

# LINGUAGEM DE PROGRAMAÇÃO II

Centro de Informática – Universidade Federal da Paraíba – Campus I

Professor: Carlos Eduardo Batista - Semestre: 2020.1

Entrega por e-mail: 23h59 de 11/12/2020.

O título do e-mail deve conter: “[LPII-20201-E004] NOME DO ALUNO – MATRICULA”. Arquivo de entrega deve anexar todos os códigos fonte em C/C++ dentro de um diretório nomeado “MATRICULA\_ALUNO-LPII-20201-E004” o qual deve ser comprimido em um arquivo ZIP (“MATRICULA\_ALUNO-LPII-20201-E004”). O arquivo ZIP deve conter obrigatoriamente um arquivo de texto chamado README.txt contemplando todas as instruções de compilação e execução, além de qualquer observação que se fizer necessária para correção.

## TRABALHO PRÁTICO: *SOCKETS (3,0 PTS)*

Faça um programa em C/C++ que atue como um servidor de verificação de palíndromos. O servidor deverá aceitar e tratar até 6 conexões simultâneas de clientes; ao receber uma nova conexão, o servidor deverá dedicar uma thread a ela, para sua correta manipulação.

O servidor enviará uma string de boas vindas para cada nova conexão (“BENVINDO”), e deverá exibir o endereço do novo cliente conectado no terminal local. Os clientes poderão enviar uma quantidade arbitrária de strings com até 10 caracteres para o servidor, que deverá responder com “PALINDROMO=TRUE” caso a string contenha um palíndromo e com “PALINDROMO=FALSE” caso a string não contenha um palíndromo. Palíndromos são frases ou palavras que podem ser lidas, indiferentemente, da esquerda para a direita, conforme o sentido habitual da leitura, ou da direita para a esquerda, sem que haja mudança da sua significação (exemplos de palíndromos com 6 letras: “AAAAAA”, “ABAABA”, “ABCCBA”, “MUSSUM”).

Caso o cliente insira a string “BYEBYE” sua conexão deverá ser finalizada. O servidor deverá tratar conexões indefinidamente, até que algum cliente insira a string “SHUTDOWN”, aí então o servidor deverá ser finalizado.

A implementação deverá ser feita em C/C++ utilizando a biblioteca Pthreads ou as threads STD (C++11) e a biblioteca sockets.h (ou winsock.h). A saída do seu programa deve ser bem planejada, de forma a mostrar o que está acontecendo em cada momento relevante. Para testes do servidor, usar netcat (nc) ou telnet para o lado do cliente.