Universidade Estácio

Juazeiro da Bahia

Desenvolvimento Full Stack

Por que não paralelizar

Relatório discente de acompanhamento

Nome: Gerson José de Almeida Júnior

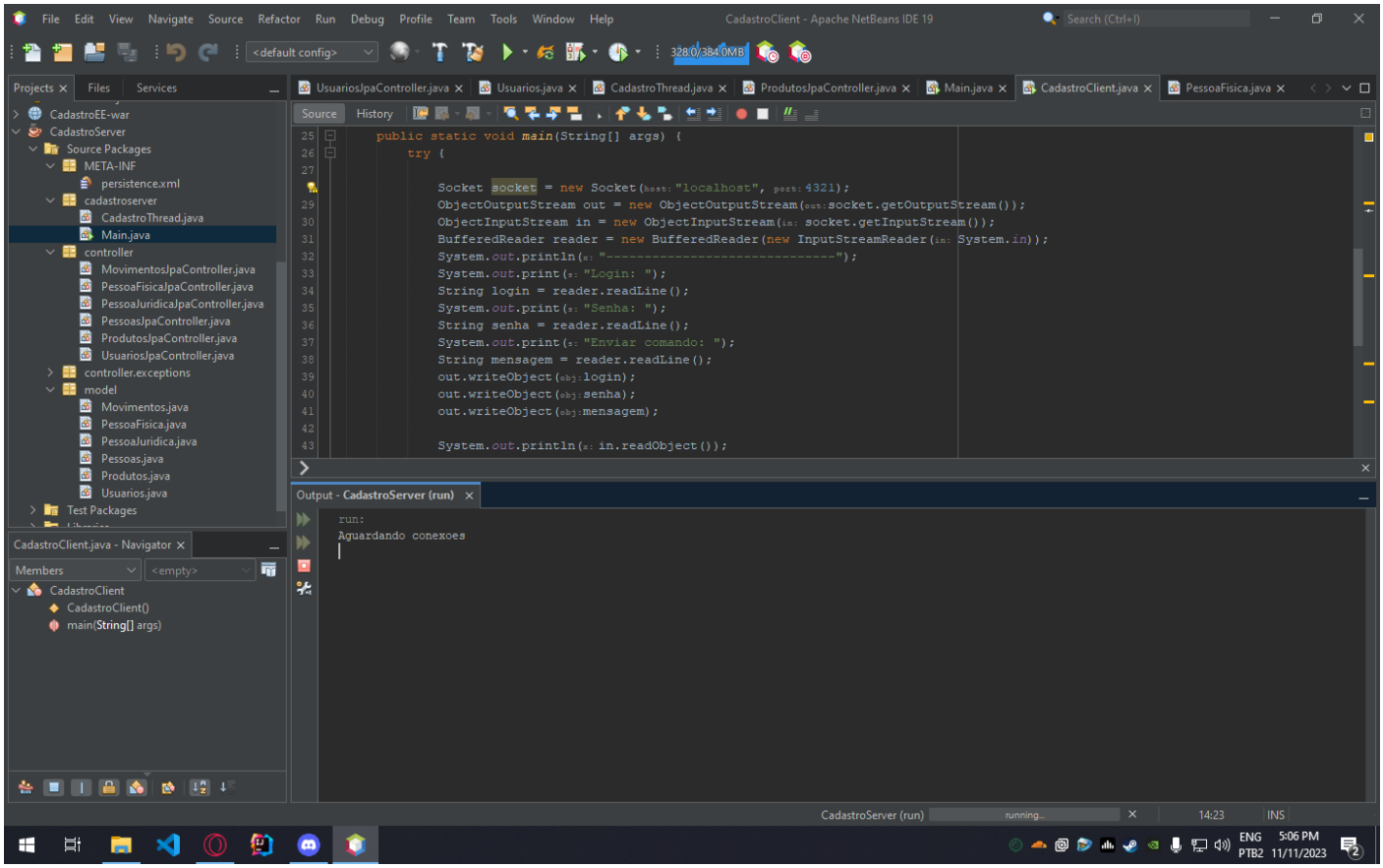
Número da Turma: 23.3

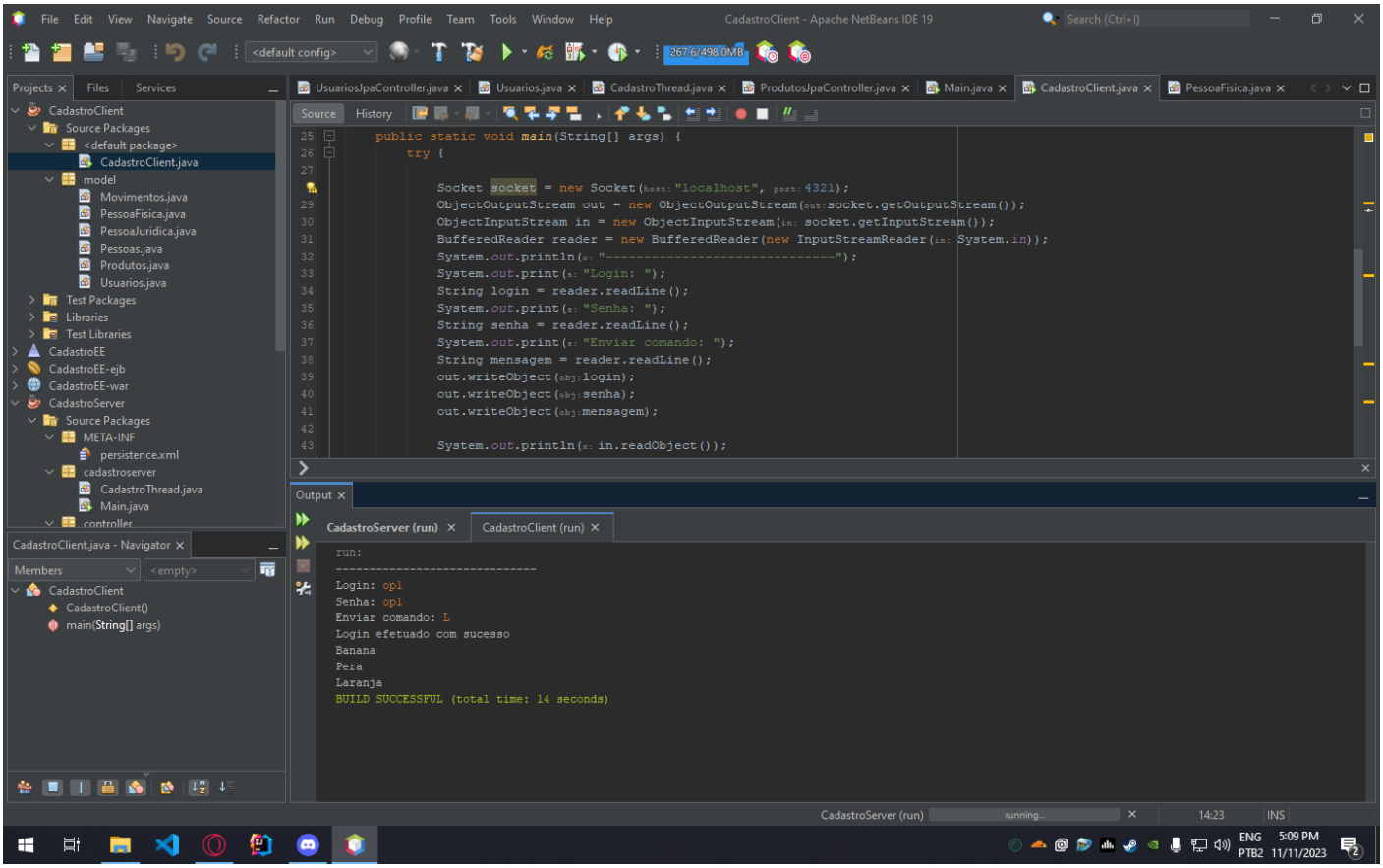
Semestre Letivo: 3° Semestre

Github: https://github.com/gersonjose9713/Missao-Pratica-Nivel-5-Mundo-3/

Criando o Servidor e Cliente de Teste

Códigos Desenvolvidos:





Análise e Conclusão:

Como funcionam as classes Socket e ServerSocket?
As classes Socket e ServerSocket são fundamentais para a comunicação em redes utilizando o protocolo TCP/IP.

Socket: Representa um endpoint em uma rede. Um socket do tipo cliente é usado para iniciar uma conexão, enquanto um socket do tipo servidor aguarda conexões de clientes. É responsável por estabelecer e manter a conexão entre dois sistemas.

ServerSocket: Utilizado no lado do servidor, ele espera por pedidos de conexão vindos de clientes. Quando um pedido é aceito, ele cria um novo socket para a comunicação com o cliente.

Qual a importância das portas para a conexão com servidores?
As portas são números de identificação atribuídos a processos específicos em um dispositivo. Em uma conexão, a combinação de endereço IP e porta permite rotear dados para o aplicativo ou serviço correto. As portas facilitam a comunicação simultânea de vários serviços em um único dispositivo.

Portas no Contexto de Servidores: Permitem que vários serviços (como servidores web, servidores de email) rodem simultaneamente no mesmo dispositivo, cada um associado a uma porta específica.

Para que servem as classes de entrada e saída ObjectInputStream e ObjectOutputStream, e por que os objetos transmitidos devem ser serializáveis?

Serialização: É o processo de converter um objeto em uma sequência de bytes. A serialização é necessária para transmitir objetos pela rede.

ObjectInputStream: Lê objetos de um fluxo de entrada.

ObjectOutputStream: Escreve objetos em um fluxo de saída.

Essas classes são usadas para serializar e desserializar objetos em Java, permitindo a transmissão de objetos pela rede

Por que, mesmo utilizando as classes de entidades JPA no cliente, foi possível garantir o isolamento do acesso ao banco de dados?

As classes de entidades JPA (Java Persistence API) no cliente fornecem uma camada de abstração sobre o banco de dados. Permite o mapeamento de objetos Java para entidades de banco de dados.

Isolamento com JPA: Mesmo que as classes de entidades sejam usadas no cliente, o acesso real ao banco de dados ocorre no lado do servidor. Gerencia a comunicação entre o cliente e o banco de dados, garantindo que as operações no banco de dados sejam controladas e isoladas no servidor.

Servidor Completo e Cliente Assíncrono

Códigos Desenvolvidos: