

## Introdução

O CUnit é uma ferramenta para teste dinâmico. Com ele é possível fazer teste com várias entradas diferentes e verificar se a saída corresponde à saída. O CUnit é composto por um registro de teste, que é composto por vários suites de teste (normalmente 1 por módulo), que é composto por vários casos de testes (vários por função). Cada um destes testes é executado toda vez que o registros de teste é chamado.

No exemplo que está na pasta exemplo tem varias funções essenciais para o funcionamento do CUnit, todas comentadas. É importante também reparar no makefile, que para compilar é necessário adicionar o parâmetro `-lcunit` para que a biblioteca `libcunit.a` seja adicionada. Além disso é necessário a pasta CUnit para que algumas das definições usadas no código.

Para a entrega do laboratório, siga as orientações de implementações dos tipos abstratos de dados (TADs) conforme a seção a seguir, e conforme a abordagem desenvolvimento dirigido a modelos (TDD). Será avaliada a **qualidade** dos testes e se eles abrangem os requisitos básicos da estrutura de dados escolhida. Tenha em mente que ter 3 ou 4 testes por função (como é o caso do exemplo) é muito pouco. Ao fazer os testes, colocar comentários explicando o testes feitos. Fazer testes com nomes mnemônicos.

Este laboratório pode ser feito em dupla. Ao enviar, deve ser enviado uma pasta contendo o os módulos de definição (.h) e de implementação (.c) com seus respectivos módulos controladores de teste. Configure um makefile para compilar e executar o programa e tudo que for necessário para que o programa possa ser compilado( incluindo a biblioteca `libcunit.a` os .h da pasta do CUnit ).

## Motivação

Tipos abstratos de dados são elementos que implementam um único conceito, conforme estudado em Programação Sistemática. Podemos nos deparar com algumas nas quais observamos um fluxo de informação que se dá de maneira circular: a trajetória de um ponto em MCU (movimento circular uniforme), um rodízio de pizzas, a contagem das horas dos dias, a rotina de uma pessoa ou Hobbit, etc.

Pensando nisso, o objetivo dessa atividade é a implementação de um módulo que trate esse tipo de informação. E, para maior confiabilidade, a elaboração de casos de teste para embasar a corretude dos algoritmos.

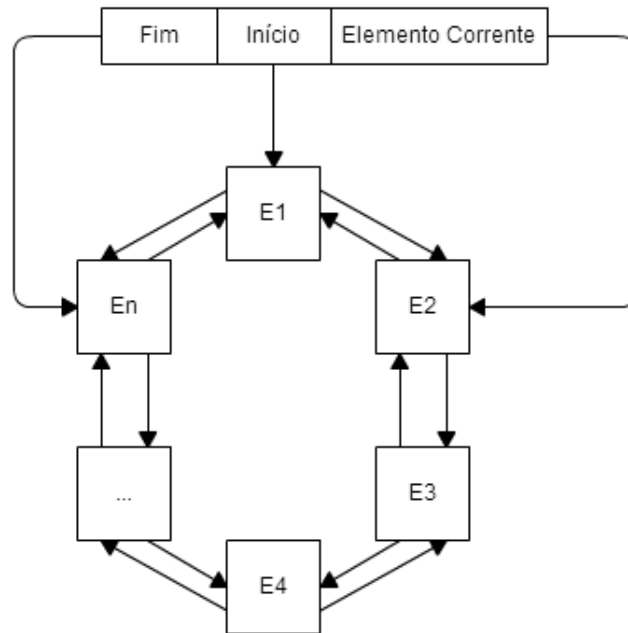
## Implementação dos TADs:

São sugeridos 2 tipos de implementação: uma lista circular duplamente encadeada ou uma pilha, conforme descrição a seguir.

### Lista circular duplamente encadeada:

Cada elemento da lista possui referência para o seu sucessor e antecessor. Para o controle, acesso e modificações da estrutura, um header armazena referências para início e fim da lista e para o elemento corrente. Assim que o elemento foi utilizado, o ponteiro corrente é atualizado para o próximo elemento.

Lista Circular



### Duas filas:

O mesmo processo pode ser feito com 2 filas, onde elas se comunicam da seguinte forma:

1. A primeira é preenchida com os elementos enquanto a segunda fila está vazia
2. Cada elemento da primeira fila é transferido para a segunda fila, ou seja, à medida que são utilizados, são passados da primeira para segunda fila.
3. Quando a primeira fila não possuir mais elementos, a segunda deverá estar cheia, e o processo deverá se repetir.

### Atividades:

Escolha um TADs e crie um módulo dividido em definição e implementação. Crie funções básicas para a manipulação da estrutura – inserção, remoção, busca, etc, conforme o conceito de cada estrutura de dados.

### Elaboração da massa de teste:

Deve-se criar um módulo controlador de teste (seguindo abordagem TDD disciplinado) usando o CUnit para testar se as principais funcionalidades e restrições do módulo conforme a especificação. O teste disciplinado deve seguir os seguintes passos:

1. Antes de testar: produzir um roteiro de teste;
2. Antes de iniciar o teste: estabelecer o cenário do teste;
3. Criar um módulo controlador de teste, usando a ferramenta CUnit para testar as principais funcionalidades de cada módulo;
4. Ao testar: produzir um laudo em que todas as discrepâncias encontradas são registradas. Esse laudo pode ser uma saída da execução do CUnit. Somente termine o teste antes de completar o roteiro, caso observe que não vale mais a pena continuar executando o roteiro, uma vez que o contexto para o resto está danificado;
5. Após a correção: repetir o teste a partir de 2 até o roteiro passar sem encontrar falhas.

**Obs.:** As atividades deverão ser feitas aos pares: enquanto um implementa o outro verifica a corretude do código visualmente. De modo que, enquanto um dos integrantes está codificando, o segundo observa a corretude do código e ambos discutem o que deve ser feito. Essas duas funções devem ser revezadas no prazo de 30 a 40 minutos (ou menos). Esse método é bastante utilizado comercialmente, e comprovadamente mais eficaz que dois programadores trabalhando separadamente.

## Problema:

Bilbo Bolseiro é um Hobbit que vive pacificamente no condado. Inesperadamente, recebe a visita do mago Gandalf e 13 anões. Todos estão famintos e querem que o pequeno Hobbit providencie toda a comida. Então, Bilbo anotou o nome e mais algumas informações de todos os presentes, para que pudesse conhecer melhor toda aquela gente e que a comida pudesse ser distribuída para cada um em uma grande mesa circular:

/\*\* Alguns podem achar uma pergunta boba essa sobre o gênero de seus convidados. Porém, Bilbo sabe que as esposas deles também têm barba. \*\*/

Nome	Idade	Gênero	Tipo	Humor atual
Thorin	222	Masculino	Anão	Ruim
Fili	82	Masculino	Anão	Moderado
Kili	77	Masculino	Anão	Moderado
Oin	167	Masculino	Anão	Péssimo
Gloin	158	Masculino	Anão	Péssimo
Balin	216	Masculino	Anão	Bom
Dwalin	169	Masculino	Anão	Péssimo
Ori	222	Masculino	Anão	Ruim
Dori	222	Masculino	Anão	Péssimo
Nori	222	Masculino	Anão	Ruim
Bifur	127	Masculino	Anão	Péssimo
Bofur	222	Masculino	Anão	Moderado
Bombur	222	Masculino	Anão	Péssimo
Gandalf	222	Masculino	Mago	Feliz

As várias perguntas de Bilbo não agradaram muito. Todos sabem que o humor dos anões melhora à medida que comem (péssimo, ruim, moderado, bom, satisfeito, feliz). Então, pensando nas estruturas discutidas anteriormente, crie um algoritmo usando o módulo criado para fazer o rodízio de comida para cada anão e o mago Gandalf. Lembre-se que eles gostam de ser tratados pelo nome, e, também, seria muito ruim se fossem confundidos com uma mulher. À medida que forem ficando felizes, eles saem da mesa e o Hobbit poderá descansar.