💼 Trabalho Prático – Sistema de Gestão de Pedreira

🎯 Objetivo Geral

Desenvolver uma aplicação de consola (linha de comandos) em Java, com base de dados PostgreSQL, que permita gerir o processo de corte de mármore numa pedreira.

🧱 Parte 1 – Registo de Blocos de Mármore (20%)

Cada bloco de mármore recebido pela pedreira deve ser registado com os seguintes dados:

- Tipo do mármore (String)
- Comprimento em metros (double)
- Altura em metros (double)
- Estado (disponível/reservado)

🛠️ O que fazer:

1. Criar uma classe `Block` com os atributos acima.
2. Criar uma opção no menu (em português) para "Registar novo bloco".
3. Guardar os dados na base de dados, tabela `blocks`, com o campo `status` (`available` ou `reserved`).

🔁 Exemplo: O utilizador insere os dados no terminal e os valores são guardados na tabela `blocks`.

📦 Parte 2 – Registo de Encomendas (25%)

Os clientes fazem encomendas indicando:

- Número de unidades desejadas (int)
- Comprimento por unidade (double)
- Altura por unidade (double)
- Espessura por unidade (double)

🛠️ O que fazer:

1. Criar uma classe `Order`.
2. Criar uma opção no menu (em português) para "Registar nova encomenda".
3. Guardar os dados na base de dados, tabela `orders`.
4. Calcular e apresentar ao utilizador uma estimativa do número de blocos necessários para satisfazer a encomenda.
5. Reservar automaticamente os blocos necessários (atualizar o campo `status` dos blocos para `reserved`).

🪚 Parte 3 – Cálculo de Cortes (Placas e Unidades) (20%)

🧮 O que calcular:

Dado uma espessura introduzida pelo utilizador, o sistema deve calcular:

1. Número teórico de **placas** por bloco disponível: `altura do bloco ÷ espessura introduzida`
2. Número de **unidades** por placa: `(comprimento × espessura do bloco) ÷ área da unidade`
3. Total de **unidades** possíveis com o stock disponível (considerando apenas blocos com `status = available`)

🛠️ O que fazer:

- Criar funções na aplicação para simular cortes.
- Mostrar os resultados ao utilizador com base no stock disponível.

📊 Parte 4 – Gestão de Stock (10%)

🛠️ O que fazer:

- Criar funcionalidade para listar blocos disponíveis (com `status = available`).

☑️ Parte 5 – Funcionalidades Obrigatórias (15%)

O menu da aplicação (em português) deve permitir:

1. Registar novo bloco
2. Listar blocos disponíveis
3. Registar nova encomenda
4. Listar encomendas
5. Simular cortes teóricos (placas/unidades)

➡️ Criar uma classe `Menu` responsável por apresentar as opções e chamar os métodos respetivos.

🩹 Parte 6 – Testes (10%)

- Usar JUnit 5 para testar os métodos de cálculo (por exemplo: cálculo de placas e unidades).
- Criar pelo menos 3 testes automatizados.

💾 Requisitos Técnicos

- Linguagem: Java (>= 17)
- Base de dados: PostgreSQL
- Interface: Linha de comandos (Scanner)
- Testes: Usar JUnit 5

🕒 Duração Máxima

- Tempo total para o desenvolvimento: **5 horas**

📦 Entrega

A entrega deve conter:

- Código-fonte completo (ficheiros `.java`)
- Script SQL de criação da base de dados
- Ficheiro `.zip` com todos os ficheiros

Boa sorte e bom trabalho!