

# 10. Tipos de Dados Abstratos

**Princípios de Engenharia de Software (Texto em Elaboração)**

*Italo S. Vega*  
italo@pucsp.br

Faculdade de Estudos Interdisciplinares (FACEI)



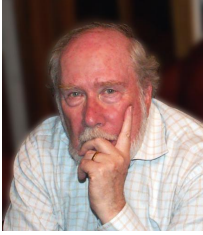
Pontifícia Universidade Católica de São Paulo



2022 Italo S. Vega

# Sumário

<b>10 Tipos de Dados Abstratos</b>	<b>4</b>
10.1 Quebra-cabeça Batalha Naval . . . . .	4
10.2 Planejamento . . . . .	6



Niklaus Wirth — Nevertheless, I consider OOP as an aspect of programming in the large; that is, as an aspect that logically follows programming in the small and requires sound knowledge of procedural programming.

# 10 Tipos de Dados Abstratos



— Consigo **identificar** tipos de dados abstratos em um determinado contexto e **elaborar** um modelo lógico que os utiliza em um programa em Python.

## 10.1 Quebra-cabeça Batalha Naval

Conversando com um colega, Fubã ficou empolgado com a história dos quebra-cabeças Batalha Naval<sup>1</sup>.



— Criarei uma versão simplificada para aprimorar o meu conhecimento a respeito de programação orientada a objetos.

Em um campo de batalha com dimensão  $5 \times 5$  encontram-se cinco navios escondidos a serem destruídos por disparos de torpedos. Cada navio ocupa uma única posição do campo de batalha. O jogador recebe uma pista a respeito de quantos navios encontram-se em cada linha e coluna do campo.



— Joguei este quebra-cabeças!

Ocara mostra o início de uma partida (Fig. 10.4). Ele está prestes a disparar um torpedo na posição (1, 5) do campo de batalha.

Em outro momento, Ocara mostra a informação produzida quando ele acerta um disparo na posição (5, 5) (Fig. ??).



— Ah! Consegui *hackear* o aplicativo e ganhei com um único disparo!

A Fig. ?? confirma a afirmação do Ocara. Embora considerada inválida (porque?) a mensagem indica que o jogo terminou com um navio destruído.

---

<sup>1</sup><https://www.conceptispuzzles.com/index.aspx?uri=puzzle/battleships/history>

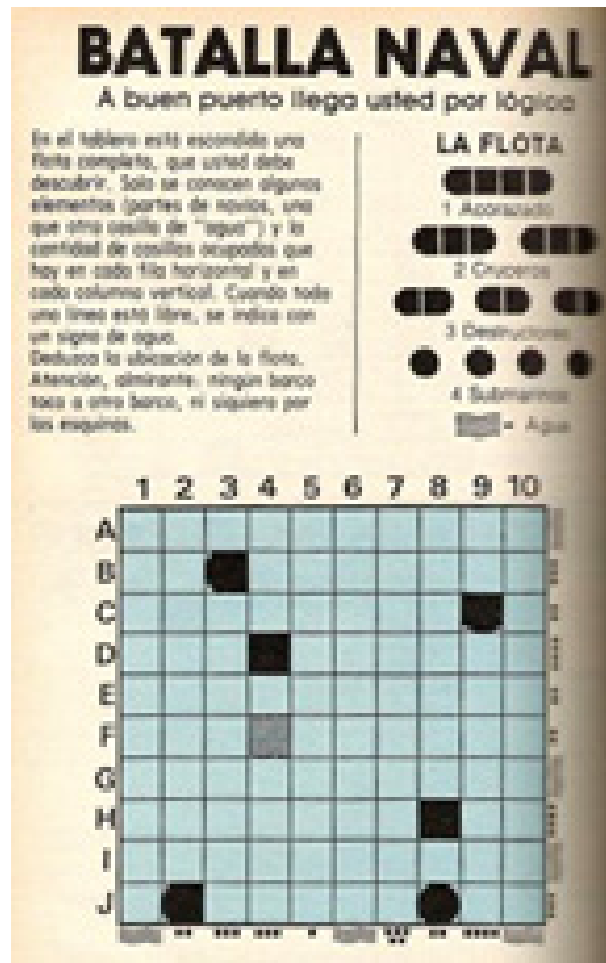


Figura 10.1: <https://www.conceptispuzzles.com/index.aspx?uri=puzzle/battleships/history>

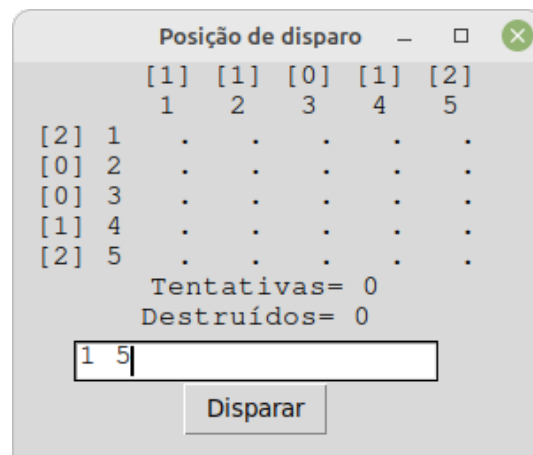


Figura 10.2: Estado inicial da partida.

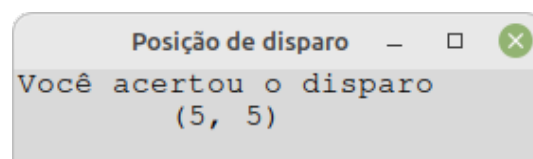


Figura 10.3: Estado inicial da partida.

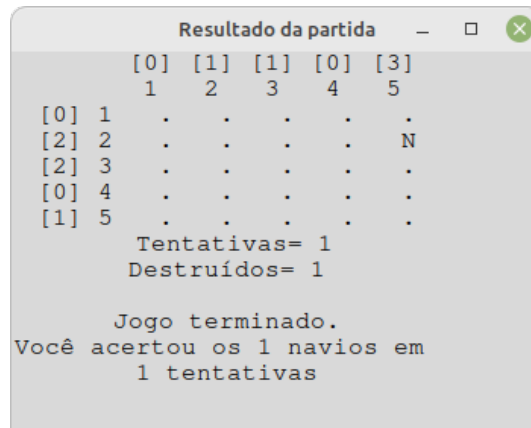


Figura 10.4: Estado inicial da partida.

## 10.2 Planejamento

Fubã abre um sorriso lateral para o Ocara, desvia o assunto, despedindo-se dele.



“Mas será que o comportamento está correto? Pedirei ajuda para o Espec. Ele pode me ajudar a especificar a lógica e implementá-la na forma de código.

Espec propõe um roteiro para o desenvolvimento do programa. Na primeira versão, a interface do usuário será abstraída, de modo que o foco se desloque para o comportamento essencial, chamado modelo de domínio. Fubã confia no Espec, embora com pouca clareza a respeito do conceito de modelo de domínio.



— Vamos tentar o roteiro a seguir.

1. Caracterização do campo de batalha. Existem 25 posições organizadas nas direções horizontal e vertical. No início, cada posição encontra-se na situação “disponível” (D) para disparo. Ao ser atingida por um torpedo, passa para o estado “água” (A) ou “com navio” (N).
2. Definição das variáveis de estado.
3. Modelagem do campo de batalha.
4. Elaboração do modelo da frota (de navios).
5. Cálculo das pistas do posicionamento dos navios.
6. Suporte ao disparo de torpedos.



— Na próxima versão, desenvolveremos a lógica de controle a camada de interface do usuário, integrando-as no modelo de domínio, depois de implementado.