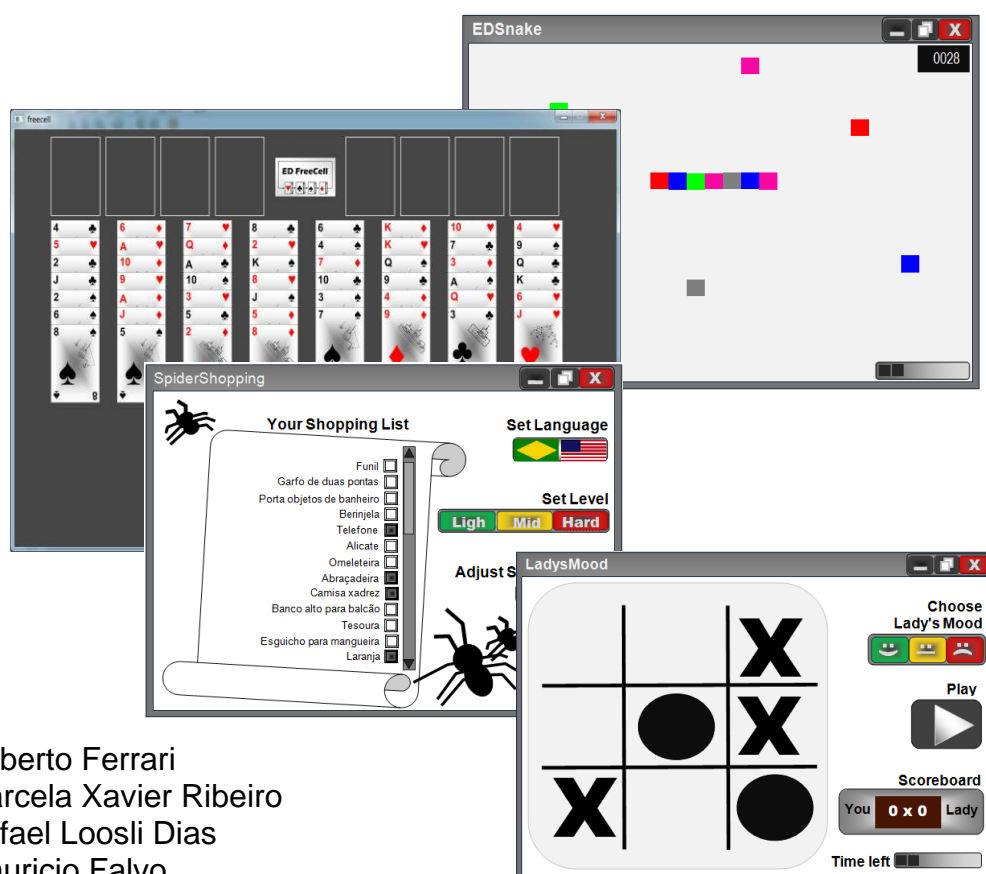


# Estruturas de Dados com Jogos

Aprenda a Programar Se Divertindo



Roberto Ferrari  
Marcela Xavier Ribeiro  
Rafael Loosli Dias  
Mauricio Falvo

Edição não comercial do livro Estruturas de Dados com Jogos © Roberto Ferrari, Marcela Xavier Ribeiro, Rafael Loosli Dias, Mauricio Falvo. Fornecida gratuitamente a estudantes em cursos e/ou disciplinas sob responsabilidade dos autores. Para uso acadêmico. Não repasse a terceiros sem a autorização dos autores.

# Conteúdo

## Apresentação

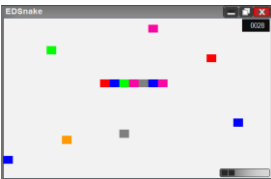


### Parte I - Pilhas

Desafio 1: Desenvolver uma Adaptação do Jogo *FreeCell*

Capítulo 1: Tipos Abstratos de Dados

Capítulo 2: Pilhas, com Alocação Sequencial e Estática



### Parte II - Filas

Desafio 2 : Desenvolver uma Adaptação do Jogo *Snake*

Capítulo 3: Filas, com Alocação Sequencial e Estática

Capítulo 4: Listas Encadeadas

Capítulo 5: Listas Encadeadas com Alocação Dinâmica

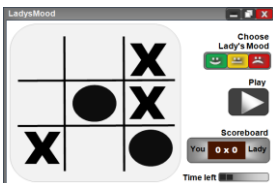


### Parte III - Listas Cadastrais

Desafio 3: Desenvolver um Jogo que use Listas de Elementos

Capítulo 6: Listas Cadastrais

Capítulo 7: Generalização de Listas Encadeadas



### Parte IV - Árvores

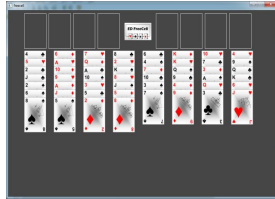
Desafio 4: Desenvolver um Game com Previsão de Jogadas

Capítulo 8: Árvores

Capítulo 9: Árvores Balanceadas

Seu Próximo Desafio

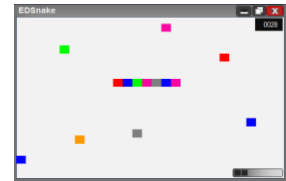
# Apresentação



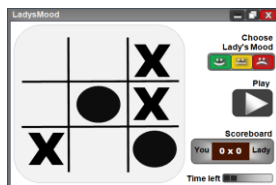
Você conhece um jogo chamado **FreeCell**? É um jogo de paciência, muito conhecido, em que você precisa manipular as cartas de um baralho, com o objetivo de colocá-las em sequência. As cartas são armazenadas em Pilhas, e só podem ser

movimentadas de acordo com regras bem específicas. Como você desenvolveria um jogo como o FreeCell? Como você faria para armazenar as cartas em Pilhas, e para garantir que as cartas sejam movimentadas de acordo com as regras do jogo?

E jogos do tipo **Snake**, você já jogou? O personagem principal desse jogo é uma cobra composta por diversos pedacinhos coloridos. Ao se movimentar e comer alguma coisa, a cobra vai ganhando ou perdendo pedacinhos, crescendo ou diminuindo, e tornando o jogo cada vez mais difícil. É um jogo de habilidade. Se você fosse desenvolver um jogo do tipo **Snake**, como representaria no programa a cobra e cada um de seus pedacinhos? Como faria para retirar ou acrescentar os pedacinhos, e para mantê-los na sequência correta?



Pense agora em um jogador humano disputando contra o computador. O jogo em si pode ser simples, como um **Jogo da Velha**, por exemplo. Mas não um jogo da velha qualquer; um jogo inteligente, no qual o computador escolhe a melhor jogada em função da jogada do adversário, sempre visando situações com as maiores chances de vitória. Como você implementaria essa inteligência em um jogo? Como faria o seu jogo prever todas as possíveis jogadas, e então escolher, conscientemente, a melhor opção?



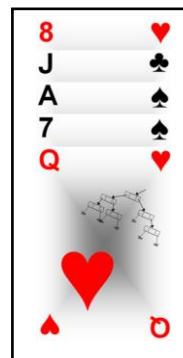
## Objetivos do Livro

O livro **Estruturas de Dados com Jogos** tem por objetivo preparar você para implementar Estruturas de Dados para representação e armazenamento de conjuntos de informações em um programa. Conjuntos de informações como uma Pilha de cartas, uma Fila de espera, uma Lista de passageiros ou de compras, por exemplo.

O foco do livro não são os Jogos em si. A ideia é que você aprenda Estruturas de Dados - desenvolva habilidades sólidas de programação, enquanto cria alguns Jogos. Os Jogos tornarão seu crescimento mais divertido!

## Os Jogos e os Capítulos

São quatro Jogos para você desenvolver. Quatro Desafios. O primeiro Desafio é desenvolver uma adaptação do *FreeCell*. Para desenvolver um bom jogo você estudará o Capítulo 1 e o Capítulo 2 de **Estruturas de Dados com Jogos**. Você vai conhecer uma Estrutura de Dados chamada **Pilha**. Você vai aprender como implementar uma Pilha de Cartas, e como deixar seus Jogos bem flexíveis. Assim será fácil ajustar o software do Jogo a novas regras e a novas situações.



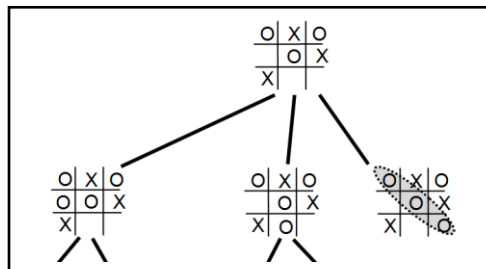
Seu segundo Desafio é desenvolver um Jogo do tipo *Snake*. Enquanto desenvolve seu próprio *Snake*, você estudará os Capítulos 3, 4 e 5, e conhecerá uma outra Estrutura de Dados chamada **Fila**. Você perceberá que uma Fila é muito útil para implementar um jogo como o *Snake*. Você irá comparar duas técnicas de implementação e escolher uma destas técnicas para implementar sua própria Fila, e seu próprio *Snake*.



No terceiro Desafio você será apresentado a um Jogo chamado **Spider Shopping**. Seu objetivo será desenvolver uma adaptação deste Jogo; ou um Jogo diferente que mantenha algumas das características do *Spider Shopping*. Para isto você estudará um outro tipo de Estrutura de Dados: a **Lista Cadastral**. Nos Capítulos 6 e 7 você estudará diversas técnicas para implementar uma Lista Cadastral, e usará sua criatividade para conceber seu próprio Jogo - sua própria aplicação das Listas Cadastrais.



O ponto chave do quarto Desafio é construir a inteligência do Jogo. Poderá ser o **Jogo da Velha**, ou outro Jogo. Mas o Jogo precisará prever as próximas jogadas, para então optar pela alternativa que ofereça maior chance de vitória. Estudando os Capítulos 8 e 9 você conhecerá uma Estrutura de Dados chamada **Árvore**, e algumas de suas aplicações. Você perceberá que com uma Árvore você poderá implementar a **Previsão de Jogadas**, e a inteligência do seu jogo.

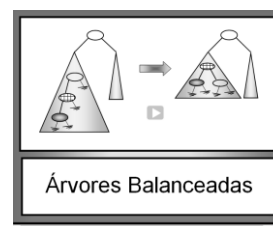
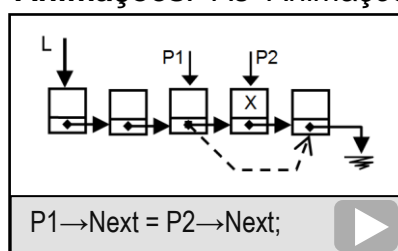


Nos quatro Desafios você terá a oportunidade de adaptar o Jogo sugerido e criar seu próprio Jogo: um Jogo com personalidade própria; um Jogo com a sua cara! Você poderá até propor Jogos

totalmente novos. Mas para que você desenvolva as habilidades pretendidas, os Jogos terão que manter algumas das características de cada Desafio. Em essência, os quatro Jogos precisarão ser aplicações de quatro Estruturas de Dados fundamentais que estudaremos ao longo deste livro: Pilhas, Filas, Listas Cadastrais e Árvores.

## Materiais Complementares

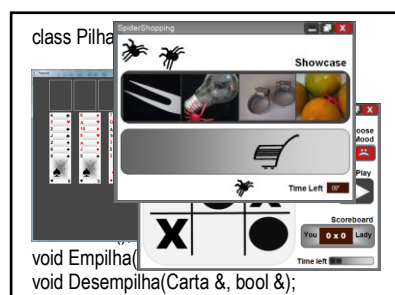
Para facilitar sua compreensão de alguns dos conceitos e algoritmos, você poderá assistir **Vídeos** com explicações e exemplos. Além dos Vídeos, os Materiais Complementares de **Estruturas de Dados com Jogos** contém algumas **Animações**. As Animações irão lhe mostrar



graficamente a execução de um trecho de programa. Você poderá interagir com as Animações, avançando a execução do programa passo a passo, enquanto observa uma representação visual do que está

acontecendo. O papel dos Vídeos e das Animações é complementar a leitura, mas nunca substituí-la. Estude o texto, faça os exercícios, e complemente seu estudo com os Vídeos e Animações dos Materiais Complementares.

Você encontrará também nos Materiais Complementares um



**Banco de Jogos.** São **Jogos Exemplo**, desenvolvidos por pessoas como você, que aceitaram o desafio de aprender Estruturas de Dados de um modo bem divertido. E que tal **Você** desenvolver Jogos legais e disponibilizá-los nesse Banco de Jogos? Que tal tornar seus Jogos públicos e conhecidos? Que tal

participar de uma **Competição de Jogos**?

## Acesso aos Materiais Complementares

Os Materiais Complementares estarão a sua disposição a qualquer tempo. Consulte-os quando quiser a partir do **Link** fornecido logo abaixo:

<http://edcomjogos.dc.ufscar.br>

## Notação Conceitual

Os algoritmos que estudaremos serão apresentados e discutidos em uma **Notação Conceitual**, sem construções de uma linguagem de programação específica. Isso permite que os conceitos sejam compreendidos em sua essência, e implementados em qualquer linguagem.

```
PegaOPrimeiro( L1, X, TemElemento );  
Enquanto TemElemento == Verdadeiro Faça  
{  
    Se EstaNaLista(L2, X)  
    Então Insere (L3, X, Ok);  
    PegaOPróximo( L1, X, TemElemento );  
};
```

Para exemplificar a implementação dos conceitos em uma linguagem de programação, em alguns momentos chave são apresentados **Códigos em C e em C++**. Mas é perfeitamente possível utilizar este livro como referência, e implementar os algoritmos em qualquer outra linguagem de programação.

```
struct Node {  
    char Info;  
    struct Node *Next; };  
typedef struct Node *NodePtr;  
NodePtr P;  
P = new Node;
```

## A Chave para um Bom Aproveitamento: Praticar

Não tenha como meta apenas *conhecer* ou *entender*. Isso não é suficiente. Tenha como meta *desenvolver habilidades* para projetar Estruturas de armazenamento de Dados, para implementar essas Estruturas e para utilizá-las na prática, seja no mundo dos Games, seja em outro contexto. E para realmente desenvolver essas habilidades, você precisará fazer uma coisa: praticar.

Pratique! Pratique muito! Você encontrará muitos exercícios neste livro, e soluções para boa parte deles. Faça os exercícios! Proponha uma primeira solução, erre, faça de novo, erre menos, faça de novo, e então acerte! Só consulte as soluções fornecidas após propor sua própria solução, ou pelo menos após tentar exaustivamente.

## Encare o Desafio!

Aprenda Estruturas de Dados. Aprenda pra valer! Em paralelo a isto, desenvolva seus Jogos! Dê personalidade própria a eles! Torne-os divertidos! Projete uma interface legal! Mostre os jogos para seus amigos! Participe de uma competição de jogos! Envolve seus colegas; desenvolva jogos junto com eles; cresça junto com eles! Divulgue seus Jogos na Internet! Faça seus jogos bombar!

Aprender a programar pode ser divertido!

<http://edcomjogos.dc.ufscar.br>