



INSTITUTO FEDERAL

Norte de Minas Gerais

Campus Januária

Estruturas de Dados 2

- Generic Lists -



INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Situação Problema





INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Situação Problema

Imagine o desenvolvimento de uma engine para jogo MMO-RPG.





INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Situação Problema

Imagine o desenvolvimento de uma engine para jogo MMO-RPG.

Teremos que pensar em estruturas de dados para representar: Jogadores Online, Inimigos, Itens coletados, etc...





INSTITUTO FEDERAL
Norte de Minas Gerais
Campus Januária

Situação Problema

Imagine o desenvolvimento de uma engine para jogo MMO-RPG.

Teremos que pensar em estruturas de dados para representar: Jogadores Online, Inimigos, Itens coletados, etc...

Estruturas Dinâmicas seriam essenciais para o desempenho.





Problema Atual

- **Problema Atual**: Existe um **forte acoplamento** entre a ***Entidade*** (objetos que queremos representar) e a ***Estrutura de Dados*** adotada para armazená-la (*p.ex.* Lista Encadeada).



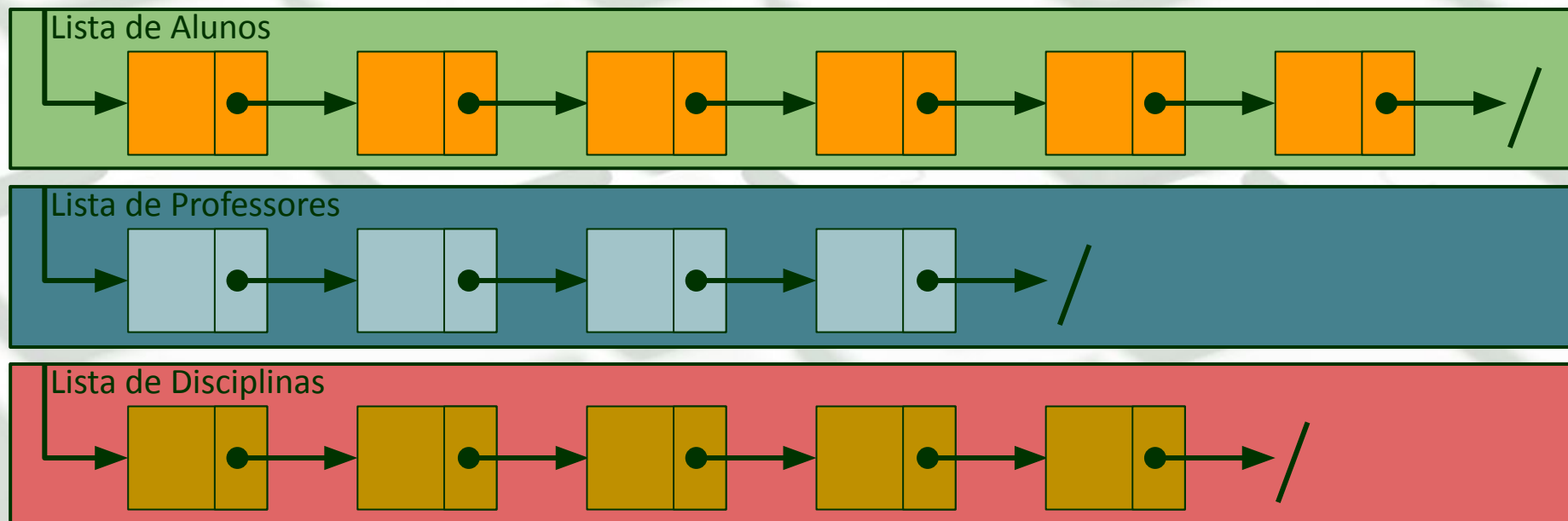
Problema Atual

- **Problema Atual**: Existe um **forte acoplamento** entre a **Entidade** (objetos que queremos representar) e a **Estrutura de Dados** adotada para armazená-la (*p.ex.* Lista Encadeada).
- Para tornar mais claro, implemente a seguinte solução...
Faça um programa que armazene, em listas dinâmicas, as seguintes estruturas: Aluno, Professor, Disciplina.
 - **Aluno** => Núm. Matrícula, Nome, Curso, etc...
 - **Professor** => CPF, Nome, Titulação, etc...
 - **Disciplina** => Código, Nome, Curso, Período, etc...



Problema Atual

- Perceba na solução que, **ao invés de implementar uma única Estrutura de Dados do tipo Lista Encadeada**, implementamos 03 tipos distintos de Lista Encadeada...





Problema Atual

- Perceba que as operações (algoritmos) para manutenção das listas são as mesmas, independentemente da **entidade**, mas sempre implementamos cada operação para cada entidade...

cadastrarAluno()
<Insere na Lista>

consultarAluno()
<Consulta Lista>

editarAluno()
<Edita Lista>

excluirAluno()
<Exclui na Lista>

cadastrarProf()
<Insere na Lista>

consultarProf()
<Consulta Lista>

editarProf()
<Edita Lista>

excluirProf()
<Exclui na Lista>

cadastrarDisc()
<Insere na Lista>

consultarDisc()
<Consulta Lista>

editarDisc()
<Edita Lista>

excluirDisc()
<Exclui na Lista>



Problema Atual

- Perceba que as operações (algoritmos) para manutenção das listas são as mesmas, independentemente da **entidade**, mas sempre implementamos cada operação para cada entidade... Isso é o **ALTO ACOPLAMENTO**.

cadastrarAluno()
<Insere na Lista>

consultarAluno()
<Consulta Lista>

editarAluno()
<Edita Lista>

excluirAluno()
<Exclui da Lista>

cadastrarProf()
<Insere na Lista>

consultarProf()
<Consulta Lista>

editarProf()
<Edita Lista>

excluirProf()
<Exclui da Lista>

cadastrarDisc()
<Insere na Lista>

consultarDisc()
<Consulta Lista>

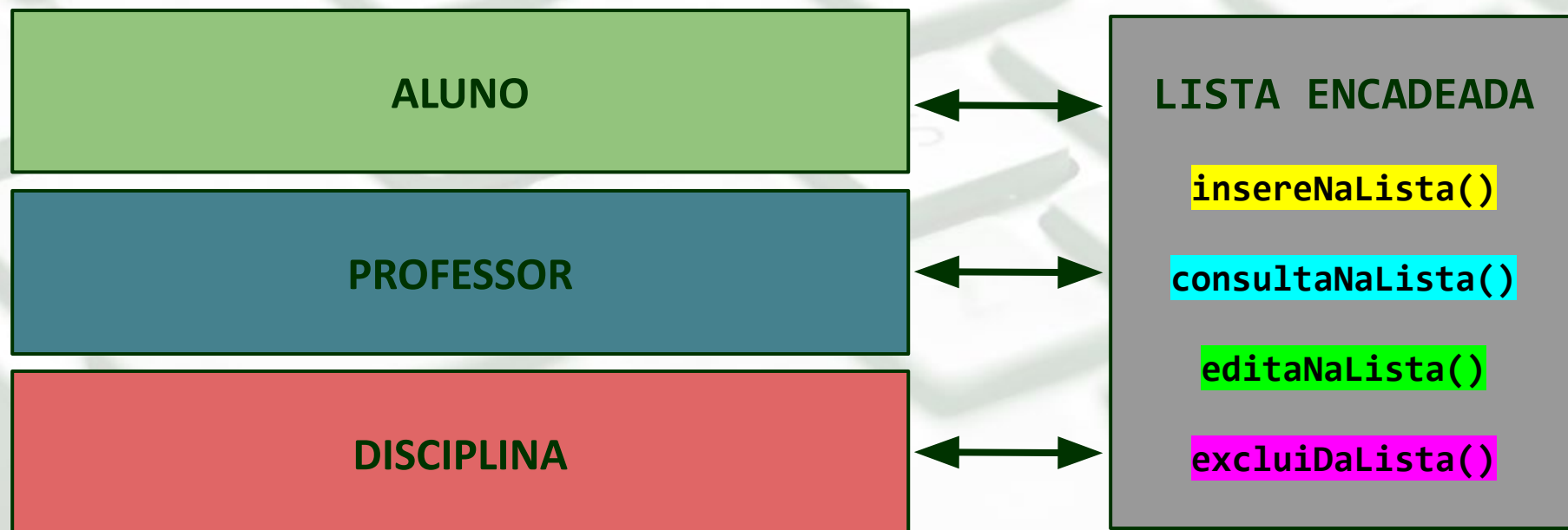
editarDisc()
<Edita Lista>

excluirDisc()
<Exclui da Lista>



Problema Atual

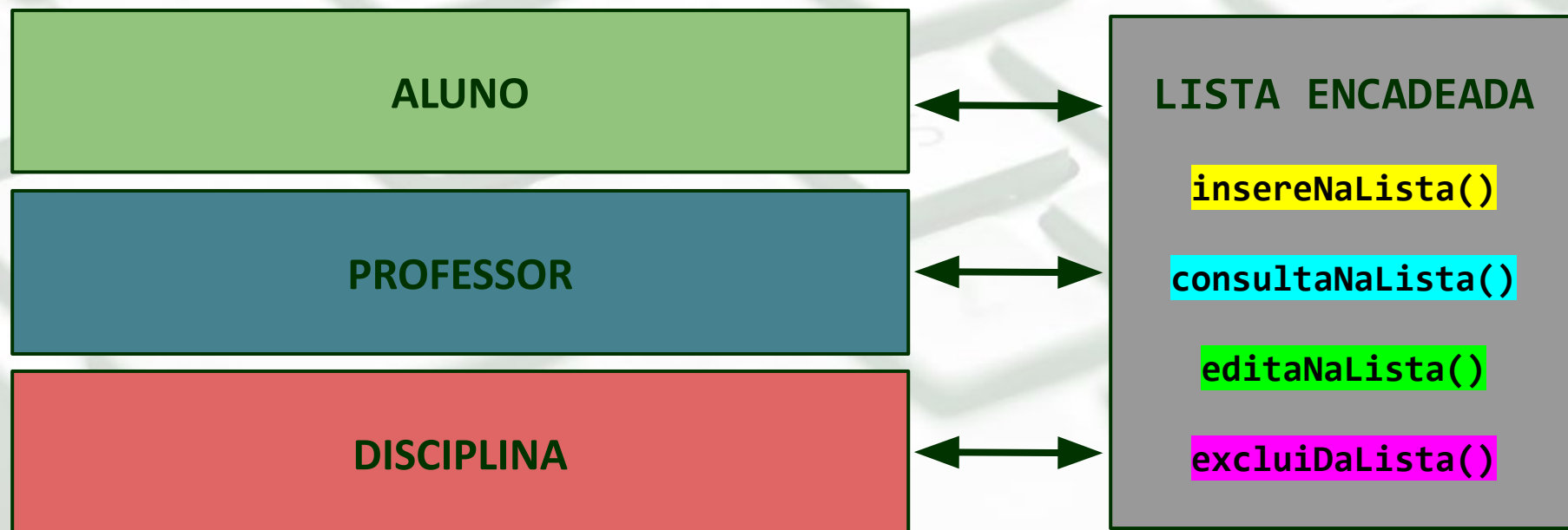
- “No mundo ideal” o mais interessante é que as operações sobre a Estrutura de Dados **Lista Encadeada** (controle) sejam **INDEPENDENTES** às entidades que a utiliza (modelo).





Listas Genéricas

- Nesta proposta, temos que implementar um tipo de **Lista Genérica**, onde todas as **operações podem ser compartilhadas por qualquer entidade**.





1º Desafio

- Como evitar o alto acoplamento sendo que, na própria entidade definimos o tipo de lista a ser utilizada...

```
typedef struct Aluno{  
    int numMatricula;  
    char nome[100];  
    char curso[100];  
    struct Aluno *prox;  
}Aluno;
```



1º Desafio

- Como evitar o alto acoplamento sendo que, na própria entidade definimos o tipo de lista a ser utilizada...

```
typedef struct Aluno{  
    int numMatricula;  
    char nome[100];  
    char curso[100];  
    struct Aluno *prox;  
}Aluno;
```

JÁ COMEÇOU ERRADO!

Se a ideia é separar a entidade (Aluno) da estrutura de dados (Lista Encadeada) não podemos deixar esses dois conceitos tão acoplados!



Listas Genéricas

■ Como definir então uma lista genérica???

```
typedef struct Node{  
    void* item;  
    struct Node* next;  
}Node;
```



Listas Genéricas

■ Como definir então uma lista genérica???

```
typedef struct Node{  
    void* item;  
    struct Node* next;  
}Node;
```





Listas Genéricas

■ Como definir então uma lista genérica???

```
typedef struct Node{  
    void* item;  
    struct Node* next;  
}Node;
```



PS... Já pensou como é possível existir linguagens de tipagem fraca???



Listas Genéricas

■ Como definir então uma lista genérica???

Clique e leia depois...

How Python objects are implemented in C?



Xinzhe Li, PhD in Language Intelligence · Follow
2 min read · Jul 10, 2022

}Node;

PS... Já pensou como é possível existir linguagens de tipagem fraca???





Listas Genéricas

■ Como definir então uma lista genérica???

```
typedef struct Node{  
    void* item;  
    struct Node* next;  
}Node;
```



Porém um Node não representa por si só toda uma lista encadeada...



Listas Genéricas

■ Como definir então uma lista genérica???

```
typedef struct Node{  
    void* item;  
    Node* next;  
}Node;
```

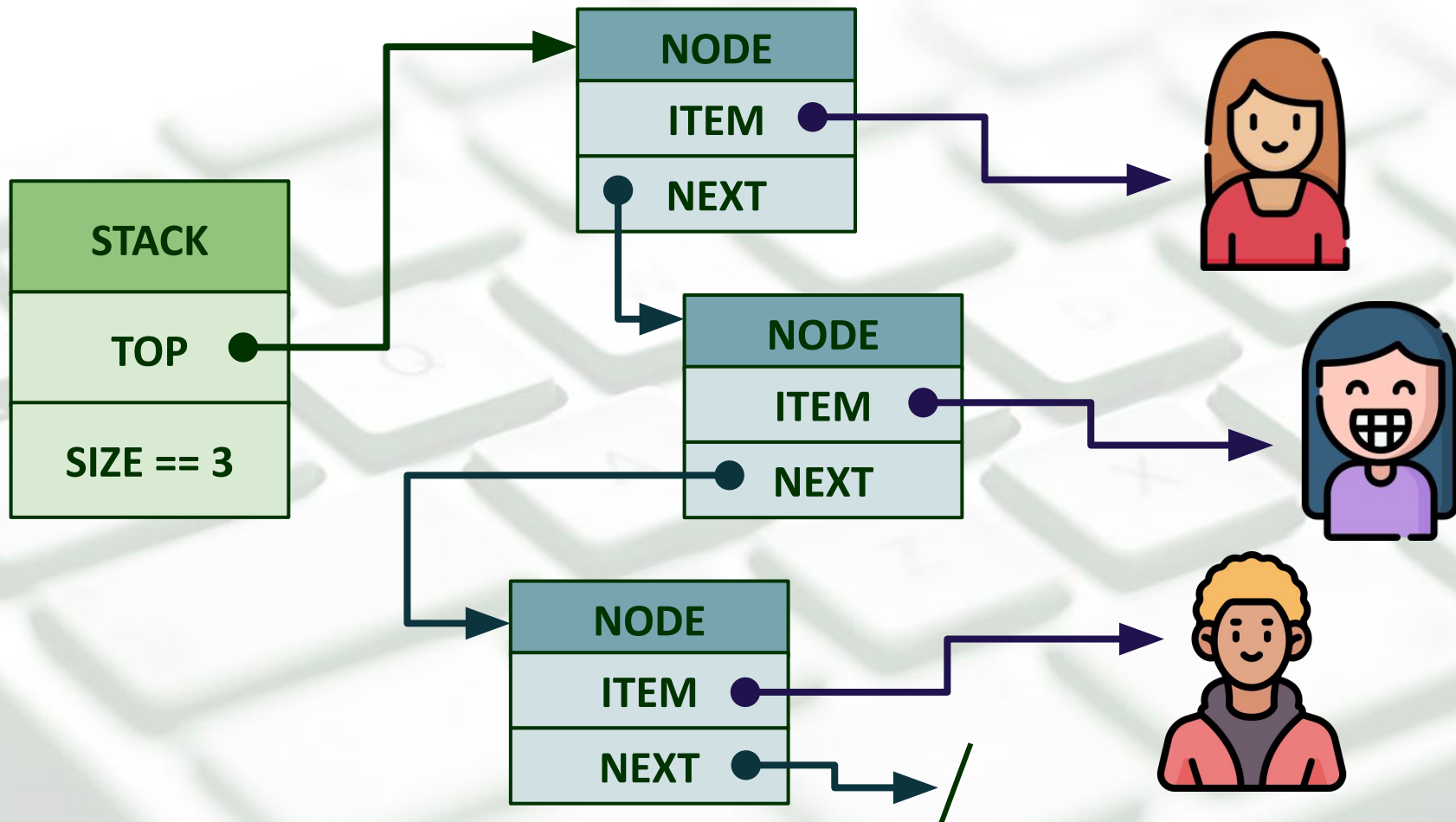
Lista Genérica em Formato Pilha (Stack)

```
typedef struct{  
    Node* top; //início da pilha  
    int size; //tamanho da pilha  
    //outros dados da pilha que desejar  
    //armazenar para acesso direto.  
}Stack;
```



Listas Genéricas

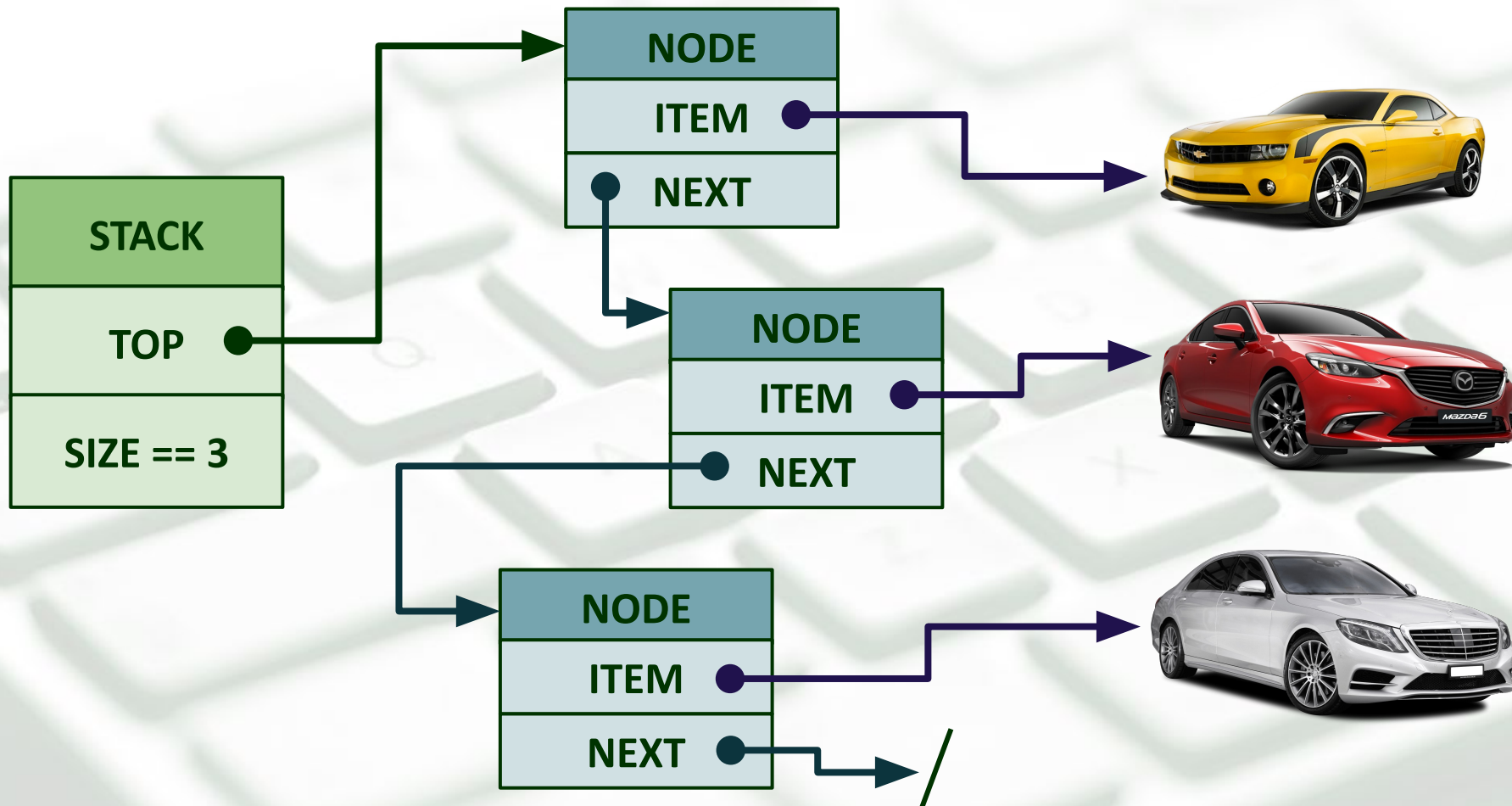
■ Projeto de Implementação de Lista Genérica...





Listas Genéricas

■ Projeto de Implementação de Lista Genérica...





Listas Genéricas

- Vamos criar uma **biblioteca com funções genéricas** para criação/tratamento de listas em formato Stack (Pilha).

`stack_create()`
◁Criar uma Pilha Vazia▷

`stack_push()`
◁Adicionar Nó na Pilha▷

`stack_pop()`
◁Excluir primeiro Nó▷

`stack_list()`
◁Listar todos Nós da Pilha▷

`stack_search()`
◁Buscar um Nó na Pilha▷

`stack_edit()`
◁Editar um Nó na Pilha▷

`stack_remove()`
◁Remover um Nó da Pilha▷

`stack_clear()`
◁Limpar toda a Pilha▷

Todas essas funções podem ser utilizadas por toda e qualquer entidade (struct)



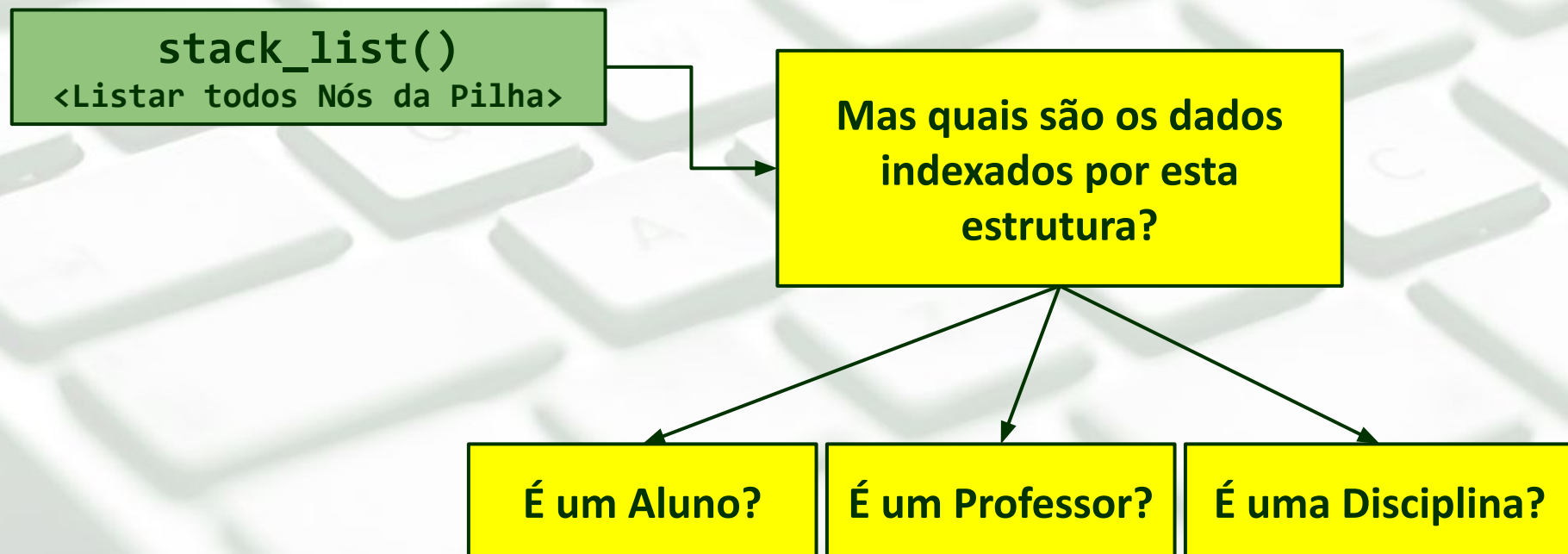
2º Desafio

- Já que os itens de uma Lista Genérica são *“indefinidos”*, **como aplicar funções específicas para estes itens?**



2º Desafio

- Já que os itens de uma Lista Genérica são *“indefinidos”*, como aplicar funções específicas para estes itens?

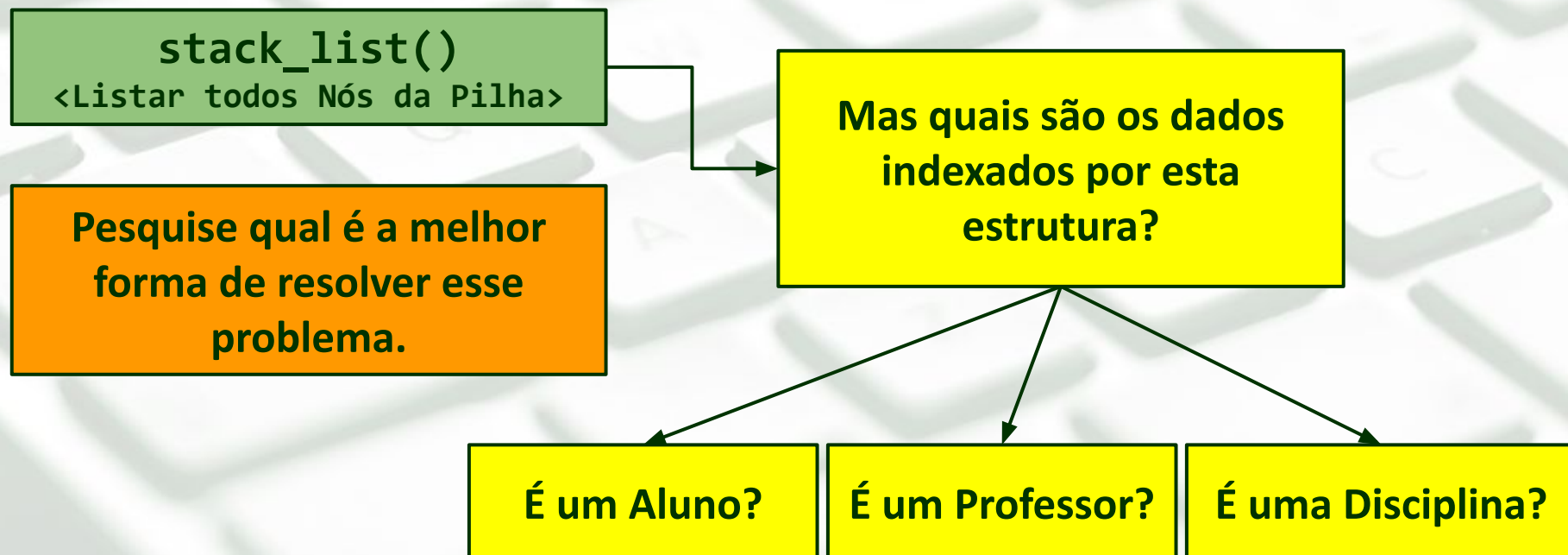




2º Desafio



- Já que os itens de uma Lista Genérica são *“indefinidos”*, como aplicar funções específicas para estes itens?





Apresentação Teórica

- **Apresentação Teórica**
- Pesquise como um processo é organizado em Memória.
- O que é o espaço de endereçamento HEAP?
- O que é o espaço de endereçamento STACK?

