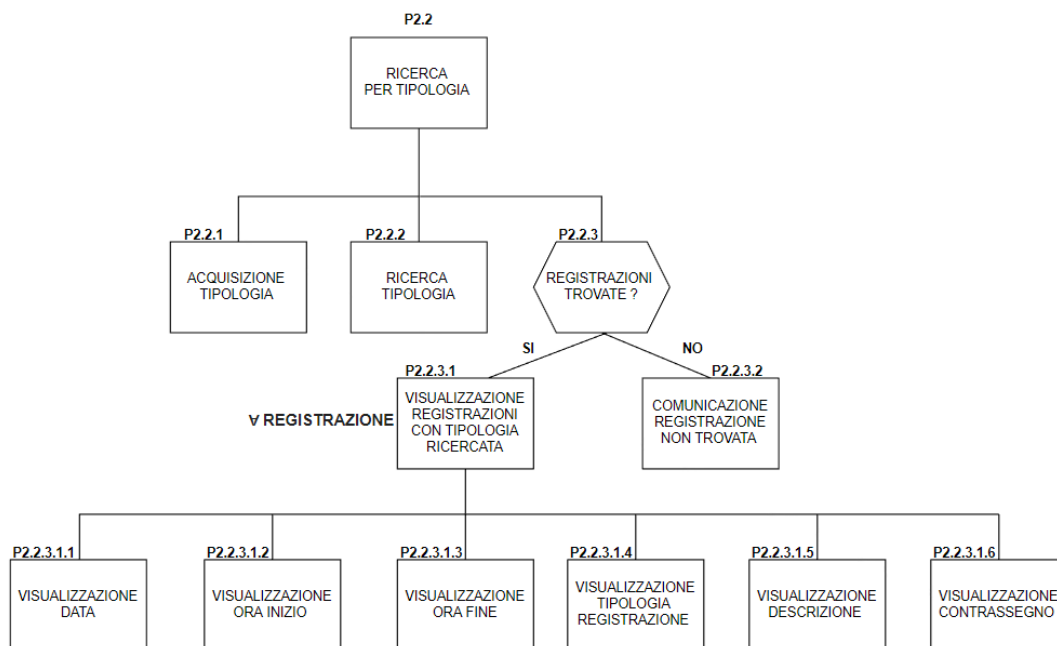
P1



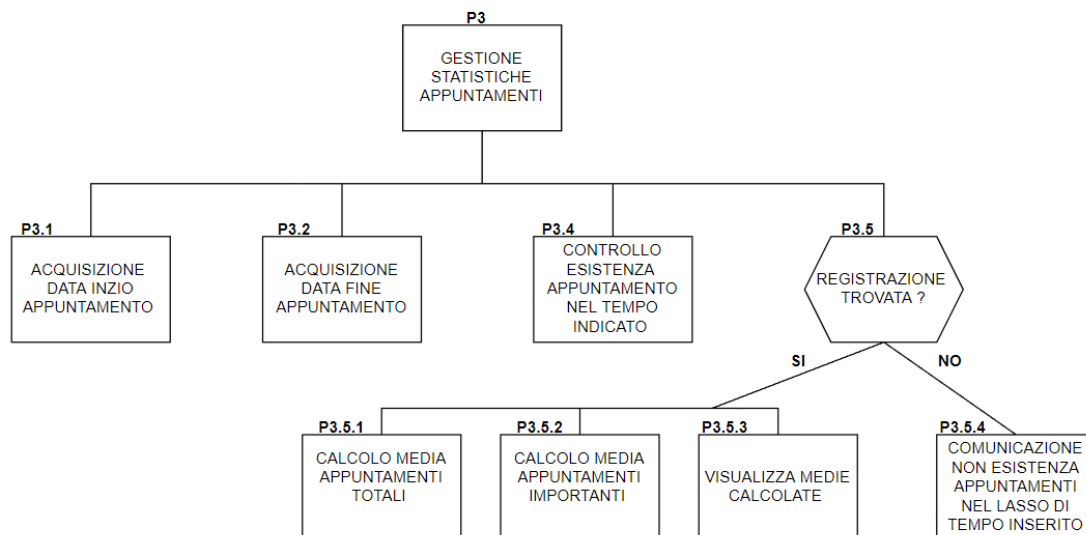


P2

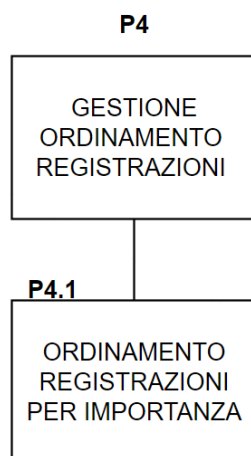




P3



P4



PSEUDOCODICE

Di seguito è riportato lo pseudocodice dei sottoproblemi principali, prodotti utilizzando il software Sublime Text.

MAIN

```
5  funzione main(void)
6
7      setvbuf(stdout, NULL, _IONBF,0);
8      scelta[1]:carattere;
9      scelta[0] <-- '0';
10
11     FILE *registrazioni_file;
12
13     Se registrazioni_file <-- apertura_file("Registrazioni.txt","rb+") == NULL ) allora    //Se non è possibile aprire il file
14         scrivi("ERRORE... Non è possibile aprire il file");                                // in modalità scrittura e lettura
15     Fine se
16
17     scrivi("*****AGENDA ELETTRONICA*****");
18
19     Finchè scelta[0] è uguale a funzione Menu_Principale() <-- diverso da '0';
20
21         switch(scelta[0])
22         Inizio
23
24             caso '1':
25                 chiamata della procedura Menu_Gestione(registrazioni_file:FILE);
26             Fine
27
28             caso '2':
29                 chiamata della procedura Menu_Ricerca(registrazioni_file:FILE);
30             Fine
31
32             caso '3':
33                 chiamata della procedura Statistica_Appuntamenti(registrazioni_file:FILE);
34             Fine
35
36             caso '4':
37                 chiamata della procedura Menu_Ordinamento(registrazioni_file:FILE);
38             Fine
39
40         Fine
41
42     Fine Finchè
43
44     chiusura_file(registrazioni_file);
45
46     ritorno 0;
47
48 Fine funzione
```




MENU_PRINCIPALE

```
5  funzione Menu_Principale();
6
7      scelta[1]:carattere;
8      scelta_corretta <-- 0;
9
10     Ripeti
11
12         scrivi("*****");
13         scrivi("*<0>  EXIT*");
14         scrivi("*<1>  GESTIONE REGISTRAZIONE*" );
15         scrivi("*<2>  RICERCA REGISTRAZIONI*");
16         scrivi("*<3>  STATISTICA APPUNTAMENTI*");
17         scrivi("*<4>  ORDINAMENTI REGISTRAZIONE *");
18         scrivi("*****");
19
20         scrivi("Inserisci la tua scelta --> ");
21         leggi(scelta);
22
23         scelta_corretta <-- funzione Scelta_Corretta_Menu(scelta:carattere);
24
25         se scelta_corretta è diverso da 1 allora
26             scrivi("Occhio! Hai inserito un valore non valido");
27         fine se
28
29     Finchè scelta_corretta è uguale a 0;
30
31     scrivi("%*^[^\n]"); //Utilizzato per pulire il buffer
32
33     ritorna scelta[0];
34
35 Fine funzione
```



INSERIMENTO_DATI_REGISTRAZIONE

```
5  funzione Inserimento_Dati_Registrazione(*registrazioni_file_p:FILE):Registrazione
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      controllo_tempo <-- 0;
9      controllo_esistenza_reg <-- 0;
10     tipologia <-- 0;
11     controllo_tipologia <-- 0;
12     contrassegno <-- 0;
13
14     scrivi("Inserimento registrazione");
15
16     Ripeti
17
18         procedura Acquisizione_Controllo_Data(&registrazione_t.data);
19
20         scrivi(" Inserisci l'orario di inizio nel formato h:m");
21         procedura Acquisizione_Controllo_Orario(&registrazione_t.orario_inizio);
22
23         scrivi(" Inserisci l'orario di fine nel formato h:m");
24         procedura Acquisizione_Controllo_Orario(&registrazione_t.orario_fine);
25
26         scrivi("Verifico la tua richiesta.....");
27
28         controllo_esistenza_reg <-- procedura Controllo_Esistenza_Registrazione(registrazione_t, registrazioni_file_p);
29
30         se controllo_esistenza_reg = 1 allora
31             scrivi("Registrazione già esistente");
32         altrimenti
33             controllo_tempo <-- 1;
34             scrivi("Richiesta accettata");
35         fine se
36
37     Finchè controllo_tempo = 0;
38
39     Ripeti
40
41         scrivi("Inserimento tipologia <1> EVENTO <2> APPUNTAMENTO <3> PROMEMORIA ----> ");
42         leggi("tipologia");
43
44         se tipologia = 1 allora
45             strcpy(registrazione_t.tipologia, "EVENTO");
46             controllo_tipologia = 0;
47
48         altrimenti se tipologia = 2 allora
49             strcpy(registrazione_t.tipologia, "APPUNTAMENTO");
50             controllo_tipologia = 0;
51
52         altrimenti se tipologia = 3 allora
53             strcpy(registrazione_t.tipologia, "PROMEMORIA");
54             controllo_tipologia = 0;
55
56         altrimenti
57             scrivi("Inserisci un valore valido");
58             controllo_tipologia <-- 1;
59
60         fine se
61
62     Finche controllo_tipologia = 1;
63
64     scrivi("Inserisci una descrizione di massimo 200 caratteri --> ");
65     fggest(registrazione_t.descrizione, 200, stdin);
66
67     scrivi("Inserire <1> per dichiarare la registrazione come importante -->");
68     leggi(contrassegno);
69
70     se contrassegno = 1 allora
71         registrazione_t.contrassegno <-- 1;
72     altrimenti
73         registrazione_t.contrassegno <-- 0;
74     fine se
75
76     restituisci registrazione_t
77
78     Fine funzione
```



MODIFICA_REGISTRAZIONE

```
5  procedura Modifica_Registrazione (*registrazioni_file_p:FILE)
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      controllo_esistenza_reg <-- 0;
9      codice <-- 0;
10     controllo_codice <-- 0;
11     input <-- 0;
12
13     scrivi("Modifica registrazione");
14
15     Ripeti
16
17         controllo_esistenza_reg <-- funzione Ricerca_Data(registrazioni_file_p:FILE) ;
18
19     Finchè controllo_esistenza_reg = 0;
20
21
22     Ripeti
23
24         scrivi("Inserisci il codice della registrazione da modificare --> ");
25         input <-- leggi(codice);
26         controllo_codice <-- funzione Controllo_Codice(registrazioni_file_p:FILE, codice:int);
27
28         se controllo_codice = 0 allora
29             scrivi("Il codice inserito non è associato ad alcuna registrazione");
30         altrimenti se input negato
31             scrivi("Inserire un valore valido");
32         fine se
33
34     Finchè controllo_codice = 0 o input negato;
35
36
37     registrazione_t <-- funzione Inserimento_Dati_Registrazione(registrazioni_file_p:FILE);
38
39     registrazione_t.codice <-- codice;
40
41     funzione posiziono_il_puntatore(registrazioni_file_p:FILE, (registrazione_t.codice-1):intero * funzione sizeof(registrazione_t:Registrazione), SEEK_SET);
42     funzione scrivo_sul_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
43
44     fine procedura
```

ELIMINAZIONE_REGISTRAZIONE

```

6  procedura Eliminazione_Registrazione(registrazione_file_p:FILE)
7
8      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"",0};
9      registrazione_r <-- {0,{0,0,0},{0,0},{0,0},"",0};
10     n_registrazioni <-- 0
11
12     posizione_il_puntatore(registrazioni_file_p:FILE, (registrazione_t.codice-1):intero * funzione sizeof(registrazione_t:Registrazione), SEEK_END);
13     scrivo_sul_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
14     n_registrazioni <-- registrazione_t.codice;
15     registrazione_a[n_registrazioni:intero]:Registrazione;
16
17     controllo_esistenza_reg <-- 0;
18     controllo_codice <-- 0;
19     codice <-- 0;
20     input <-- 0;
21     i <-- 0;
22     k <-- 0;
23
24     scrivi("Eliminazione registrazione");
25
26     Ripeti
27
28         controllo_esistenza_reg <-- funzione Ricerca_Data(registrazioni_file_p:FILE);
29
30     Finchè controllo_esistenza_reg = 0;
31
32     Ripeti
33
34         scrivi("Inserisci il codice della registrazione da voler eliminare --> ");
35         input <--- leggi(codice);
36
37         se input è negato allora
38             scrivi(inserire un valore valido);
39         fine se
40
41     Finchè input è negato;
42
43
44     se codice è diverso da -1 allora
45         controllo_codice <-- funzione Controllo_Codice(registrazioni_file_p:FILE, codice:int);
46
47         se controllo_codice è uguale a 1 allora //mettere fine se quando scivei che il codice non è associato
48             posizione_il_puntatore(registrazioni_file_p:FILE, (codice-1):intero * funzione sizeof(Registrazione), SEEK_SET);
49             scrivo_sul_file(&registrazione_r:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
50             riposiziono_puntatore_file_all_inizio(registrazioni_file_p:FILE);
51             leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
52
53             Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
54
55                 se registrazione_t.codice è maggiore di 0 allora
56                     i <-- i + 1;
57                     registrazione_t.codice = i;
58                     registrazione_a[i] = registrazione_t;
59                 fine se
60
61             funzione leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
62
63         Fine finchè
64
65         chiudo_il_file(registrazioni_file_p);
66
67         se registrazioni_file_p <-- apro_il_file(("Registrazioni.txt", "wb+") == NULL ) allora //Se non è possibile aprire il file
68             scrivi("ERRORE");
69
70         altrimenti
71
72             riposiziono_puntatore_file_all_inizio(registrazioni_file_p);
73
74             per k uguale a 0; k minore di n_registrazioni; incremento k
75
76                 registrazione_t <-- registrazione_a[k];
77                 posizione_il_puntatore(registrazioni_file_p:FILE, (registrazione_t.codice-1):intero * funzione sizeof(registrazione_t:Registrazione), SEEK_SET);
78                 scrivo_sul_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
79
80             fine
81
82             scrivi("Eliminazione avvenuta");
83
84         fine se
85
86     altrimenti
87
88         scrivi("Il codice inserito non è associato a nessuna registrazione");
89
90     fine se
91
92     fine se
93
94     fine procedura

```

RICERCA_DATA

```

5  funzione Ricerca_Data(registrazioni_file_p:FILE)
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      registrazione_p <-- {0};
9      controllo <-- 0;
10
11     procedura Acquisizione_Controllo_Data(&registrazione_p);
12
13     riposizione_puntatore_file_all_inizio(registrazioni_file_p);
14     funzione leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
15
16     Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
17
18         se registrazione_p.anno è uguale a registrazione_t.data.anno allora
19             se registrazione_p.mese è uguale a registrazione_t.data.mese allora
20                 se registrazione_p.giorno è uguale a registrazione_t.data.giorno allora
21                     procedura visualizzazione_file(registrazione_t:Registrazione);
22                     controllo <-- 1;
23
24                 fine se
25
26             fine se
27
28         fine se
29
30         leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
31
32     fine finchè
33
34     se controllo è uguale a 0 allora
35         scrivi("Non esiste registrazione nella data ricercata");
36     fine se
37
38     ritorno controllo;
39
40 fine funzione

```

RICERCA_TIPOLOGIA

```

5  procedura Ricerca_Tipologia(registrazione_t:FILE)
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      tipologia <-- 0;
9      controllo_tipologia <-- 0;
10     controllo <-- 0;
11     tipologia_c[MAX_LUNGHEZZA_TIPO];
12
13     Ripeti
14
15         scrivi("Inserimento tipologia <1> EVENTO <2> APPUNTAMENTO <3> PROMEMORIA --->");
16         leggi(tipologia);
17
18         se tipologia è uguale a 1 allora
19             strcpy(tipologia_c, "EVENTO"); //copia una stringa da una locazione ad un'altra
20
21         altrimenti se tipologia è uguale a 2 allora
22             strcpy(tipologia_c, "APPUNTAMENTO");
23
24         altrimenti se tipologia è uguale a 3 allora
25             strcpy(tipologia_c, "PROMEMORIA");
26
27         altrimenti
28             scrivi("Inserisci un valore valido");
29
30         fine se
31
32     Finchè controllo_tipologia è uguale 1;
33
34     riposizione_puntatore_file_all_inizio(registrazioni_file_p:FILE);
35     funzione leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
36
37     Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
38
39         se le stringhe tipologia_c e registrazione_t.tipologia sono uguali allora
40             procedura visualizzazione_file(registrazione_t);
41             controllo <-- 1;
42         fine se
43
44         leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
45
46     fine finchè
47
48     se controllo è uguale a 0 allora
49         scrivi("Non esistono registrazioni con la tipologia ricercata");
50     fine se
51
52 Fine procedura

```



RICERCA_OGGETTO

```
5  procedura Ricerca_Oggetto(registrazione_file_p:FILE);
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"", "",0};
8      controllo <-- 0;
9      oggetto_ric[MAX_LUNGHEZZA_DESCRIZIONE]:char;
10
11      scrivi("Ricerca per oggetto");
12      scrivi("N.B Vale anche per sottostringa");
13
14      riposizione_puntatore_file_all_inizio(registrazioni_file_p:FILE);
15      scrivi("Inserisci l'argomento da ricercare --> ");
16      leggi(oggetto_ric);
17      procedura stringa_in_minuscolo_descrizione(oggetto_ric);
18
19      leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
20
21      Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
22
23          stringa_in_minuscolo_descrizione(registrazione_t.descrizione);
24
25          se le stringhe registrazione_t.descrizione e oggetto_ric sono uguali allora
26              controllo <-- 1;
27
28              procedura visualizzazione_file(registrazione_t);
29          fine se
30
31          leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
32
33      Fine finchè
34
35      se controllo è uguale a 0 allora
36          scrivi("Non è stata trovata nessuna registrazione");
37      fine se
38
39      Fine procedura
```

STATISTICA_APPUNTAMENTI

```
3  procedura Statistica_Appuntamenti(registrazioni_file_p:FILE);
4
5      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"", "",0};
6
7      controllo, n_giorni, n_reg_comprese, n_reg_importanti, data_inizio_g, data_fine_g <-- 0;
8      media <-- 0;
9      data_inizio, data_fine <-- {0,0,0};
10
11      scrivi("Inserisci la data di inizio");
12      procedura Acquisizione_Controllo_Data(&data_inizio);
13      scrivi("Inserisci la data di fine");
14      procedura Acquisizione_Controllo_Data(&data_fine);
15
16      data_inizio_g <-- procedura Data_in_Giorni(data_inizio);
17      data_fine_g <-- procedura Data_in_Giorni(data_fine);
18
19      riposizione_puntatore_file_all_inizio(registrazioni_file_p:FILE);
20      leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
21
22
23      Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
24
25          n_giorni <-- procedura Data_in_Giorni(registrazione_t.data);
26
27          se n_giorni è compreso uguale tra data_inizio_g e data_fine_g allora
28              se confrontando le stringhe registrazione_t.tipologia e "APPUNTAMENTO" restituisce un valore diverso da NULL allora
29                  controllo <-- 1;
30                  n_reg_comprese <-- n_reg_comprese + 1;
31
32                  se registrazione_t.contrassegno è uguale a 1
33                      n_reg_importanti <-- n_reg_importanti + 1;
34
35                  fine se
36
37          fine se
38
39      fine se
40
41      leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
42
43      Fine finchè
44
45      se controllo è uguale a 1 allora
46          media <-- (float)n_reg_comprese / ((float)data_fine_g - (float)data_inizio_g);
47          scrivi("Media appuntamenti giornalieri --> " media);
48          scrivi("Gli appuntamenti importanti in questo periodo sono --> " n_reg_importanti);
49      fine se
50
51      Fine procedura
```



ORDINAMENTO_DATA

```
5  procedura Ordinamento_Data(registrazione_file_p:FILE)
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      data_in_giorni <-- 0;
9      n_registrazioni <-- 0;
10     importanza <-- 0;
11     i,j <-- 0;
12
13     posiziona_il_puntatore(registrazioni_file_p:FILE, (registrazione_t.codice-1):intero * funzione sizeof(registrazione_t), SEEK_END);
14     leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
15     n_registrazioni <-- registrazione_t.codice;
16
17     giorni[n_registrazioni]:interi;
18     registrazioni_app[n_registrazioni]:Registrazione;
19
20     riposiziona_puntatore_file_all_inizio(registrazioni_file_p:FILE);
21     leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
22
23     Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
24
25         data_in_giorni <-- funzione Data_in_Giorni(registrazione_t.data);
26         giorni[i] <-- data_in_giorni;
27         registrazione_app[i] <-- registrazione_t;
28         i <-- i + 1;
29         leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
30
31     Fine finchè
32
33     procedura Selection_Sort_data(registrazioni_app, giorni, n_registrazioni);
34
35     per j uguale a 0; j minore di n_registrazioni; incremento j;
36         registrazione_t <-- registrazioni_app[j];
37         procedura visualizzazione_file(registrazione_t);
38     fine
39
40     scrivi("Per mandare in alto quelle importanti <1> altrimenti un carattere qualsiasi --> ");
41     leggi(importanza);
42
43     se importanza è uguale 1 allora
44         procedura Importanza(registrazioni_app, n_registrazioni);
45     fine se
46
47     Fine procedura
```

ORDINAMENTO_IMPORTANZA

```
5  procedura Ordinamento_Importanza(registrazione_file_p:FILE)
6
7      registrazione_t <-- {0,{0,0,0},{0,0},{0,0},"","",0};
8      n_registrazioni <-- 0;
9      i <-- 0;
10
11     posiziona_il_puntatore(registrazioni_file_p:FILE, (registrazione_t.codice-1):intero * funzione sizeof(registrazione_t), SEEK_END);
12     leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
13     n_registrazioni <-- registrazione_t.codice;
14
15     registrazioni_app[n_registrazioni]:Registrazione;
16
17     riposiziona_puntatore_file_all_inizio(registrazioni_file_p:FILE);
18     leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
19
20     Finchè (NOT funzione feof(registrazioni_file_p:FILE)) ripeti //Funzione utilizzata per sapere se il file pointer si trova posizionato a fine file.
21
22         registrazioni_app[i] <-- registrazione_t;
23         i <-- i + 1;
24         leggo_il_file(&registrazione_t:Registrazione, funzione sizeof(Registrazione), 1, registrazioni_file_p:FILE);
25
26     Fine finchè
27
28     procedura Importanza(registrazioni_app, n_registrazioni);
29
30     Fine procedura
```

3. Testing

3.1 Definizione del Piano di Test

Codice Requisito	Codice Test	Nome	Eventuale Input	Risultato Atteso	Risultato Ottenuto
R01	1.1	Menu Iniziale	1	Menu Gestione registrazioni	Menù Gestione registrazioni
R01	1.2	Menu iniziale	Carattere	Messaggio d'errore e visualizzazione main	Messaggio d'errore e visualizzazione e main
R01	1.3	Menu iniziale	7	Messaggio d'errore e visualizzazione main	Messaggio d'errore e visualizzazione e main
R02	2.1	Menu Gestione registrazioni	0	Torna indietro al menù gestione registrazione	Torna indietro al menu gestione registrazioni
R02	2.2	Menu Gestione registrazioni	-1	Messaggio d'errore e visualizzazione menù registrazione	Messaggio d'errore e visualizzazione e menù registrazione
R03	3.1	Aggiungi data registrazione	Stringa di char	Messaggio d'errore e inserimento data	Messaggio d'errore e inserimento data
R03	3.2	Aggiunta data registrazione	2022/11/33	Messaggio d'errore reinserimento data	Messaggio d'errore reinserimento data
R04	4.1	Inserimento tipologia da ricercare	3(indica il promemoria)	Visualizzazione registrazioni di tipologia ricercata (promemoria)	Visualizzazione registrazioni di tipologia ricercata (promemoria)
R04	4.2	Inserimento tipologia da ricercare	5	Messaggio d'errore e reinserimento tipologia	Messaggio d'errore e reinserimento tipologia

R05	5.1	Inserimento codice registrazione da eliminare	Codice non associato	Messaggio d'errore e ritorno al menu principale	Messaggio di errore ritorno al menu principale
R05	5.2	Inserimento codice registrazione da eliminare	Codice associato a reg.	Cancellazione registrazione con cod indicato	Cancellazione registrazione con cod indicato
R06	6.1	Inserimento ora inizio registrazione	Stringa di char	Messaggio d'errore e reinserimento orario	Messaggio di errore e reinserimento orario
R06	6.2	Inserimento ora fine registrazione	30:12	Messaggio d'errore e reinserimento orario	Messaggio di errore e reinserimento orario
R06	6.3	Inserimento ora inizio e ora fine registrazione	Orario nel lasso di tempo di una regi già presente	Messaggio d'errore e visualizzazione registrazioni presenti e reinserimento campi	Messaggio di errore e visualizzazione e registrazioni presenti e reinserimento campi
R07	7.1	Statistica appuntamenti	Inserimento data inizio e fine in cui ci sono appuntamenti	Visualizzazione media giornaliera appuntamenti e numero appuntamenti importanti	Visualizzazione media giornaliera appuntamenti e numero appuntamenti importanti
R08	8.1	Ricerca per oggetto	Ricerca una che non è presente in nessun'altra registrazione	Messaggio d'errore	



3.2 Esiti del Piano di Test ed eventuali commenti

7.1

Inserisci la data di inizio

Inserisci la data nel formato yyyy/mm/dd

--->2022/11/1

Data inserita correttamente
inserimento data fine

Inserisci la data nel formato yyyy/mm/dd

--->2022/11/30

Data inserita correttamente

Media appuntamenti giornalieri ->0.103448

Gli appuntamenti importanti in questo periodo sono ->3

6.3

Inserire l'orario di inizio nel formato h:m

--->9:30

Inserire l'orario di fine nel formato h:m

--->13:50

Verifico la tua richiesta.....

CODICE -->3

IMPORTANTE -->SI

DATA -->2022/12/25

ORARIO INIZIO -->13:0

DURATA -->2.50

TIPOLOGIA -->EVENTO

DESCRIZIONE -->Pranzo di Natale

Registrazione già esistente

Inserisci la data nel formato yyyy/mm/dd

--->

1.1

```

*****

*          <0>  EXIT                      *
*          <1>  GESTIONE REGISTRAZIONE    *
*          <2>  RICERCA REGISTRAZIONI     *
*          <3>  STATISTICA APPUNTAMENTI   *
*          <4>  ORDINAMENTI REGISTRAZIONE *
*****

Inserisci la tua scelta --> 1
|
*****

*          <1>  Aggiungi registrazione    *
*          <2>  Modifica registrazione    *
*          <3>  Elimina registrazione     *
*          <0>  Torna indietro            *
*****

Inserisci la tua scelta -->

```

8.1

```

Ricerca per oggetto
N.B. Vale anche per sottostringa
Inserisci l'argomento da ricercare
--->ciao
|
Non e stata trovata nessuna registrazione

*****

*          <0>  EXIT                      *
*          <1>  GESTIONE REGISTRAZIONE    *
*          <2>  RICERCA REGISTRAZIONI     *
*          <3>  STATISTICA APPUNTAMENTI   *
*          <4>  ORDINAMENTI REGISTRAZIONE *
*****

Inserisci la tua scelta -->

```

5.1

```
Inserisci il codice della registrazione da modificare
-->23
|
Il codice inserito non e associato a nessuna registrazione
Inserisci il codice della registrazione da modificare
-->
```

4.2

```
Inserimento tipologia
<1> EVENTO      <2> APPUNTAMENTO      <3> PROMEMORIA
----> 5
|
Inserisci un valore valido
Inserimento tipologia
<1> EVENTO      <2> APPUNTAMENTO      <3> PROMEMORIA
---->
```