

# Laboratório de Algoritmia I

## Laboratórios de Informática II

9 de Maio de 2018



- Implementar em C um projeto;
- Aprender a trabalhar em grupo;
- Aprender a planear um projeto de software;



- Usar o **gcc** com o C11;
- Em **Linux**;
- Parte gráfica usando um servidor web com CGIs;
- O código é desenvolvido na **máquina virtual** disponibilizada.

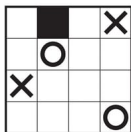
# Parte I

## O Jogo

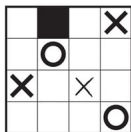
- Puzzle inspirado no Jogo do Galo;
- Jogado numa grelha;
- Tem solução única;
- Não podem existir mais do que 2 símbolos iguais consecutivos na horizontal, vertical ou diagonal.



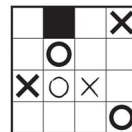
A



B



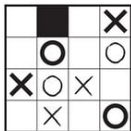
C



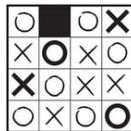
- A A posição inicial;
- B Tem que ser um **X** porque senão teríamos 3 **O** consecutivos;
- C Tem que ser um **O** porque senão teríamos 3 **X** consecutivos.



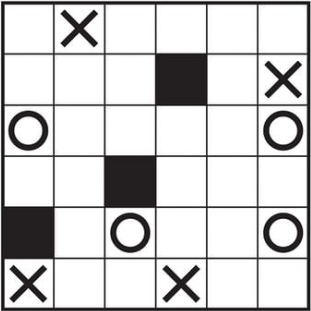
D



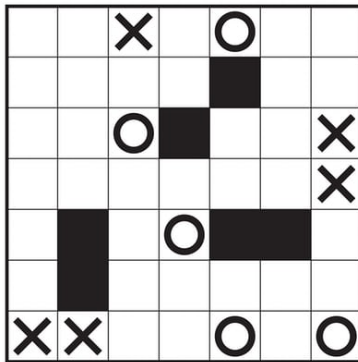
E

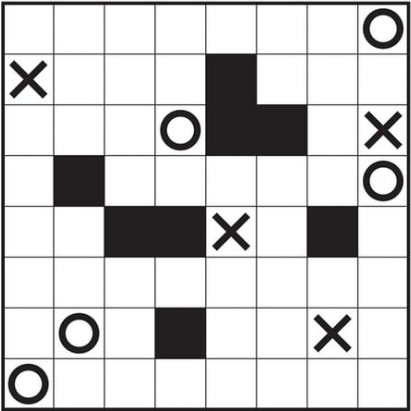


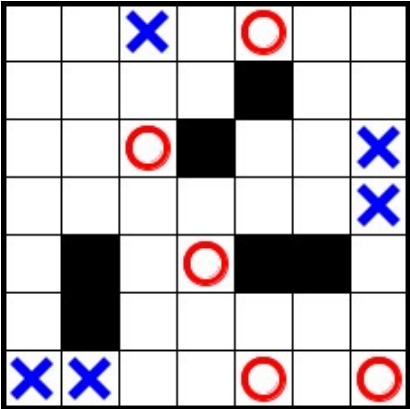
- D Seguindo o mesmo raciocínio, colocamos um **X** e depois um **O**;
- E O puzzle terminado.

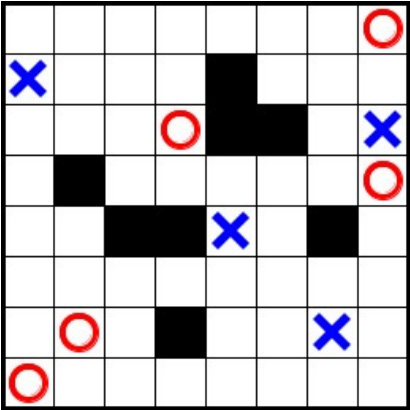








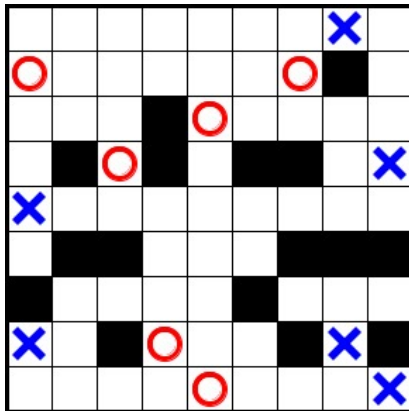


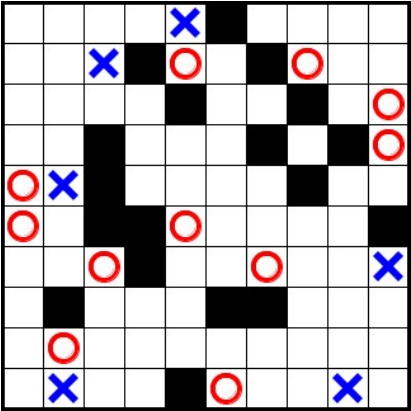




# GandaGalo

Puzzles para treinar





## Parte II

### Etapas

18/03 Jogo básico

29/04 Jogo melhorado

27/05 Jogo final

11/06 a 16/06 Defesa





# Etapa 1

## Jogo básico

- Desenhar a grelha;
- Permitir colocar um **X** ou um **O** numa casa;



## Etapa 2

Jogo melhorado

- Apresentar uma lista de puzzles e permite jogá-los;
- Ler os puzzles a partir de ficheiros;
- Verificar se o puzzle é válido;
- Dar ajudas;
- Permitir anular movimentos e colocar âncoras.



- 2 Resolver puzzles;
- 2 Gerar puzzles com várias dificuldades;
- 1 Compilar com as opções  
**gcc -std=c11 -Wall -Wextra -pedantic -O2**  
sem warnings nem erros;
- 1 Avaliação da documentação;
- 2 Avaliação de modularidade e legibilidade;
- 1 Estado em ficheiro e multi-utilizador;
- 1 Utilização de estruturas dinâmicas;
- 1 Tarefa de análise de código (gdb);
- 2 Interface com o utilizador;



- Entregar numa diretoria à parte chamada gerar;
- O comando `make` deve criar um executável chamado gerar;
- O programa deve receber os seguintes argumentos:  
`gerar <difículdade> <nº de linhas> <nº de colunas>`
- O programa deve imprimir o tabuleiro gerado segundo a estrutura dos ficheiros
- A dificuldade deve ser 1 para fácil e 2 para difícil;
- O tabuleiro impresso (no `stdout`) deve poder ser consultado pelo programa e deve ter uma única solução e a dificuldade esperada.

```
$ gerar 1 3 4  
3 4  
X.X.  
...0  
X...
```