



Atividade 1 – Estruturas de Dados

Docente

Prof. Otacílio José Pereira

Discente

Felipe Sacramento de Lima

Introdução ao Conceito de Indústria 4.0

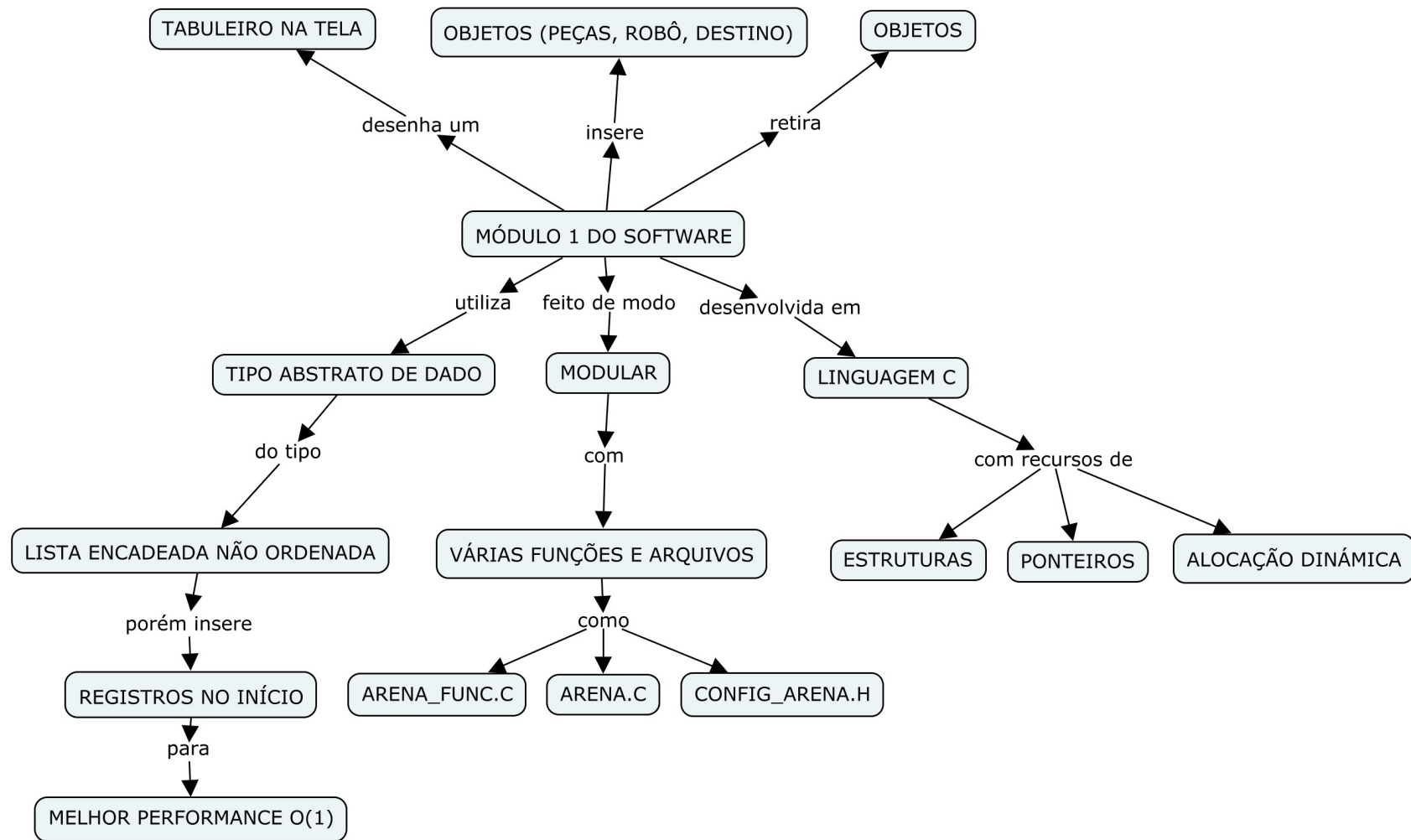
4º Revolução Industrial

Objetivos:

- “Promover a automatização da manufatura...
- Decisões são tomadas de forma automática a partir do uso de um grande conjunto de dados armazenados, chamado de Big Data.
- Requer infraestrutura tecnológica formada por sistemas físicos e virtuais, com apoio de Big Data, robôs automatizados, simulações, manufatura avançada, realidade aumentada e da IoT.”



Mapa Conceitual do Software Desenvolvido



Detalhes sobre a Implementação

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #include "config_arena.h" // Contém definição de estruturas, protótipo das funções
5                             // e declaração de constantes.
6
7 /* Protótipo de um simulador no qual os objetos (peças, robô, destinos)
8  * tem uma representação dentro da arena.
9  */
10
11 int main(void)
12 {
13     int opcao; // Armazena a opção do usuário do Menu Principal.
14     char resposta_acao; // Armazena a opção de deseja continuar laço.
15     TOBJETO objeto; // Armazena o registro do objeto.
16     TLISTA lista; // Início da lista encadeada com registro dos objetos.
17
18     inicializar_lista(&lista); // Inicializa a lista encadeada.
19     inicializar_arena(); // Inicializa a arena preenchendo com pontos.
20
21     do {
22         opcao = menu_principal(); // Roda a função e retorna a opção do usuário.
23
24         switch (opcao) {
25             case 1: objeto = ler_objeto(PECA); // Ler dados da peça
26                 posicionar_obj(&lista, objeto); // Posicionar peça
27                 break;
28             case 2: visualizar_lista_obj(lista); // Visualizar lista de objetos
29                 retirar_obj(&lista); // Retirar peça
30                 break;
31             case 3: objeto = ler_objeto(ROBO); // Ler dados do robô
32                 posicionar_obj(&lista, objeto); // Posicionar robô
33                 break;
34             case 4: objeto = ler_objeto(DESTINO); // Ler dados de um destino
35                 posicionar_obj(&lista, objeto); // Posicionar destino
36                 break;
37             case 5: visualizar_arena(); // Visualizar arena
38                 break;
39             case 9: exit(0); // Finalizar aplicativo
40
41             default: printf("\n Opção inválida. Por favor tente novamente.");
42                     break;
43         }
44         printf("\n Você deseja voltar ao Menu Principal? (S)im ou (N)ao: ");
45         scanf(" %c", &resposta_acao);
46     }
```

```
1 #ifndef MODULO1_CABECALHO_H
2 #define MODULO1_CABECALHO_H
3
4 #define LINHAS 6 // Protótipo da arena é uma matrix de 10x10.
5 #define COLUNAS 6
6
7 #define RED "\x1B[31m" // Solução de colorir para terminais Linux.
8 #define GRN "\x1B[32m"
9 #define BLU "\x1B[34m"
10 #define RESET "\x1B[0m"
11 #define YEL "\x1B[33m"
12
13 enum TIPOS_OBJETOS { PECA = 0, ROBO, DESTINO };
14 enum DESTINOS { VERMELHO = '1', VERDE, AZUL };
15
16 typedef struct { // Definição de um novo tipo de dado chamado TOBJETO.
17     int x, y; // Coordenadas X e Y para posicionar peças.
18     char cor; // Cor da peça: R para vermelho, G para verde e B para azul.
19     int tipo; // Tipo: 0 para peça, 1 para robô, 2 para destino.
20 } TOBJETO;
21
22 typedef struct TNO { // Implementação de uma lista encadeada para armazenar objetos
23     TOBJETO objeto;
24     struct TNO *prox;
25 } *TLISTA;
26
27 char arena[LINHAS][COLUNAS]; // Variável global que representa a matrix da arena.
28
29 void inicializar_lista(TLISTA *pLista); // Inicializa a lista encadeada
30 void inicializar_arena(void); // Inicializa matrix com pontos
31 int menu_principal(void); // Mostra Menu Principal ao usuário
32 void posicionar_obj(TLISTA *pLista, TOBJETO pObj); // Lista encadeada insere objeto
33 TOBJETO ler_objeto(int tipo); // Ler dados do objeto.
34 void retirar_obj(TLISTA *pLista); // Retira algum objeto.
35 void posicionar_destinos(void); // Posiciona destinos.
36 void visualizar_arena(void); // Visualiza arena formatada
37 void visualizar_lista_obj(TLISTA pLista); // Visualiza lista de todos os objetos
38 void colorir_obj_arena(int i, int j); // Mostra objetos com cores.
39 char *nomear_objeto(int tipo); // Retorna nomes dos objetos
40 #endif
```

https://github.com/fslima0/UNIFACS/tree/master/projeto_smart_cities_industria_4_0

Módulo do Aplicativo em Operação

```
-- Menu Principal --  
(1) Posicionar peças  
(2) Retirar peças  
(3) Posicionar robô inicial  
(4) Posicionar destinos  
(5) Visualizar arena  
(9) Sair
```

Opção : 5

	0	1	2	3	4	5
0	R	.	G	.	B	.
1
2	.	.	0	.	.	.
3
4	D	.	D	.	D	.
5

Você deseja voltar ao Menu Principal? (S)im ou (N)ao:

Visualização da arena

```
-- Menu Principal --  
(1) Posicionar peças  
(2) Retirar peças  
(3) Posicionar robô inicial  
(4) Posicionar destinos  
(5) Visualizar arena  
(9) Sair
```

Opção : 2

1. (4, 4, DESTINO, AZUL)
2. (4, 2, DESTINO, VERDE)
3. (4, 0, DESTINO, VERMELHO)
4. (2, 2, ROBÔ, AMARELO)
5. (0, 4, PEÇA, AZUL)

Qual objeto deseja retirar:

Retira objetos

```
-- Menu Principal --  
(1) Posicionar peças  
(2) Retirar peças  
(3) Posicionar robô inicial  
(4) Posicionar destinos  
(5) Visualizar arena  
(9) Sair
```

Opção : 1

1. Coordenadas da peça

 Digite X: 3

 Digite Y: 3

2. Cor da peça

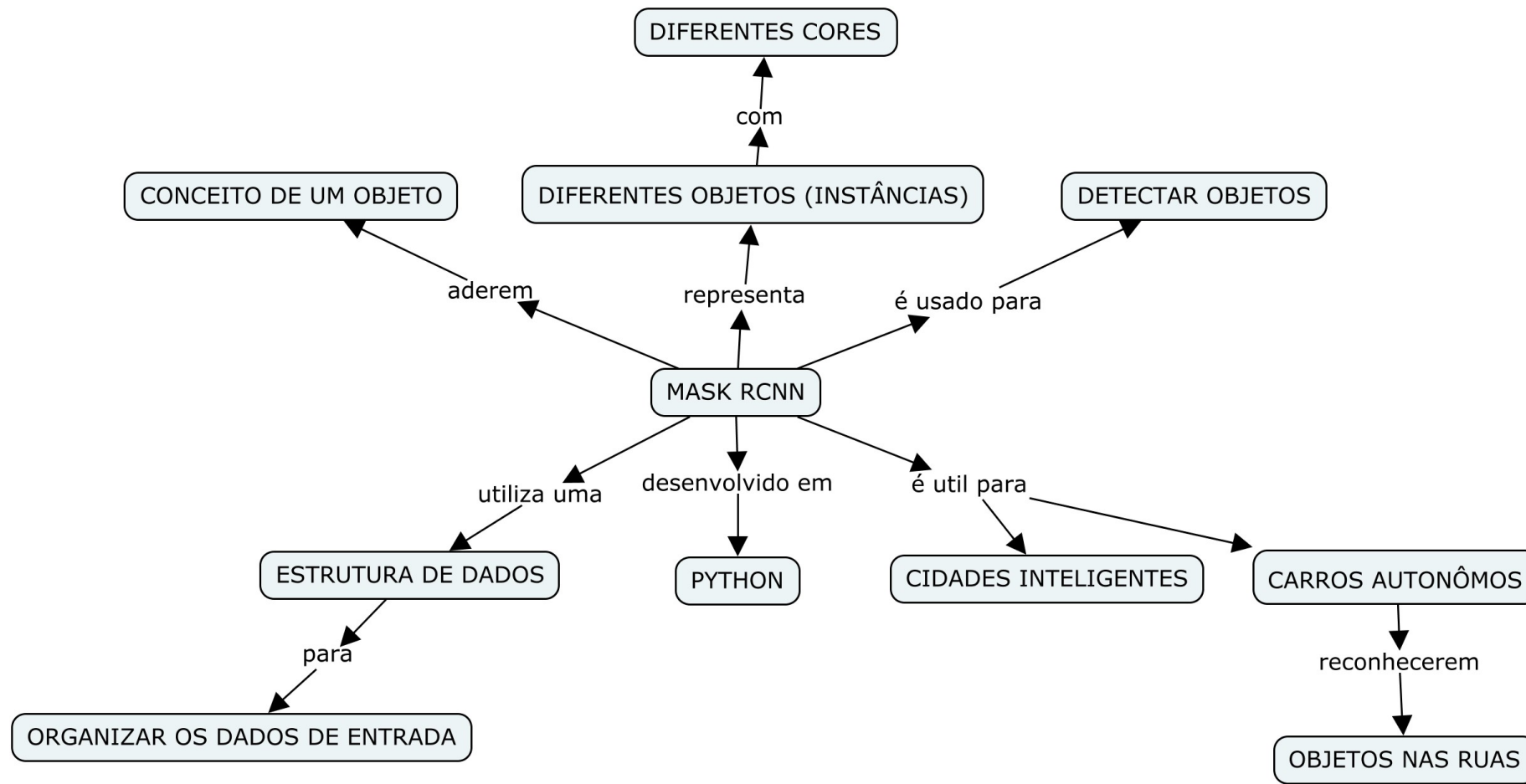
<R> para vermelho, <G> para verde, para azul

 Digite a cor: R

Insere peças

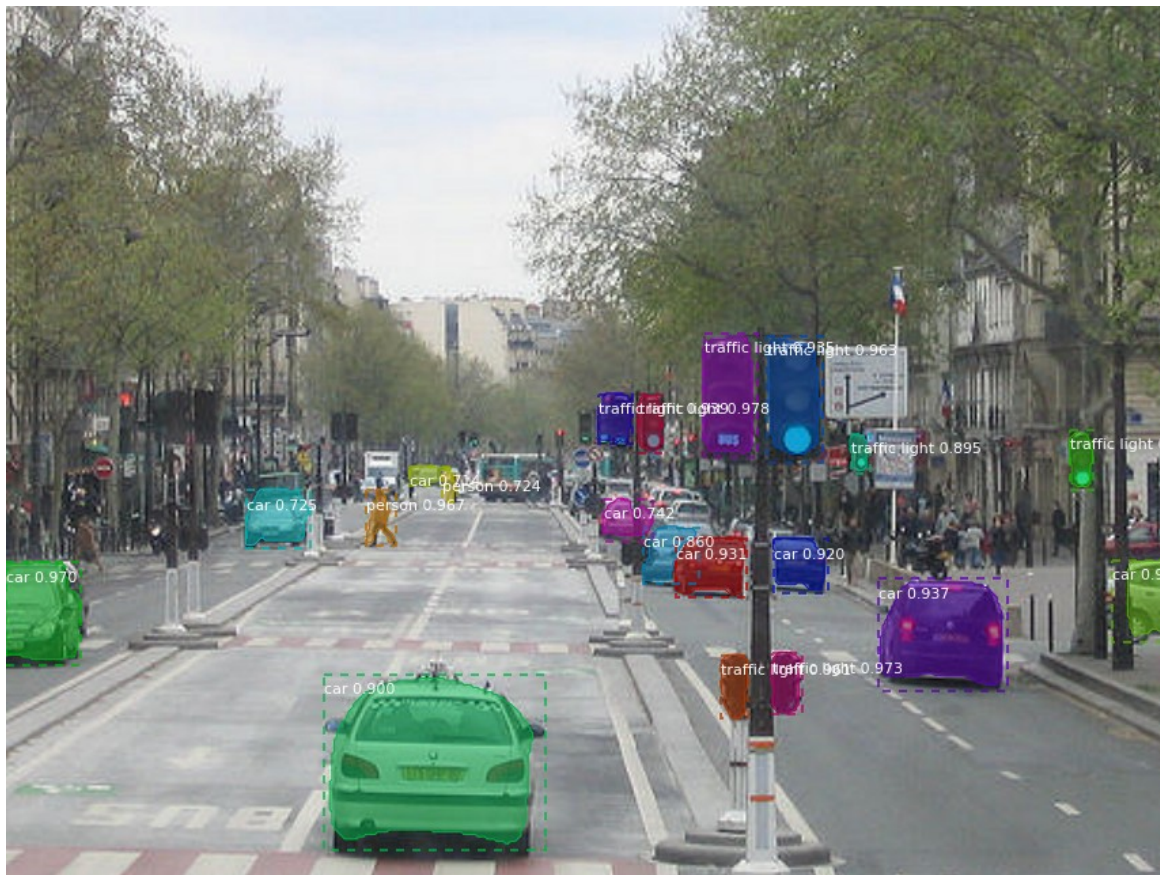
https://github.com/fslima0/UNIFACS/tree/master/projeto_smart_cities_industria_4_0

Mapa Conceitual de uma Software Similar



https://github.com/matterport/Mask_RCNN

Aplicativo em Operação (Mask RCNN)



https://github.com/matterport/Mask_RCNN

Similaridades:

- Utilizam um tipo de estrutura de dados para organizados dados
- Utiliza cores para diferentes objetos

Diferenças:

Desenvolvidas em diferentes linguagens de programação

Finalidades diferentes (Simulador vs Detector)

Referências

1. ASCENCIO, Ana Fernanda Gomes; ARAÚJO, Graziela Santos. Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++ [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. São Paulo. Pearson. 2010
2. PUBLICAÇÕES FIRJAN. Indústria 4.0: Internet das Coisas. Disponível em:
<<http://www.firjan.com.br/lumis/portal/file/fileDownload.jsp?fileId=2C908A8A557F574001559C03258877DC>>