Disciplina:	Introdução a Back-end Development
Professor:	Leonardo Ruiz Orabona

Descrição da Atividade

Objetivo: *Desenvolver uma aplicação em JavaScript utilizando Node.js que permita a gestão de notas e médias dos alunos. A aplicação deve receber o nome do aluno, nome das matérias, as notas das matérias, calcular a média individual e contabilizar as faltas.*

Funcionalidades:

1. **Cadastro do Aluno:**
   - Solicitar o nome do aluno.
2. **Cadastro das Matérias:**
   - O aluno pode cadastrar no mínimo 3 matérias.
   - O cadastro de matérias continua até que o usuário decida parar.
3. **Cadastro de Notas:**
   - Para cada matéria cadastrada, solicitar 3 notas.
4. **Cálculo de Média:**
   - Calcular a média individual de cada matéria.
5. **Cadastro e Contabilização de Faltas:**
   - Solicitar o número de faltas para cada matéria.
   - Verificar se o aluno está reprovado por faltas (mais de 5 faltas em qualquer matéria).
6. **Resultados:**
   - Exibir a média de cada matéria.
   - Indicar se o aluno está aprovado ou reprovado em cada matéria, considerando tanto a média das notas quanto as faltas.

Requisitos Técnicos:

- Utilizar JavaScript com Node.js.
- Implementar lógica para cálculos e verificações necessárias.

Formas de Entrega:

- **Link do GitHub:** Suba o código em um repositório público ou privado no GitHub e forneça o link.
- **Documento Word:** Insira o código e as instruções em um documento Word.
- **Pasta Zipada:** Coloque todos os arquivos em uma pasta, compacte a pasta em um arquivo zip e envie o arquivo zipado.

Critérios de Avaliação:

7. **Funcionalidade:**
    - A aplicação deve ser capaz de solicitar o nome do aluno e permitir o cadastro de matérias e notas.
    - A aplicação deve calcular corretamente a média das notas para cada matéria.
    - A aplicação deve solicitar e contabilizar o número de faltas para cada matéria.
    - A aplicação deve exibir corretamente os resultados, incluindo a média, faltas e status de aprovação ou reprovação.
8. **Usabilidade:**
    - A interface de linha de comando deve ser intuitiva e fácil de usar.
    - As instruções e perguntas feitas ao usuário devem ser claras e compreensíveis.
    - A aplicação deve permitir a adição de múltiplas matérias até que o usuário decida parar.
9. **Robustez e Tratamento de Erros:**
    - A aplicação deve tratar entradas inválidas (como notas não numéricas, faltas negativas, etc.) de forma adequada.
    - A aplicação deve lidar com possíveis erros durante a execução, sem quebrar ou fechar inesperadamente.
10. **Código e Estrutura:**
    - O código deve ser bem estruturado, com funções bem definidas e nomes de variáveis claros.
    - O código deve seguir boas práticas de programação, incluindo comentários explicativos onde necessário.
    - O código deve ser modular e reutilizável, facilitando futuras expansões e manutenção.