

Esquema das estruturas utilizadas

Estrutura das Tarefas

```
int id;  
char tarefa[TLENG];  
long int prioridade;  
Data prazo;  
int id_user;  
struct task* next;
```

Estrutura dos Utilizadores

```
char nome[NLEN];  
int id;  
char email[TLENG];  
int n_tarefas;  
struct user* next;  
Task* l task;
```

Estrutura da data

```
int dia;  
int mes;  
int ano;
```

Neste projeto foram usadas três estruturas, uma para as tarefas, uma para os utilizadores e outra para a data.

A estrutura das tarefas (Task) contém o identificador da tarefa do tipo “int”, a descrição textual da tarefa do tipo “char”, a prioridade da tarefa, que pode ir de 0 a 10, o seu prazo do tipo “Data” (usando a estrutura de datas), sendo estas três últimas inseridas pelo utilizador na criação de uma nova tarefa, outro tipo “int” que vai ser usado, quando passarmos a tarefa do “to_do” para o “doing”, para guardar o identificador único do utilizador no qual a tarefa vai passar a ficar associada, e temos também um ponteiro para avançar para a próxima tarefa.

A estrutura dos utilizadores (User) contém dois tipo “char” associados ao nome e ao email do utilizador, o identificador do utilizador e o número de tarefas que o utilizador já tem, ambos do tipo “int”, um ponteiro para a estrutura das tarefas, esta vai servir para guardar tarefas associadas a um utilizador, e também um ponteiro para o próximo utilizador.

A estrutura da data tem apenas três variáveis, sendo elas o dia, o mês e o ano, do tipo “int”.

O programa, quando executado, inicia as listas com a função *inicia_lista()*, dá load das tarefas nas respetivas listas (“to do”, “doing” ou “done”) que foram criadas/usadas da última vez que este foi aberto, inicia a lista de utilizadores com a função, dá load dos utilizadores que estão inseridos num ficheiro (users.txt) e load ao máximo de tarefas que está, também, num ficheiro.

O programa começa com um menu de opções que permita ao utilizador escolher entre:

- Lista de pessoas - Apresenta os utilizadores com as suas respectivas tarefas associadas;
- Nova Tarefa em To Do - Esta permite ao utilizador criar uma nova tarefa. Usa a função *new_task(to_do)* no qual é introduzido com parâmetro a lista *to_do*. Esta função põe inicialmente o ponteiro da ultima tarefa para a próxima posição e aloca memória para a mesma. Pede ao utilizador o identificador único da tarefa e compara-o com os identificadores das tarefas existentes, caso haja algum com esse identificador, o programa pede outra vez o id. Após isso, pede a tarefa e guarda-a na lista. Pede a prioridade que pode ir de 0 a 9, consoante a sua importância. Criada a tarefa, o programa vai organizar as tarefas atuais consoante a sua prioridade com a função *priorize_tasks(to_do)*.
- Muda de lista - Permite ao utilizador mudar uma tarefa de lista. Pede ao utilizador a lista no qual pretende trocar. Tem 3 opções sendo estas do "To Do", do "Doing" e do "Done". Se seleccionar "To_Do" ele vai correr a função *switch_list_user(to_do,doing,l_users,max)*, que pedirá o ID do utilizador ao qual a tarefa vai ficar associada e vai verificar se este não excede o máximo de tarefas. Após isso, irá dar as tarefas que estão atualmente em "To_Do", no qual podemos seleccionar, e irá trocar as informações da tarefa para a lista "Doing". Pedirá também um prazo para a tarefa. Após isso chama a função *transfer_to_l_user*, que guardará as informações da tarefa na estrutura do utilizador atual (mais especificamente no *l_task* que é um ponteiro para uma estrutura do tipo tarefa). E guarda as informações no ficheiro "Doing.txt" e "to_do.txt".
Se seleccionar "Doing", poderá seleccionar para qual lista quer mudar a tarefa("To_Do" ou "Done") e qual a tarefa que pretende mover. Ela irá trocar as informações da tarefa da lista "Doing" para a lista seleccionada. Após isso guardará as informações das tarefas da lista "Doing" em "doing.txt" e da lista seleccionada no ficheiro correspondente.
Se seleccionar "Done", poderá seleccionar a lista para qual quer mover a tarefa. Poderá seleccionar a "Doing" ou "To_do". Vai mover as informações da tarefa seleccionada da lista "Done" para a lista seleccionada e guardará as informações nos respetivos ficheiros;
- Apagar Tarefas - Permite apagar uma tarefa. Pedirá a lista que contém a tarefa que pretende apagar. Após seleccionar, vai correr a função *delete_task* em que poderá escolher a tarefa que pretende remover e remove. Após isso guarda as informações no respetivo ficheiro (consoante a lista escolhida);
- Visualizar lista de tarefas - Dá print as tarefas todas, organizadas de acordo com a lista em que estão ("To Do", "Doing" ou "Done") e chama as função *display_tasks* 3 vezes sendo o parâmetro de cada uma, as correspondentes listas;
- Redefinir Prioridade - Permite redefinir a prioridade de todas as tarefas na lista "To Do" usando a função *define_priority* no qual corre a lista "To Do" e para cada tarefa pede a nova prioridade. Após isso organiza-as com a função *priorize_tasks* e guarda as informações no ficheiro "to_do.txt"
- Apagar tudo - Elimina todas as tarefas com a função *reset_everything* que é usada três vezes, sendo o parâmetro de entrada diferente em cada uma delas(*to_do,doing* e *done*);