TAD **Ponto R3:**

* **Dados:** 3 inteiros, representando as coordenadas do ponto.
* **Lista de operações:** criar\_pto, set\_pto, get\_pto, apaga\_pto, dist.

**Operações:**

**-criar\_pto:**

* **Entrada:** nenhuma
* **Pré-condição:** nenhuma
* **Processo:** Cria um ponto R3 (P)
* **Saída:** endereço do ponto P criado
* **Pós-condição:** nenhuma

**-set\_pto**

* **Entrada:** Endereço de um Ponto R3(P) e 3 inteiros (P1, P2, P3)
* **Pré-condição** endereço do ponto P ser válido
* **Processo:** atribuir à coordenada X o inteiro de P1, a Y o de P2 e Z o de P3
* **Saída:** 1 (sucesso) ou 0 (falha)
* **Pós-condição:** Altera as coordenadas de P1

**-get\_pto**

* **Entrada:** Endereço de um Ponto R3(P) e endereço de 3 inteiros (P1, P2, P3)
* **Pré-condição:** endereço do ponto P e dos inteiros P1, P2 e P3 serem válidos
* **Processo:** atribui ao inteiro P1 o valor da coordenada X, a de Y ao inteiro P2 e a de Z ao P3
* **Saída:** 1 (sucesso) ou 0 (falha)
* **Pós-condição:** Os inteiros com as coordenadas de X, Y, Z

**-apaga\_pto**

* **Entrada:** Endereço de um endereço de um ponto R3 (P)
* **Pré-condição:** endereço do ponto P ser válido
* **Processo:** liberar a memória usada pelo ponto P e limpar o seu endereço
* **Saída:** Nenhuma
* **Pós-condição:** Ponto P3 liberado

**-dist**

* **Entrada:** Endereço de dois pontos R3 (P1 e P2)
* **Pré-condição:** endereço dos pontos P1 e P2 serem válidos
* **Processo:** Cria um número real Distância e atribui a ele o valor da distância entre P1 e P2:

**Delta de X =** (coordenada de X do P1 – coordenada de X do P2) elevado ao quadrado

**Delta de Y =** (coordenada de Y do P1 – coordenada de Y do P2) elevado ao quadrado

**Delta de Z =** (coordenada de Z do P1 – coordenada de Z do P2) elevado ao quadrado

**Distância =** raiz quadrada de (Delta de X + Delta de Y + Delta de Z)

* **Saída:** valor do número Real Distância ou NULL se a operação falhar
* **Pós-condição:** Nenhuma