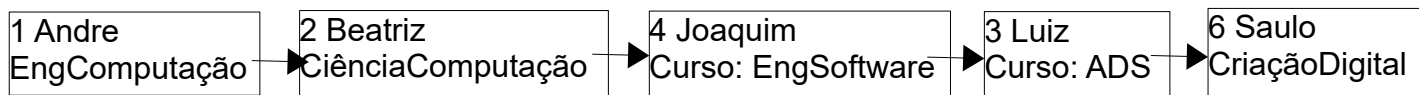


Ligas acadêmicas

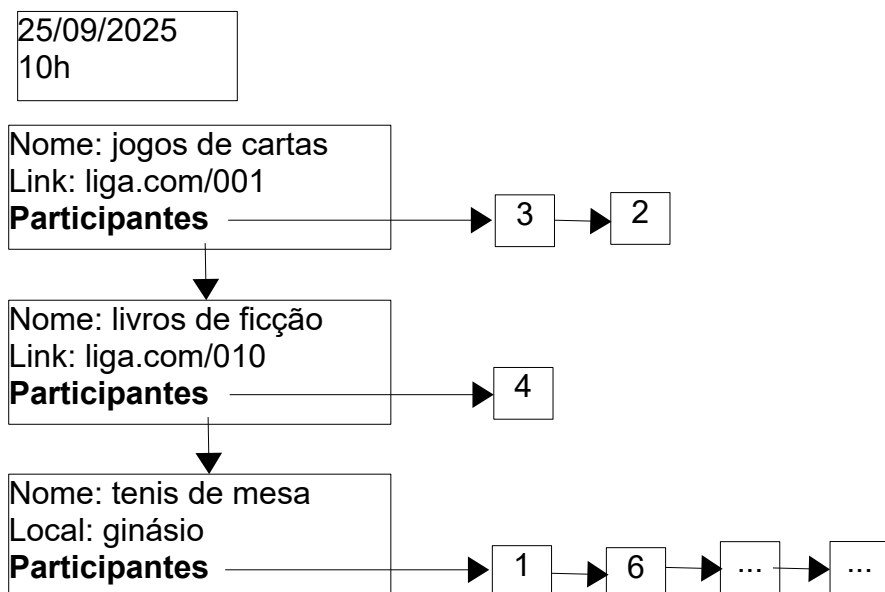
Uma liga acadêmica é uma associação de estudantes sem fins lucrativos focada no aprofundamento de conhecimento e pesquisa em uma área específica. Há ligas de e-games, futebol, livros de ficção, história da cidade, jantares alternativos, jogos de cartas, etc. Para participar das ligas, basta fazer um **cadastro** (lista de estudantes) e participar das atividades, que podem ser presenciais ou *online*. Não é preciso estar inscrito em uma liga específica, apenas estar cadastrado e **inscrever-se na atividade no dia e horário de sua realização**. As atividades são realizadas no mesmo dia e horário (sábado, das 10h às 12h) de forma paralela, ou seja, ao participar de atividade de uma liga não pode participar de outra no mesmo dia.

O trabalho será implementar o cadastro dos estudantes participantes (lista de estudantes, mantida em ordem alfabética) e a agenda dos encontros das ligas (para um dia). De cada estudante, mantém-se no cadastro: **código**, seu **nome** e seu **curso**. Para a **agenda dos ligas** mantém-se **data e horário** (usar *header*), e a **lista das atividades programadas**. Cada atividade programada deve ter **o nome da liga**, a **lista de participantes inscritos**, e o **local** (**link** para acesso para atividades online e endereço para atividades presenciais).

Cadastro



Ligas



Organização:

O trabalho será realizado **em duplas**, e deverá ser entregue/apresentado até **03/10/2025** pela dupla. Ambos os alunos devem estar presentes na apresentação, cada um apresenta uma

parte).

Até o dia da apresentação, postar na tarefa do AVA: código-fonte da implementação.

Implementação:

- a implementação pode ser feita em C, Java, C++, C#, Python;
- a interface pode ser bem simples (textual), o importante é que as operações funcionem;
- **as listas (cadastro, atividades, participantes de cada atividade) devem ser implementadas com alocação dinâmica de memória, através de listas encadeadas (simples ou duplas).** Defina as estruturas de dados e de implementação mais adequadas à execução das operações.
- **a implementação das listas encadeadas** não pode conter classes prontas da linguagem de programação (Java, C++, C#, Python) que mantenham os encadeamentos de forma automática, nem para as listas nem para os nodos (a implementação e manipulação das listas encadeadas fazem parte do trabalho). As referências entre nodos devem ser implementadas no trabalho e mantidas nas diferentes operações.
- as operações solicitadas podem fazer parte de um menu de opções.

Operações básicas a serem implementadas:

Pode haver variações nas entradas e saídas de dados das operações, novas operações podem ser definidas, os nomes e/ou os parâmetros das operações podem ser modificados.

As principais operações a serem implementadas são:

TAD Cadastro{

Dados: estudante

Operações:

insere(E: dados do estudante)

remove(E: código ou nome)

consultatodos(S: lista de estudantes em ordem alfabética):

outras operações...

}

TAD AgendaLigas{

Dados: atividade

Operações:

insere_atividade(E: dados da atividade) : insere a atividade na agenda

remove_atividade (E: nomeatividade): remove a atividade da agenda

insere_participantes (E: nomeatividade, participante): insere um participante em uma atividade. Só pode ser inserido se estiver cadastrado, e se não estiver inscrito em outra atividade

consulta_atividade (E: nomeatividade; S: dados da atividade): consulta nome da atividade, e exibe todas as informações daquela atividade

consulta_quantidades (S: total de participantes por atividade)

consulta_todas_atividades (S: lista de atividades): lista todas as atividades da agenda

consulta_participantes (E: nomeatividade; S: dados da atividade, nomes dos participantes): exibe nomes dos participantes daquela atividade

outras operações...

}

Avaliação:

A implementação das estruturas e operações apresentadas definidas conforme especificado e funcionando, dará ao trabalho nota máxima 9,0. Essa nota poderá chegar a 10,0 com o acréscimo de outras funcionalidades que permitam incrementar as estruturas de dados e suas operações: outras operações de consulta, definição e uso adequado de *header* (cabeçalho) para as listas, etc.