

## Laboratório de Listas Duplamente Encadeadas

- 1) Implemente o TAD DLinkedList utilizando um nó descritor como o mostrado abaixo

### Padronização de erros

Sugestão para padronizar códigos de erro

.h

```
#include <stdlib.h>
#define SUCCESS 0
#define INVALID_NULL_POINTER -1
#define OUT_OF_MEMORY -2
#define OUT_OF_RANGE -3
#define ELEM_NOT_FOUND -4

struct aluno{
    int matricula;
    char nome[30];
    float n1,n2,n3;
};

typedef struct DLinkedList List;
```

.C

```
struct DLinkedList {
    DLNode *begin;
    DLNode *end;
    int size;
};

struct dlnode {
    struct aluno data;
    DLNode *next;
    DLNode *prev;
};
```

"TDLinkedList.h"

```
List* list_create();
int list_free(List *li);

int list_push_front(List *li, struct aluno al);
int list_push_back(List *li, struct aluno al);
int list_insert(List *li, int pos, struct aluno al);
int list_size(List *li);

int list_pop_front(List *li);
int list_pop_back(List *li);
```

```
int list_erase(List *li, int pos);

int list_find_pos(List *li, int pos, struct aluno *al);
int list_find_mat(List *li, int nmat, struct aluno *al);
int list_front(List *li, struct aluno *al);
int list_back(List *li, struct aluno *al);
int list_get_pos(List *li, int nmat, int *pos);

int list_print_forward(List *li);
int list_print_reverse(List *li);
```