

# Trabalho 2 – Programação socket

## “Jogo da velha”online

---

### 1.1 Orientações gerais

- O trabalho deverá ser realizado em dupla.
- Trabalhos iguais a outros trabalhos terão nota zero.

### 1.2 Objetivos

Objetivo geral do trabalho é desenvolver um “Jogo da Velha online” utilizando a biblioteca de programação *socket* na linguagem C, que permita uma partida entre dois jogadores que acessem o servidor pela rede.

Os objetivos específicos do trabalho são:

- Familiarizar-se com a programação utilizando a API *socket*.
- Implementar um protocolo simples para envio e recebimento de mensagens.
- Enviar e receber dados em uma aplicação que utiliza a arquitetura Cliente/Servidor.

### 1.3 Descrição do trabalho

Jogos online estão entre os principais entretenimentos disponíveis na Internet. Por este motivo, você está sendo desafiado a desenvolver um jogo simples, com o objetivo de desenvolver e consolidar as suas habilidades em programação, em especial a programação utilizando a API *socket*.

O jogo será desenvolvido utilizando uma arquitetura cliente/servidor, na qual o servidor coordenará o jogo entre dois clientes conectados a ele e que funcionarão a seguinte forma.

Os programas desenvolvidos deverão estar modularizados em funções para cada macro ação necessária.

#### 1.3.1 Protocolo

**Funcionamento básico do protocolo:**

- Do lado do servidor:
  1. O servidor é inicializado e aguarda a conexão de dois clientes para iniciar o jogo;
  2. Após a conexão dos dois clientes, o servidor encaminha uma mensagem de boas vindas para cada cliente.
  3. Em seguida, o servidor encaminhar uma mensagem, para o primeiro cliente, contendo: o nome do oponente, o tabuleiro vazio e a autorização para que ele faça sua jogada.
  4. Ao receber a jogada do primeiro cliente, o servidor encaminhar uma mensagem, para o segundo cliente, contendo: nome do oponente, o tabuleiro atualizado e a autorização para que ele faça sua jogada.
  5. A cada jogada o servidor verifica se houve vencedor.

6. Este processo se repete até que um usuário seja considerado vencedor ou até que finalizar as posições disponíveis do tabuleiro.
  7. O servidor aguarda 30 segundos. Em seguida desconecta todos os clientes e volta ao início esperando novos clientes.
- Do lado do cliente:
    1. O cliente aguarda a digitação do nome do usuário.
    2. Após o usuário digitar o seu nome, o cliente estabelece a conexão com o servidor, envia uma mensagem com o nome do usuário e aguarda pela mensagem inicial do servidor.
    3. Ao receber a mensagem, o cliente apresenta uma tela contendo o nome dos jogadores, o tabuleiro com as posições marcadas com símbolos diferentes para cada jogador (se houver) e solicita ao jogador que faça a sua jogada.
    4. Após a jogada, o cliente envia o tabuleiro atualizado para o servidor.
    5. Os passos 3 e 4 se repetem até que um usuário seja considerado vencedor ou até finalizar as posições disponíveis do tabuleiro.

### 1.3.2 O tabuleiro

O tabuleiro será representado pela estrutura de dados que melhor atender cada aluno. No entanto, as posições do tabuleiro devem seguir a ordem abaixo:

1	2	3
4	5	6
7	8	9

### 1.3.3 A mensagem

Cada trabalho deverá criar uma estrutura contendo a mensagem com os tipos de dados que julgar mais adequado, para atender os requisitos do protocolo.

## 1.4 Entrega do trabalho

O trabalho deverá ser postado na sala da disciplina no Moodle, utilizando a tarefa “*Envio do trabalho 2 – Programação Socket*”. É importante notar que a data limite para entrega dos trabalhos será: **12 de dezembro de 2018**. As duplas que não enviarem o trabalho dentro do prazo receberão nota zero.

Deverá ser enviado um único arquivo compactado, em formato “zip” contendo:

- Um documento no formato TXT apresentando: os autores trabalho e um minitutorial do programa desenvolvido.
  - O minitutorial deve detalhar, no mínimo:
    - Como compilar os arquivos.
    - Como executar o software cliente.

- Como executar o software servidor.
- Como visualizar o arquivo de registro.
- Os códigos fontes dos trabalhos.

## 1.5 Pontuação

O valor total do trabalho será de 100 pontos, distribuídos da seguinte forma:

- 95 pontos para o desenvolvimento dos códigos fontes.
- 05 pontos para o minitutorial.